The SirCam Worm and a NetWare Network
Genesius C. Jaromsky
SANS Security Essentials
GSEC Practical Assignment
Version 1.2e
August 15, 2001

Introduction

All because you are running the Novell NetWare operating system on your network does not mean you cannot be affected by one of the latest and possibly greatest (well designed) viruses/worms to strike computers thus far. Enter the SirCam Worm.

First, What is a worm?

"Originally coined in a 1982 paper by researchers John Shoch and Jon Hupp of the Xerox Palo Alto Research Center, the term "worm" is derived from "The Shockwave Rider", a 1972 science-fiction novel about the downfall of an Orwellian society caused, to some degree, by a "tapeworm" program that liberated data as it proliferated through networks." [1]

"Computer worms are not ordinary viruses. Their ability to spread quickly across the Internet has made worms the weapon of choice for malicious vandals to spread their latest creations. Furthermore, the programs can be easily copied and changed, and point-and-click tools to create complex worms are readily available." [2]

Many worms are written by "script-kiddies" using virus making kits that can be found on numerous sites on the Internet. This worm however, is very special.

What is so special about SirCam?

SirCam is a multifaceted piece of malicious code. It propagates through different internal mechanisms. It can also cause a DDoS (Distributed Denial of Service) attack on your PC (or network) and possibly delete all your files (based on several factors, including a random number generator).

SirCam, or W32.SirCam.Worm@mm, "…is malicious code that spreads through email and potentially through unprotected network shares. Once the malicious code has been executed on a system, it may reveal or delete sensitive information."[3]

It propagates via email through use of it's own SMTP client included in the worm. It steals email addresses from .wab (Windows Address Book) files and searches for them in "\Temporary Internet Files\ folder ('sho*', 'get*', 'hot*', and '*.html')". [4]

"SirCam arrives with a virus-infected attachment and a random document culled from an infected machine. If the recipient clicks on the attachment, his or her machine is infected with SirCam. The virus, which affects all e-mail programs -- not just Outlook -- is causing problems for people who practice safe computing and don't even click on the infested attachment." [5]

Some Damage Already Caused by SirCam

Since it's discovery, on July 17, 2001, there have been two high-profile incidents of sensitive information being transmitted across the Internet because of the SirCam Worm. First, on Wednesday, July 25th, "…the worm infected the PC of a researcher at the NIPC, and though it did not spread throughout the NIPC, Sircam did send eight internal documents marked "official use only" to outsiders, the Journal reported. The Journal also reported that no classified or sensitive information was released, according to FBI spokeswoman Debbie Weierman. Weierman did not return repeated calls for comment from the IDG News Service Wednesday morning" [6].

Next, "Reuters reported that a Ukrainian Web site said Thursday it had received secret documents from the administration of President Leonid Kuchma, including an itinerary showing his whereabouts during the country's upcoming independence celebration." [7]

Given that SirCam includes its own SMTP engine it is not restricted to using Microsoft Outlook Express to spread. It searches for WAB files. WAB is used for storing contact information (including email addresses) for other programs, besides Outlook Express to use. For example, WAB is used by Microsoft Money Deluxe & Business to keep track of the names, addresses, email of your payees, debtors, etc.

If SirCam is not able to find any email address within WAB or temporary Internet files, the worm will send infected files to one of four default email servers located in Mexico.

1 prodigy.net.mx
2 enlace.net.mx
3 goeke.net.mx
4 dobleclick.net.mx

It is believed that the writer/programmer of this worm is from Mexico because of information found within the code. This could be a diversion though, to throw authorities and emergency response teams off their trail.

"SirCam could have been both the smartest and the most destructive virus ever, if its actions…had been programmed to activate every single time the virus infected a machine." [8]

Luckily for security professionals, network administrators, and computer users in general, this worm uses a random number generator to decide the fate of the infected user's PC. Also, SirCam was written as a compiled program; therefore, it will be very difficult, although not impossible, for a copycat of this worm to hit the net in the near future.

My Dilemma with SirCam and Novell.

Even though the Novell NetWare operating system is immune from viruses and other malicious code, a Novell network might not be. For versions 4.x and below of Novell NetWare, IPX is the protocol of choice for the network transport. In the past, NetWare has been free from the virus and security pitfalls that Microsoft Windows 9.x and above have been plagued with. SirCam might be NetWare's first viral adversary.

My company runs a Novell NetWare 4.2 network with approximately thirty offices, each with a Netware file server. Just two after the worm's discovery, on Thursday, July 19th, several Windows 9.x PCs located in our Atlanta office were infected with the SirCam worm. It took us several man-hours to clean up because our McAfee anti virus dat files were out of date (and in some cases –the engine as well). However, the outbreak was localized to this one office and our desktop support team was able to contain it. Over the weekend, several antiviral manufacturers raised their risk assessment of SirCam to 'High'.

On Monday, July 23rd our networking staff was hit with a serious issue of their own. Users in various offices across our US WAN were logging into file servers other than those, which they have, access rights to. At this time our desktop team experienced another, but more wide spread, outbreak of SirCam. For the next few days, our IT department was busy trying to resolve these two, apparently separate, problems.

While our desktop team attempted to control the outbreak, our network staff was on the phone with Novell Technical Support. We ran a Sniffer trace and sent the file to Novell for further analysis. The trace included traffic generated by one of the users that was logging into multiple servers. While checking the trace file myself I found some very interesting data (breakdown follows). *Note: Certain names within the Novell context have been removed for security/privacy reasons.

Sniffer Trace File Screen Shots

Following is some important information to know and understand.

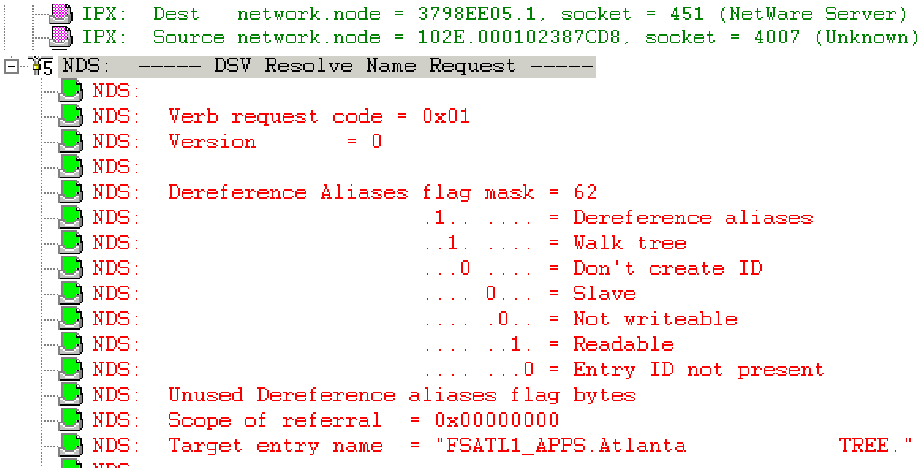
network.node	Alias (for this report)	Comments
102E.000102387cd8	SirCamPC	This is the IPX network and the MAC address of the infected PC.
3798EE05.1	LocalFS	This is the local file server that the infected PC has file access rights to.

NDS or NCP Packet Type	Function
NDS "DSV Resolve Name Request"	A request from the PC to the server for information about an object found within the tree. The request is in the form of a distinguished name.
NDS "DSV Resolve Name Response"	The response to the PC from the server. The response is in the form of an EID.
NDS "DSV Read Request"	A request from the PC to the server for information, i.e. network address, regarding an EID.
NDS "DSV Read Reply"	The reply to the PC containing another EID or the network address.

NCP “Open/Create File or Subdirectory Request”	The request to copy or create a file.
NCP “Open/Create File or Subdirectory Reply”	The reply from a copy or create file request.

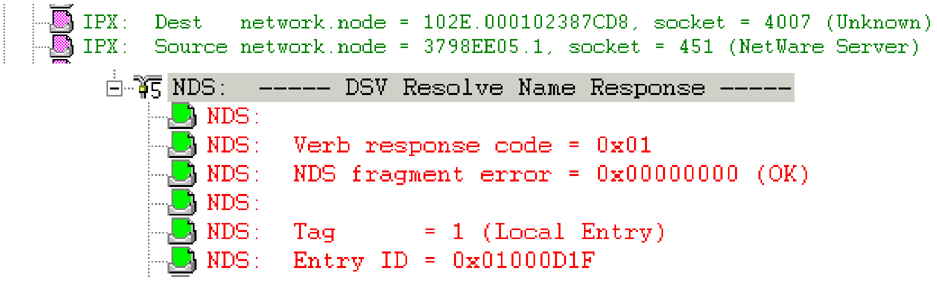
SirCam’s job is to find Windows network shares to infect.

In Frame 8, (see Fig. 1), SirCamPC sends a "*DSV Resolve Name Request*" packet to LocalFS to locate information for a Netware volume, ‘FSATL1_APPS.Atlanta.TREE’, that the Novell client on the SirCamPC found. SirCamPC tells LocalFS to walk the tree.



```
IPX:  Dest   network.node = 3798EE05.1, socket = 451 (NetWare Server)
IPX:  Source network.node = 102E.000102387CD8, socket = 4007 (Unknown)
NDS:  ----- DSV Resolve Name Request -----
NDS:
NDS:  Verb request code = 0x01
NDS:  Version       = 0
NDS:
NDS:  Dereference Aliases flag mask = 62
NDS:                        .1.. .... = Dereference aliases
NDS:                        ..1. .... = Walk tree
NDS:                        ...0 .... = Don't create ID
NDS:                        .... 0... = Slave
NDS:                        .... .0.. = Not writeable
NDS:                        .... ..1. = Readable
NDS:                        .... ...0 = Entry ID not present
NDS:  Unused Dereference aliases flag bytes
NDS:  Scope of referral  = 0x00000000
NDS:  Target entry name  = "FSATL1_APPS.Atlanta            TREE."
```

Figure 1 – Frame 8.

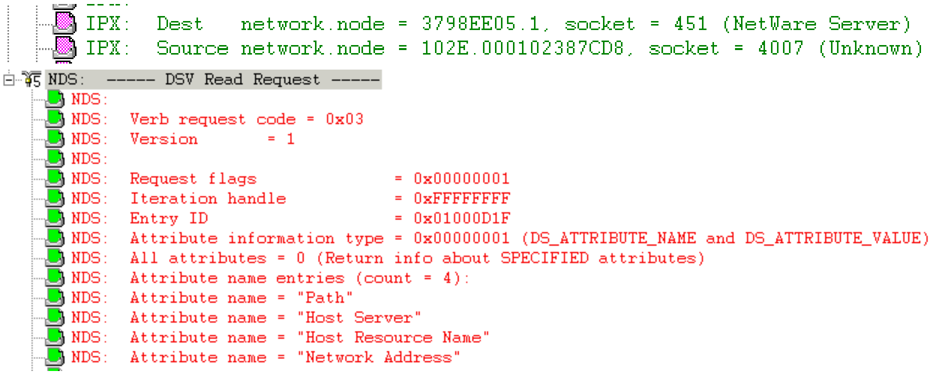


```
IPX:  Dest   network.node = 102E.000102387CD8, socket = 4007 (Unknown)
IPX:  Source network.node = 3798EE05.1, socket = 451 (NetWare Server)
NDS:  ----- DSV Resolve Name Response -----
NDS:
NDS:  Verb response code = 0x01
NDS:  NDS fragment error = 0x00000000 (OK)
NDS:
NDS:  Tag       = 1 (Local Entry)
NDS:  Entry ID = 0x01000D1F
```

Figure 2 – Frame 9.

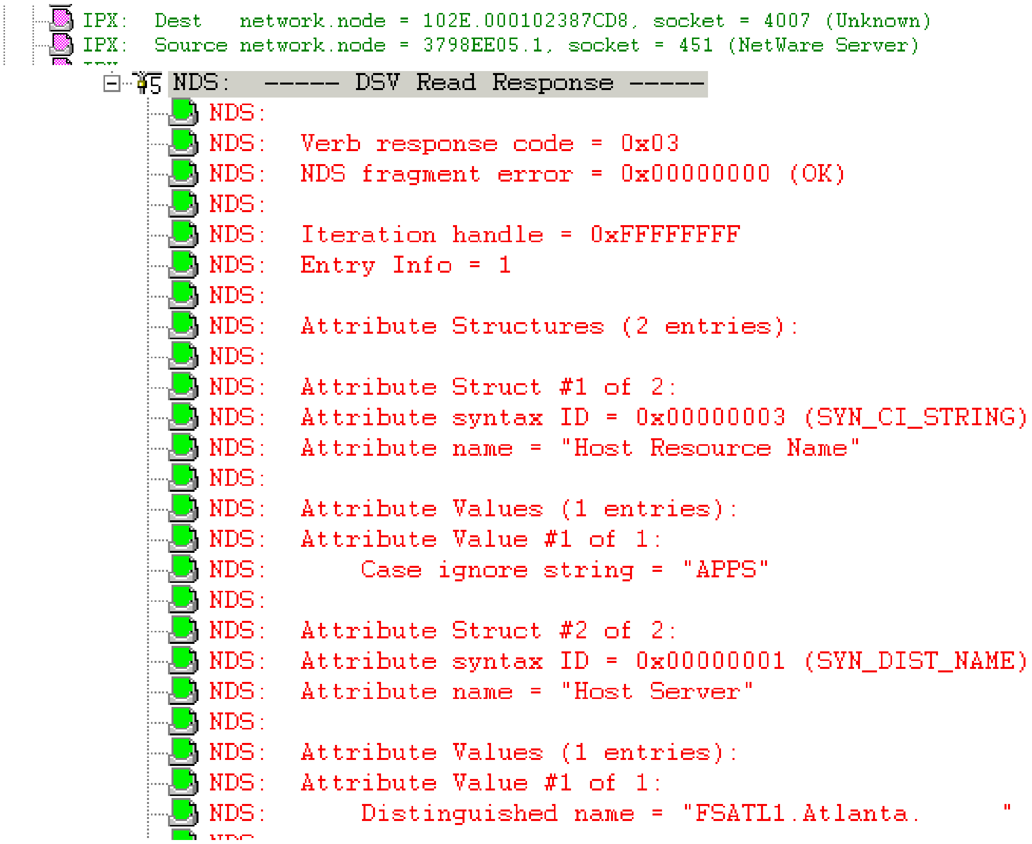
In Frame 9, (see above Fig. 2), LocalFS returns a "*DSV Resolve Name Response*" packet with an Entry ID (EID) that is tagged as local: ‘0x01000d1f’.

In Frame 12, (see Fig. 3) SirCamPC continues its pursuit to find a network share to infect and sends a "*DSV Read Request*" packet to LocalFS to find specific attributes about the EID ‘0x01000d1f’. The attributes are: Path, Host Server, Host Resource Name, and Network Address.



```
IPX:  Dest   network.node = 3798EE05.1, socket = 451 (NetWare Server)
IPX:  Source network.node = 102E.000102387CD8, socket = 4007 (Unknown)
NDS:  ----- DSV Read Request -----
NDS:
NDS:  Verb request code = 0x03
NDS:  Version       = 1
NDS:
NDS:  Request flags              = 0x00000001
NDS:  Iteration handle           = 0xFFFFFFFF
NDS:  Entry ID                   = 0x01000D1F
NDS:  Attribute information type = 0x00000001 (DS_ATTRIBUTE_NAME and DS_ATTRIBUTE_VALUE)
NDS:  All attributes = 0 (Return info about SPECIFIED attributes)
NDS:  Attribute name entries (count = 4):
NDS:  Attribute name = "Path"
NDS:  Attribute name = "Host Server"
NDS:  Attribute name = "Host Resource Name"
NDS:  Attribute name = "Network Address"
```

Figure 3 – Frame 12.

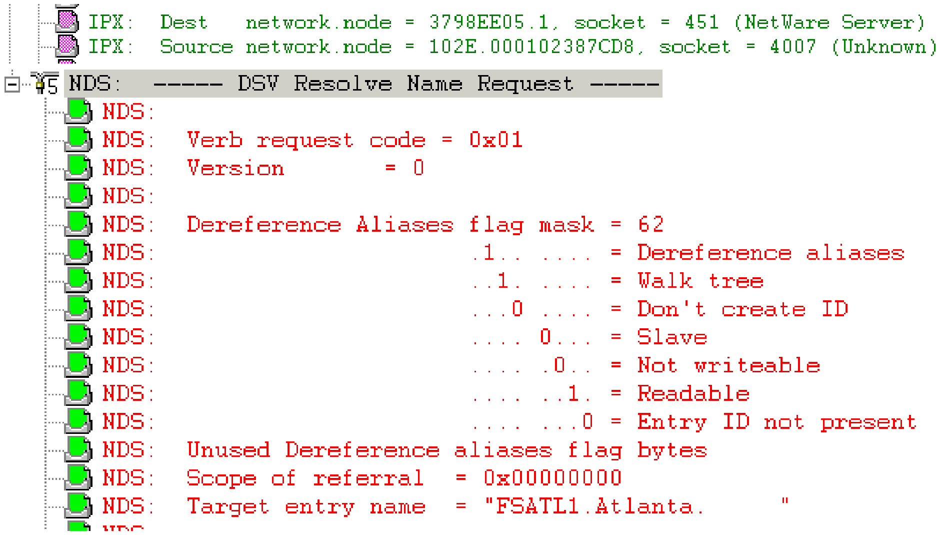


```
IPX:  Dest   network.node = 102E.000102387CD8, socket = 4007 (Unknown)
IPX:  Source network.node = 3798EE05.1, socket = 451 (NetWare Server)
NDS:  ----- DSV Read Response -----
NDS:
NDS:  Verb response code = 0x03
NDS:  NDS fragment error = 0x00000000 (OK)
NDS:
NDS:  Iteration handle = 0xFFFFFFFF
NDS:  Entry Info = 1
NDS:
NDS:  Attribute Structures (2 entries):
NDS:
NDS:  Attribute Struct #1 of  2:
NDS:  Attribute syntax ID = 0x00000003 (SYN_CI_STRING)
NDS:  Attribute name = "Host Resource Name"
NDS:
NDS:  Attribute Values (1 entries):
NDS:  Attribute Value #1 of  1:
NDS:      Case ignore string = "APPS"
NDS:
NDS:  Attribute Struct #2 of  2:
NDS:  Attribute syntax ID = 0x00000001 (SYN_DIST_NAME)
NDS:  Attribute name = "Host Server"
NDS:
NDS:  Attribute Values (1 entries):
NDS:  Attribute Value #1 of  1:
NDS:      Distinguished name = "FSATL1.Atlanta.     "
```

Figure 4 – Frame 14.

In Frame 14, (see above Fig. 4), LocalFS returns a "*DSV Read Response"* packet with a distinguished name "FSATL1.Atlanta" to SirCamPC. LocalFS notifies SirCamPC to ignore ‘APPS’, the volume name (path) used in the original DSV Read Request.

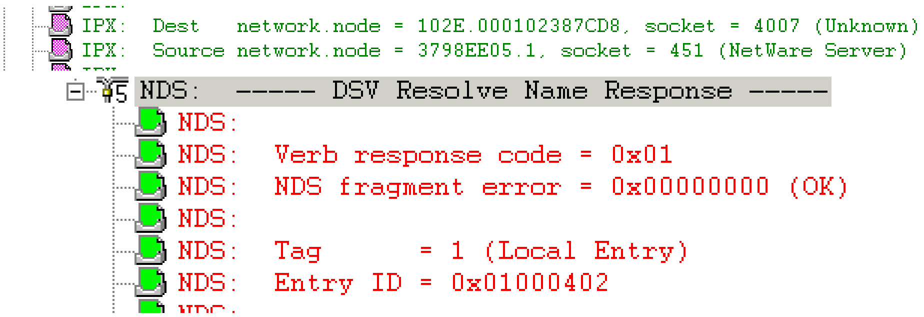
In Frame 17, (see Fig. 5), SirCamPC returns a "*DSV Resolve Name Request"* packet with a distinguished name ‘FSATL1.Atlanta.’ to USER.



```
IPX:  Dest   network.node = 3798EE05.1, socket = 451 (NetWare Server)
IPX:  Source network.node = 102E.000102387CD8, socket = 4007 (Unknown)
NDS:  ----- DSV Resolve Name Request -----
NDS:
NDS:  Verb request code = 0x01
NDS:  Version        = 0
NDS:
NDS:  Dereference Aliases flag mask = 62
NDS:                      .1.. .... = Dereference aliases
NDS:                      ..1. .... = Walk tree
NDS:                      ...0 .... = Don't create ID
NDS:                      .... 0... = Slave
NDS:                      .... .0.. = Not writeable
NDS:                      .... ..1. = Readable
NDS:                      .... ...0 = Entry ID not present
NDS:  Unused Dereference aliases flag bytes
NDS:  Scope of referral  = 0x00000000
NDS:  Target entry name  = "FSATL1.Atlanta.     "
```

Figure 5 – Frame 17.

In Frame 18, (see Fig. 6), LocalFS returns a "*DSV Resolve Name Response*" packet with an EID tagged local: ‘0x01000402’ to SirCamPC.

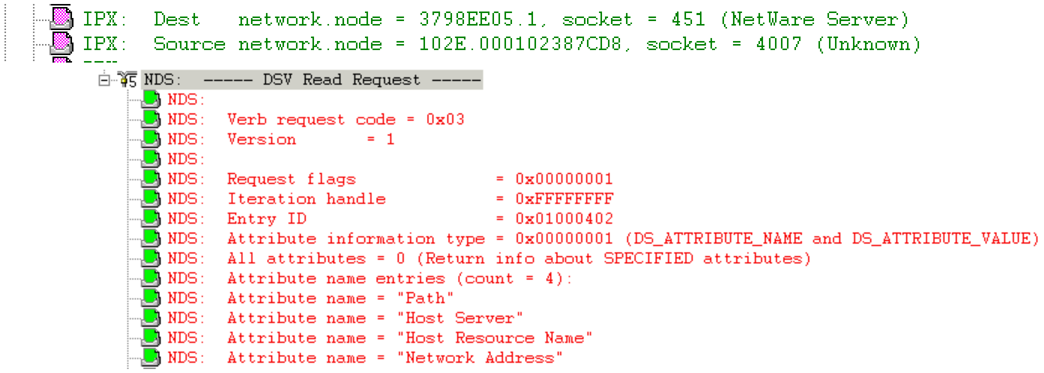


```
IPX:  Dest   network.node = 102E.000102387CD8, socket = 4007 (Unknown)
IPX:  Source network.node = 3798EE05.1, socket = 451 (NetWare Server)
NDS:  ------ DSV Resolve Name Response ------
  NDS:
  NDS:  Verb response code = 0x01
  NDS:  NDS fragment error = 0x00000000 (OK)
  NDS:
  NDS:  Tag        = 1 (Local Entry)
  NDS:  Entry ID = 0x01000402
```

Figure 6 – Frame 18.

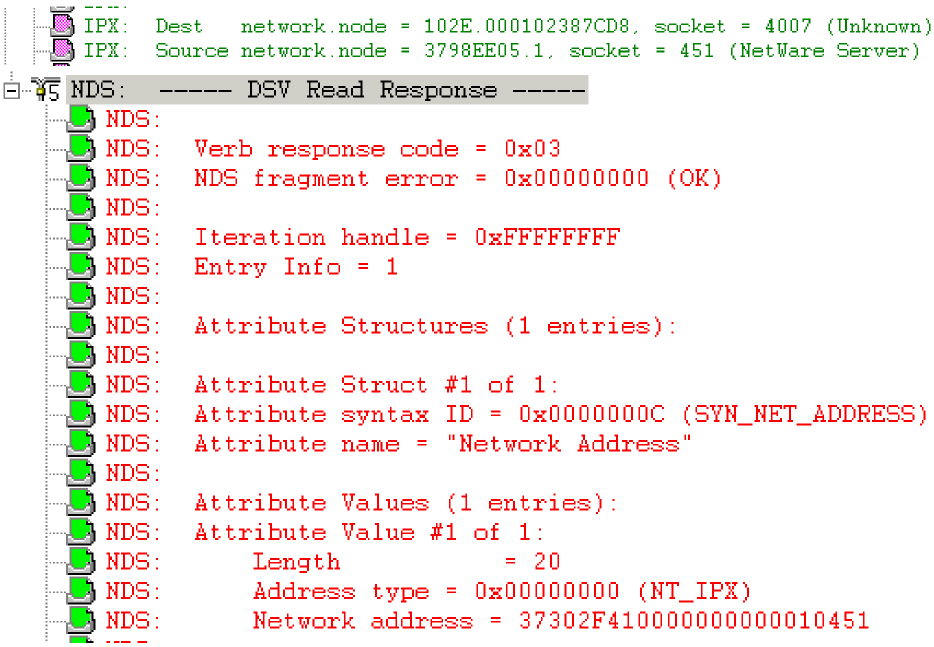
In Frame 21, (see below Fig. 7), SirCamPC sends LocalFS a "*DSV Read Request*" packet for specific attributes about the EID ‘0x01000402’. Those attributes are: Path, Host Server, Host Resource Name, and Network Address.

In Frame 22, (see below Fig. 8), LocalFS returns a "*DSV Read Response*" packet to SirCamPC which includes the network address ‘37302f410000000000010451’.



```
IPX:  Dest   network.node = 3798EE05.1, socket = 451 (NetWare Server)
IPX:  Source network.node = 102E.000102387CD8, socket = 4007 (Unknown)
NDS:  ----- DSV Read Request -----
  NDS:
  NDS:  Verb request code = 0x03
  NDS:  Version       = 1
  NDS:
  NDS:  Request flags              = 0x00000001
  NDS:  Iteration handle           = 0xFFFFFFFF
  NDS:  Entry ID                   = 0x01000402
  NDS:  Attribute information type = 0x00000001 (DS_ATTRIBUTE_NAME and DS_ATTRIBUTE_VALUE)
  NDS:  All attributes = 0 (Return info about SPECIFIED attributes)
  NDS:  Attribute name entries (count = 4):
  NDS:  Attribute name = "Path"
  NDS:  Attribute name = "Host Server"
  NDS:  Attribute name = "Host Resource Name"
  NDS:  Attribute name = "Network Address"
```

Figure 7 – Frame 21.

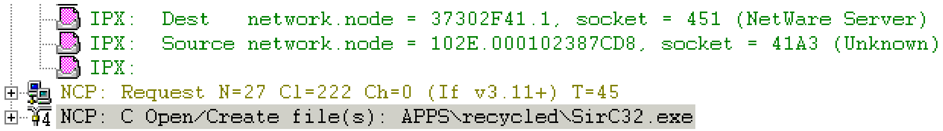


```
IPX:  Dest   network.node = 102E.000102387CD8, socket = 4007 (Unknown)
IPX:  Source network.node = 3798EE05.1, socket = 451 (NetWare Server)
NDS:  ----- DSV Read Response -----
  NDS:
  NDS:  Verb response code = 0x03
  NDS:  NDS fragment error = 0x00000000 (OK)
  NDS:
  NDS:  Iteration handle = 0xFFFFFFFF
  NDS:  Entry Info = 1
  NDS:
  NDS:  Attribute Structures (1 entries):
  NDS:
  NDS:  Attribute Struct #1 of 1:
  NDS:  Attribute syntax ID = 0x0000000C (SYN_NET_ADDRESS)
  NDS:  Attribute name = "Network Address"
  NDS:
  NDS:  Attribute Values (1 entries):
  NDS:  Attribute Value #1 of 1:
  NDS:      Length          = 20
  NDS:      Address type = 0x00000000 (NT_IPX)
  NDS:      Network address = 37302F41000000000010451
```

Figure 8 – Frame 22.

The worm has found a network share and will now try to infect it.

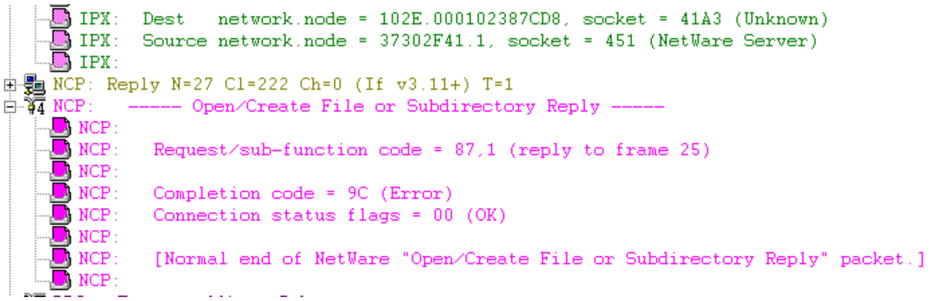
In Frame 25, (see Fig. 9), the Source network.node is still SirCamPC, but the Dest network.node has changed. It is '37302F41.1', which is the remote Novell file server (FSATL1). This is a server, which the user of the SirCamPC does not have file access rights to (RemoteFS). SirCamPC sends an "*Open/Create File or Subdirectory Request*" packet to RemoteFS. SirCamPC attempts to copy SirC32.exe (an infected executable file) to APPS\recycled\SirC32.exe.



```
  IPX:  Dest   network.node = 37302F41.1, socket = 451 (NetWare Server)
  IPX:  Source network.node = 102E.000102387CD8, socket = 41A3 (Unknown)
  IPX:
NCP: Request N=27 C1=222 Ch=0 (If v3.11+) T=45
NCP: C Open/Create file(s): APPS\recycled\SirC32.exe
```

Figure 9 – Frame 25.

In Frame 143, (see Fig. 10), the RemoteFS responds with an "*Open/Create File or Subdirectory Reply*" packet to SirCamPC indicating a 9C completion code. This is an "Invalid Path" and denies SirCamPC from copying the worm's infected file.



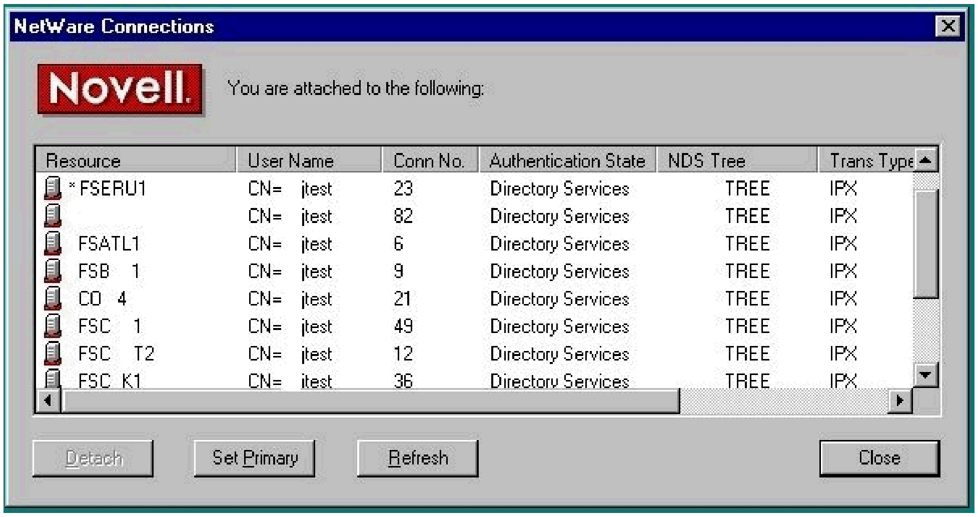
```
IPX:  Dest   network.node = 102E.000102387CD8, socket = 41A3 (Unknown)
IPX:  Source network.node = 37302F41.1, socket = 451 (NetWare Server)
IPX:
NCP: Reply N=27 C1=222 Ch=0 (If v3.11+) T=1
NCP:   ----- Open/Create File or Subdirectory Reply -----
 NCP:
 NCP:   Request/sub-function code = 87,1 (reply to frame 25)
 NCP:
 NCP:   Completion code = 9C (Error)
 NCP:   Connection status flags = 00 (OK)
 NCP:
 NCP:   [Normal end of NetWare "Open/Create File or Subdirectory Reply" packet.]
 NCP:
```

Figure 10 – Frame 143.

The first nine frames (Figs. 1 – 9) took less than .003 ms. The response from RemoteFS to SirCamPC (Fig. 10) took less than .045 ms. During that time; SirCamPC continues its search for more potential victims. SirCamPC relentlessly queries LocalFS (sending numerous DSV Resolve Name Requests) based on other objects (servers and volumes) found/known by the Novell client software.

More Sniffer Screen Shots

In order to better understand how the worm wrecked havoc on my network, I infected a PC to see the impact. I ran two additional Sniffer traces (in a controlled environment) and found new interesting data.



You are attached to the following:

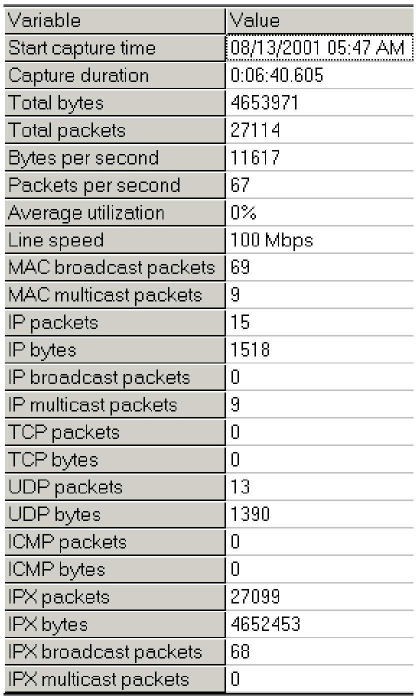
Resource	User Name	Conn No.	Authentication State	NDS Tree	Trans Type
* FSERU1	CN= jtest	23	Directory Services	TREE	IPX
	CN= jtest	82	Directory Services	TREE	IPX
FSATL1	CN= jtest	6	Directory Services	TREE	IPX
FSB 1	CN= jtest	9	Directory Services	TREE	IPX
CO 4	CN= jtest	21	Directory Services	TREE	IPX
FSC 1	CN= jtest	49	Directory Services	TREE	IPX
FSC T2	CN= jtest	12	Directory Services	TREE	IPX
FSC K1	CN= jtest	36	Directory Services	TREE	IPX

From the first controlled trace, every Novell file server within my NDS tree had been "touched" by the SirCamPC in just less than seven minutes (see Figure 11). During that time, the infected PC generated more than 4.6 megabytes (11.6 kbps) of traffic. Over twenty-seven thousand packets (frames) were produced at the

Figure 11.

rate 67 pkts/sec. See Fig. 12 for the full statistics from the trace.

On a 100 mbps LAN you would need approximately 8,620 infected PCs to totally saturate it. However, on an Ethernet non-switched LAN over 30% utilization of the bandwidth is considered over-utilized. This adds up to approximately 2,500 PCs to bring the network to a halt by a DDoS attack. And if you are running a T1 (1.54 Kbps) WAN link, you will need only 133 PCs to make use of the entire bandwidth. With smaller WAN links you can see how easily this worm can shut down a network.

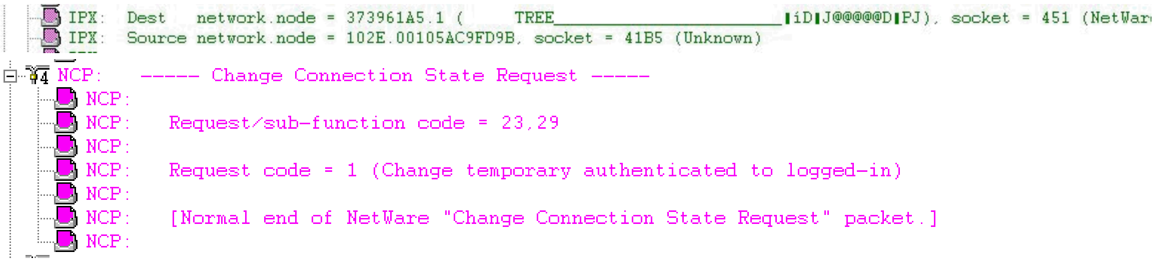


Variable	Value
Start capture time	08/13/2001 05:47 AM
Capture duration	0:06:40.605
Total bytes	4653971
Total packets	27114
Bytes per second	11617
Packets per second	67
Average utilization	0%
Line speed	100 Mbps
MAC broadcast packets	69
MAC multicast packets	9
IP packets	15
IP bytes	1518
IP broadcast packets	0
IP multicast packets	9
TCP packets	0
TCP bytes	0
UDP packets	13
UDP bytes	1390
ICMP packets	0
ICMP bytes	0
IPX packets	27099
IPX bytes	4652453
IPX broadcast packets	68
IPX multicast packets	0

Our network is composed primarily of IPX traffic and therefore less than 1.5 Kbytes of the traffic was non-IPX. This indicates that the worm found all of the Novell shares (file servers and volumes) via the Novell client. When a user authenticates to the tree they have rights to only one file server and its volumes, though they do have browse rights to objects within the entire tree.

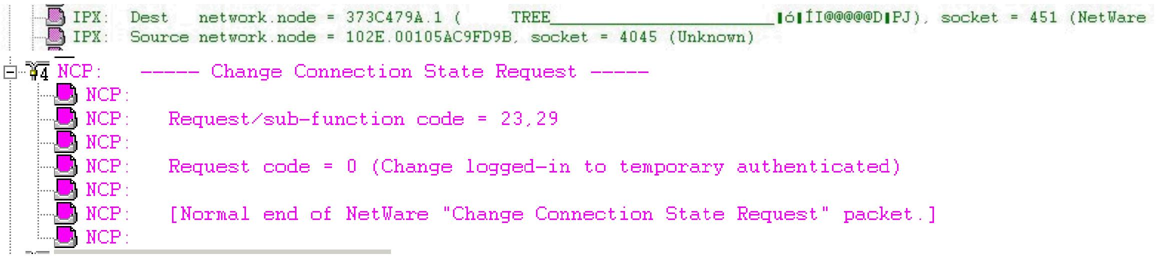
Throughout the six and a-half minute capture, the infected PC makes a “Change Connection State Request” from ‘temporary authenticated ‘ to ‘logged-in’ (see below Fig. 13). Then, after the barrage of “Copy/Create File Request” all of which failed, the infected PC issues another “Change Connection State Request” from ‘logged-in’ to ‘temporary authenticated’ (see below Fig. 14). The time elapsed from frames 3890 to 5030 was 8.633 ms and generated almost 160 Kb to Dest network.node 373961a5.1. Then it moves on to find its next victim.

Figure 12.



```
IPX:  Dest   network.node = 373961A5.1 (     TREE_______________________▌iD▌J@@@@@D▌PJ), socket = 451 (NetWar
IPX:  Source network.node = 102E.00105AC9FD9B, socket = 41B5 (Unknown)
NCP:   ----- Change Connection State Request -----
  NCP:
  NCP:   Request/sub-function code = 23,29
  NCP:
  NCP:   Request code = 1 (Change temporary authenticated to logged-in)
  NCP:
  NCP:   [Normal end of NetWare "Change Connection State Request" packet.]
  NCP:
```

Figure 13 – Frame 3890



```
IPX:  Dest   network.node = 373C479A.1 (     TREE_______________________▌ó▌ÍI@@@@@D▌PJ), socket = 451 (NetWare
IPX:  Source network.node = 102E.00105AC9FD9B, socket = 4045 (Unknown)
NCP:   ----- Change Connection State Request -----
  NCP:
  NCP:   Request/sub-function code = 23,29
  NCP:
  NCP:   Request code = 0 (Change logged-in to temporary authenticated)
  NCP:
  NCP:   [Normal end of NetWare "Change Connection State Request" packet.]
  NCP:
```

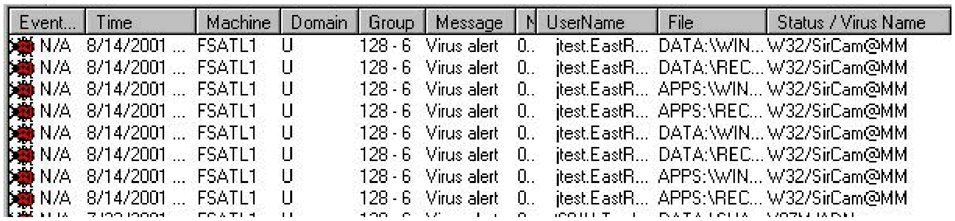
Figure 14 – Frame 5030

During that time there numerous unsuccessful attempts to copy/create the infected file (run32dll.exe, run32.exe, etc.) to the file server.

For the second controlled trace I created directories on a server across the WAN. I granted the user of the infected PC full file access rights to the following directories and files.

1 APPS\RECYCLED
2 APPS\WINDOWS
3 DATA\RECYCLED
4 DATA\WINDOWS
5 SYS\RECYCLED
6 SYS\WINDOWS

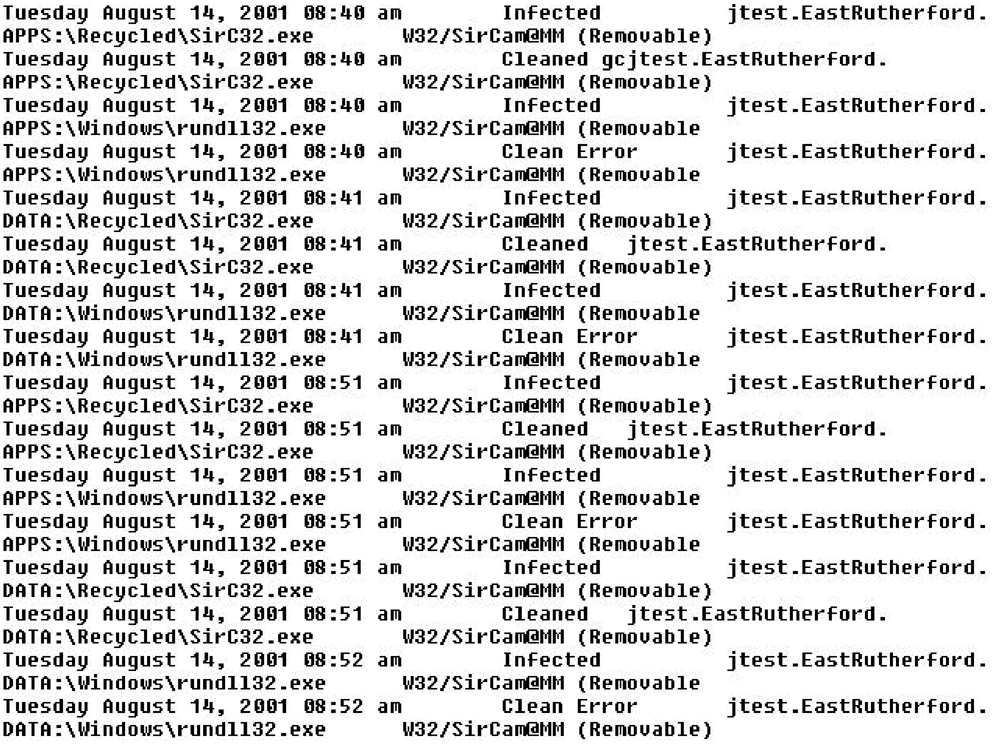
I also created an AUTOEXEC.BAT file on the APPS, DATA and SYS volumes.



Event...	Time	Machine	Domain	Group	Message	N	UserName	File	Status / Virus Name
N/A	8/14/2001 ...	FSATL1	U	128 - 6	Virus alert	0..	jtest.EastR...	DATA:\WIN...	W32/SirCam@MM
N/A	8/14/2001 ...	FSATL1	U	128 - 6	Virus alert	0..	jtest.EastR...	DATA:\REC...	W32/SirCam@MM
N/A	8/14/2001 ...	FSATL1	U	128 - 6	Virus alert	0..	jtest.EastR...	APPS:\WIN...	W32/SirCam@MM
N/A	8/14/2001 ...	FSATL1	U	128 - 6	Virus alert	0..	jtest.EastR...	APPS:\REC...	W32/SirCam@MM
N/A	8/14/2001 ...	FSATL1	U	128 - 6	Virus alert	0..	jtest.EastR...	DATA:\WIN...	W32/SirCam@MM
N/A	8/14/2001 ...	FSATL1	U	128 - 6	Virus alert	0..	jtest.EastR...	DATA:\REC...	W32/SirCam@MM
N/A	8/14/2001 ...	FSATL1	U	128 - 6	Virus alert	0..	jtest.EastR...	APPS:\WIN...	W32/SirCam@MM
N/A	8/14/2001 ...	FSATL1	U	128 - 6	Virus alert	0..	jtest.EastR...	APPS:\REC...	W32/SirCam@MM

Though the user of the infected PC had full rights to these directories and files, McAfee NetShield protected the server Figure 15.

and the files were not copied because the (see Figs. 15 & 16). Fig. 15 is a shot from McAfee TVD Management Edition Console and Fig. 16 is from the activity.txt file on the Novell server)



```
Tuesday August 14, 2001 08:40 am        Infected          jtest.EastRutherford.
APPS:\Recycled\SirC32.exe       W32/SirCam@MM (Removable)
Tuesday August 14, 2001 08:40 am        Cleaned gcjtest.EastRutherford.
APPS:\Recycled\SirC32.exe       W32/SirCam@MM (Removable)
Tuesday August 14, 2001 08:40 am        Infected          jtest.EastRutherford.
APPS:\Windows\rundll32.exe      W32/SirCam@MM (Removable
Tuesday August 14, 2001 08:40 am        Clean Error       jtest.EastRutherford.
APPS:\Windows\rundll32.exe      W32/SirCam@MM (Removable
Tuesday August 14, 2001 08:41 am        Infected          jtest.EastRutherford.
DATA:\Recycled\SirC32.exe       W32/SirCam@MM (Removable)
Tuesday August 14, 2001 08:41 am        Cleaned   jtest.EastRutherford.
DATA:\Recycled\SirC32.exe       W32/SirCam@MM (Removable)
Tuesday August 14, 2001 08:41 am        Infected          jtest.EastRutherford.
DATA:\Windows\rundll32.exe      W32/SirCam@MM (Removable
Tuesday August 14, 2001 08:41 am        Clean Error       jtest.EastRutherford.
DATA:\Windows\rundll32.exe      W32/SirCam@MM (Removable
Tuesday August 14, 2001 08:51 am        Infected          jtest.EastRutherford.
APPS:\Recycled\SirC32.exe       W32/SirCam@MM (Removable)
Tuesday August 14, 2001 08:51 am        Cleaned   jtest.EastRutherford.
APPS:\Recycled\SirC32.exe       W32/SirCam@MM (Removable)
Tuesday August 14, 2001 08:51 am        Infected          jtest.EastRutherford.
APPS:\Windows\rundll32.exe      W32/SirCam@MM (Removable
Tuesday August 14, 2001 08:51 am        Clean Error       jtest.EastRutherford.
APPS:\Windows\rundll32.exe      W32/SirCam@MM (Removable
Tuesday August 14, 2001 08:51 am        Infected          jtest.EastRutherford.
DATA:\Recycled\SirC32.exe       W32/SirCam@MM (Removable)
Tuesday August 14, 2001 08:51 am        Cleaned   jtest.EastRutherford.
DATA:\Recycled\SirC32.exe       W32/SirCam@MM (Removable)
Tuesday August 14, 2001 08:52 am        Infected          jtest.EastRutherford.
DATA:\Windows\rundll32.exe      W32/SirCam@MM (Removable
Tuesday August 14, 2001 08:52 am        Clean Error       jtest.EastRutherford.
DATA:\Windows\rundll32.exe      W32/SirCam@MM (Removable)
```

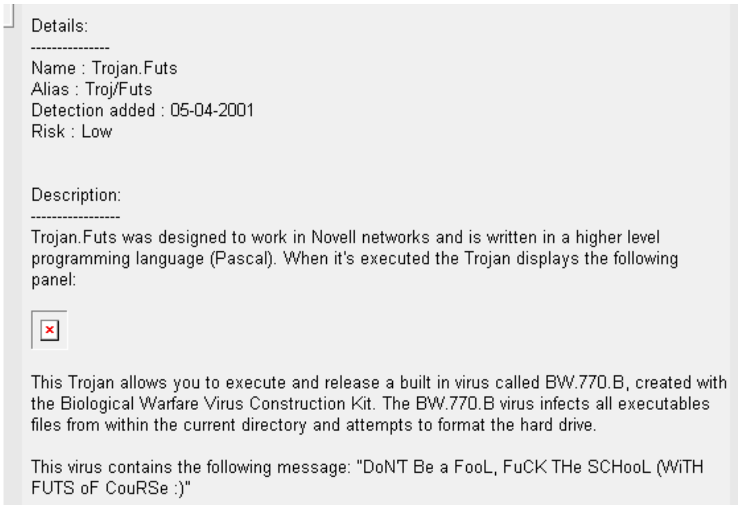
Figure 16.

Is This Really a First for Novell?

Well, according to both McAfee and Sophos, two large antiviral companies, there are no viruses that infect the Novell OS. See the following URL's for their answers.

http://vil.nai.com/vil/alphar.asp

http://www.sophos.com/virusinfo/analyses/



Details:

Name : Trojan.Futs
Alias : Troj/Futs
Detection added : 05-04-2001
Risk : Low

Description:

Trojan.Futs was designed to work in Novell networks and is written in a higher level programming language (Pascal). When it's executed the Trojan displays the following panel:

This Trojan allows you to execute and release a built in virus called BW.770.B, created with the Biological Warfare Virus Construction Kit. The BW.770.B virus infects all executables files from within the current directory and attempts to format the hard drive.

This virus contains the following message: "DoN'T Be a FooL, FuCK THe SCHooL (WiTH FUTS oF CouRSe :)"

Two more antiviral manufacturers, Central Command and Symantec, list a virus that infects Novell - Trojan.Futs. Trojan.Futs was discovered earlier this year but has not been considered a serious threat. See Fig. 11, for Command Central's information.

Figure 11.

See the following URL for Command Central for a description of the Trojan.Futs.

http://support.centralcommand.com/cgi-bin/command.cfg/php/enduser/std_adp.php?p_sid=BDPxn*Uf&p_lva=&p_refno=010405-000009&p_created=986488763&p_sp=cF9ncmlkc29ydD0mcF9yb3dfY250PTEmcF9zZWFyY2hfdGV4dD1mdXRzJnBfc2VhcmNoX3R5cGU9MyZwX3Byb2RfbHZsMT1_YW55fiZwX2NhdF9sdmwxPTQmcF9zb3J0X2J5PWRmbHQmcF9wYWdlPTE*&p_li=

What Does This Mean?

This means that although the Novell operating system is so far immune from any major virus or worm, a Novell network is still susceptible to attack and spreading infection if proper care is not taken in the design and maintenance thereof.

What Can We Do?

To help cease the spread of a network aware virus on a Novell network and to reduce the heavy traffic load created by infected PCs searching for new prey, do not allow users to browse outside of their own local context unless absolutely necessary. Make sure that the latest virus definition files are installed on all of your files servers (Novell and others). And, unless required, don't use directory names like those used by the Microsoft products; i.e. Windows or Recycled.

List of References:

[1] Robert Lemos. "Fast-spreading code is weapon of choice for Net vandals" Special to CNET News.com. March 15, 2001, 4:00 a.m. PT
http://news.cnet.com/news/0-1003-201-5125673-0.html

[2] Robert Lemos. "Fast-spreading code is weapon of choice for Net vandals" Special to CNET News.com. March 15, 2001, 4:00 a.m. PT
http://news.cnet.com/news/0-1003-201-5125673-0.html

[3] Roman Danyliw, Chad Dougherty, and Allen Householder. "CERT® Advisory CA-2001-22 W32/Sircam Malicious Code" July 25, 2001 CERT/CC
http://www.cert.org/advisories/CA-2001-22.html

[4] Gergely Erdelyi, and Alexey Podrezov. F-Secure Corporation. July 18-23, 2001
http://www.datafellows.com/v-descs/sircam.shtml

[5] Michelle Delio. "Love Bug, SirCam Neck and Neck" Wired News. July 23, 2001, 11:50 a.m.
http://www.wired.com/news/technology/0,1282,45476,00.html

[6] Sam Costello. "Report: Sircam hits FBI cybersecurity group" IDG News Service. July 25, 2001
http://www.nwfusion.com/news/2001/0725sircam.html

[7] Ian Fried - Staff Writer, CNET News.com. "Sircam worm still spreading documents" August 2, 2001, 3:00 p.m. PT
http://news.cnet.com/news/0-1003-200-6759035.html

[8] Michelle Delio. "SirCam: Devious, But Not Sinister" Wired News. July 25, 2001, 2:00 a.m.
http://www.wired.com/news/technology/0,1282,45506,00.html