

AFFIDAVIT OF NATHANIEL E FRANK-WHITE

1. I am a Records Request Processor at the Internet Archive. I make this declaration of my own personal knowledge.

2. The Internet Archive is a website that provides access to a digital library of Internet sites and other cultural artifacts in digital form. Like a paper library, we provide free access to researchers, historians, scholars, and the general public. The Internet Archive has partnered with and receives support from various institutions, including the Library of Congress.

3. The Internet Archive has created a service known as the Wayback Machine. The Wayback Machine makes it possible to browse more than 450 billion pages stored in the Internet Archive's web archive. Visitors to the Wayback Machine can search archives by URL (i.e., a website address). If archived records for a URL are available, the visitor will be presented with a display of available dates. The visitor may select one of those dates, and begin browsing an archived version of the Web. Links on archived files in the Wayback Machine point to other archived files (whether HTML pages or other file types), if any are found for the URL indicated by a given link. For instance, the Wayback Machine is designed such that when a visitor clicks on a hyperlink on an archived page that points to another URL, the visitor will be served the archived file found for the hyperlink's URL with the closest available date to the initial file containing the hyperlink.

4. The archived data made viewable and browsable by the Wayback Machine is obtained by use of web archiving software that automatically stores copies of files available via the Internet, each file preserved as it existed at a particular point in time.

5. The Internet Archive assigns a URL on its site to the archived files in the format http://web.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL] aka an "extended URL". Thus, the extended URL http://web.archive.org/web/19970126045828/http://www.archive.org/ would be the URL for the record of the Internet Archive home page HTML file (http://www.archive.org/) archived on January 26, 1997 at 4:58 a.m. and 28 seconds (1997/01/26 at 04:58:28). The date indicated by an extended URL applies to a preserved instance of a file for a given URL, but not necessarily to any other files linked therein. Thus, in the case of a page constituted by a primary HTML file and other separate files (e.g., files with images, audio, multimedia, design elements, or other embedded content) linked within that primary HTML file, the primary HTML file and the other files will each have their own respective extended URLs and may not have been archived on the same dates.

6. Attached hereto as Exhibit A are true and accurate copies of screenshots of the Internet Archive's records of the archived files for the URLs and the dates specified in the attached coversheet of each printout.

PAN EX1030
PAN v. Centripetal
IPR2023-01353
Page 00001



7. Attached hereto as Exhibit B are true and accurate copies of the Internet Archive's records of the archived files for the URLs and the dates specified in the attached coversheet of each file.

8. I declare under penalty of perjury that the foregoing is true and correct.

DATE: 12/12/2022

Please see attached All Purpose Jurat form for additional Notary Events

Nathaniel Frank-White

Nathaniel E Frank-White

JURAT ATTACHMENT

A notary public or other officer completing this certificate verifies only the identity of the individual who signed the document to which this certificate is attached, and not the truthfulness, accuracy, or validity of that document.

STATE OF Texas }

COUNTY OF Harris }

The foregoing instrument was subscribed and sworn before me this date of 12/12/2022, by Nathaniel Frank-White

This notarial act was an online notarization.



(Notary Seal)

Notary's Signature Ana Laura Salazar Uribe

Registration No.: 131757026

Commission Expiration Date: October 11, 2026

EXHIBIT A

https://web.archive.org/web/20101117075949/http://rules.emergingthreats.net/blockrules/emerging-drop.rules

http://rules.emergingthreats.net/blockrules/emerging-drop.rules
Go
28 captures
17 Nov 2010 - 12 Nov 2020
OCT NOV JUN
17
2009 2010 2011
About this capture

```
#
# $Id: emerging-drop.rules $
# Emerging Threats Spamhaus DROP List rules.
#
# Rules to block Spamhaus DROP listed networks (www.spamhaus.org)
#
# More information available at www.emergingthreats.net
#
# Please submit any feedback or ideas to emerging@emergingthreats.net or the emerging-sigs mailing list
#
#************************************************************
#
#  Copyright (c) 2003-2010, Emerging Threats
#  All rights reserved.
#
#  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
#  following conditions are met:
#
#  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following
#    disclaimer.
#  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
#    following disclaimer in the documentation and/or other materials provided with the distribution.
#  * Neither the name of the nor the names of its contributors may be used to endorse or promote products derived
#    from this software without specific prior written permission.
#
#  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES,
#  INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
#  DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
#  SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
#  SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
#  WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
#  USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#

#  VERSION 2084


#  Generated 2010-11-16 00:05:01 EDT

alert tcp
[116.199.128.0/19,121.46.64.0/18,122.202.96.0/19,128.168.0.0/16,128.199.0.0/16,132.232.0.0/16,132.240.0.0/16,134.33.0.0/16,136.228.0.0/16,138.252.0.0/16,138.43.0
.0/16,139.167.0.0/16,14.1.96.0/19,14.102.160.0/19,140.170.0.0/16,143.135.0.0/16,143.49.0.0/16,143.95.0.0/16,148.178.0.0/16,148.248.0.0/16] any <> $HOME_NET any
(msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type limit, track by_src, seconds 3600,
count 1; classtype:misc-attack; sid:2400000; rev:2084;)
alert tcp
[157.226.0.0/16,159.223.0.0/16,167.28.0.0/16,167.97.0.0/16,168.151.0.0/16,170.106.0.0/16,170.120.0.0/16,170.67.0.0/16,170.75.0.0/16,173.225.112.0/20,174.138.144.
0/20,180.210.190.0/23,188.130.250.0/23,188.170.192.0/21,188.170.216.0/21,188.170.243.0/24,188.212.0.0/20,188.229.13.0/24,188.229.16.0/21,188.229.40.0/21] any <>
$HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type limit, track by_src,
seconds 3600, count 1; classtype:misc-attack; sid:2400001; rev:2084;)
alert tcp
[188.95.159.0/24,190.109.64.0/19,192.101.200.0/21,192.101.240.0/21,192.101.248.0/23,192.112.112.0/20,192.160.44.0/24,192.223.64.0/18,192.26.25.0/24,192.31.212.0/
23,192.43.153.0/24,192.43.154.0/23,192.43.156.0/22,192.43.160.0/24,192.43.175.0/24,192.43.176.0/21,192.43.184.0/24,192.67.16.0/24,192.86.85.0/24,193.104.106.0/24
] any <> $HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type limit, track
by_src, seconds 3600, count 1; classtype:misc-attack; sid:2400002; rev:2084;)
alert tcp
[193.104.253.0/24,193.104.27.0/24,193.104.34.0/24,193.104.41.0/24,193.104.94.0/24,193.105.0.0/24,193.105.132.0/24,193.105.141.0/24,193.105.154.0/24,193.105.174.0
/24,193.105.184.0/24,193.105.189.0/24,193.105.207.0/24,193.105.245.0/24,193.106.32.0/22,193.110.136.0/24,193.142.244.0/24,193.148.47.0/24,193.16.100.0/24,193.169
.234.0/23] any <> $HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type
limit, track by_src, seconds 3600, count 1; classtype:misc-attack; sid:2400003; rev:2084;)
alert tcp
[193.23.126.0/24,193.23.126.30/32,193.27.232.0/23,193.27.246.0/23,193.41.38.0/24,193.43.134.0/24,193.46.211.0/24,194.0.221.0/24,194.110.160.0/22,194.116.146.0/23
,194.126.193.0/24,194.126.251.0/24,194.140.229.0/24,194.143.130.0/23,194.146.204.0/22,194.165.4.0/23,194.28.44.0/22,194.8.250.0/23,194.8.74.0/23,195.114.8.0/23]
any <> $HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type limit, track
by_src, seconds 3600, count 1; classtype:misc-attack; sid:2400004; rev:2084;)
alert tcp
[195.2.212.0/23,195.225.176.0/22,195.226.197.0/24,195.226.220.0/24,195.234.159.0/24,195.238.242.0/24,195.5.161.0/24,195.5.168.0/24,195.54.170.0/23,195.74.88.0/23
,195.78.108.0/23,195.78.122.0/23,195.88.144.0/23,195.88.190.0/23,195.88.226.0/23,195.93.184.0/23,195.93.208.0/23,195.95.151.0/24,195.95.155.0/24,196.1.176.0/20]
any <> $HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type limit, track
by_src, seconds 3600, count 1; classtype:misc-attack; sid:2400005; rev:2084;)
alert tcp
[198.186.25.0/24,198.204.0.0/21,199.120.163.0/24,199.165.32.0/19,199.166.200.0/22,199.230.64.0/19,199.230.96.0/21,199.245.138.0/24,199.60.102.0/24,200.106.128.0/
20,200.115.112.0/21,200.115.96.0/20,200.123.224.0/20,200.124.160.0/21,200.22.0.0/16,200.50.192.0/19,201.71.0.0/20,202.59.236.0/24,203.19.101.0/24,203.31.88.0/23]
any <> $HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type limit, track
by_src, seconds 3600, count 1; classtype:misc-attack; sid:2400006; rev:2084;)
alert tcp
[204.187.160.0/19,204.187.224.0/20,204.187.240.0/21,204.187.248.0/22,204.187.252.0/23,204.187.254.0/24,204.194.184.0/21,204.236.0.0/19,204.28.104.0/21,204.52.255
.0/24,204.89.224.0/24,205.235.64.0/20,205.236.189.0/24,206.197.175.0/24,206.197.176.0/24,206.197.177.0/24,206.197.28.0/24,206.197.29.0/24,206.201.48.0/20,207.189
.0.0/19] any <> $HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type
limit, track by_src, seconds 3600, count 1; classtype:misc-attack; sid:2400007; rev:2084;)
alert tcp
[209.236.96.0/20,213.109.176.0/20,213.109.208.0/20,213.109.96.0/22,216.212.192.0/19,216.47.32.0/20,218.75.128.0/19,41.221.112.0/20,46.51.100.0/24,46.51.101.0/24,
46.51.102.0/23,58.83.12.0/22,58.83.8.0/22,62.122.32.0/21,62.122.72.0/23,62.182.152.0/21,64.112.0.0/17,64.112.128.0/18,64.15.0.0/20,64.20.80.0/20] any <>
$HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type limit, track by_src,
seconds 3600, count 1; classtype:misc-attack; sid:2400008; rev:2084;)
alert tcp
[67.211.208.0/20,67.213.128.0/20,67.218.208.0/20,69.8.176.0/20,72.13.16.0/20,72.2.176.0/20,72.50.192.0/19,78.155.220.0/23,78.31.184.0/21,79.110.144.0/20,79.110.1
6.0/20,79.110.160.0/20,79.110.176.0/20,79.110.48.0/20,79.135.128.0/19,79.174.0.0/19,85.202.160.0/20,85.202.192.0/20,85.255.112.0/20,86.55.243.0/24] any <>
$HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type limit, track by_src,
seconds 3600, count 1; classtype:misc-attack; sid:2400009; rev:2084;)
alert tcp
[91.197.96.0/22,91.198.109.0/24,91.198.127.0/24,91.198.217.0/24,91.199.112.0/24,91.199.123.0/24,91.200.164.0/22,91.200.248.0/22,91.201.124.0/22,91.201.196.0/22,9
1.201.212.0/22,91.201.236.0/22,91.201.4.0/22,91.202.52.0/22,91.203.92.0/22,91.205.40.0/22,91.207.116.0/23,91.208.0.0/24,91.208.162.0/24,91.208.228.0/24] any <>
$HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type limit, track by_src,
```

seconds 3600, count 1; classtype:misc-attack; sid:2400010; rev:2084;)
alert tcp
[91.209.48.0/24,91.209.58.0/24,91.210.172.0/22,91.211.224.0/22,91.211.64.0/22,91.211.88.0/22,91.212.107.0/24,91.212.123.0/24,91.212.132.0/24,91.212.163.0/24,91.212.201.0/24,91.212.220.0/24,91.212.41.0/24,91.212.45.0/24,91.212.65.0/24,91.213.121.0/24,91.213.126.0/24,91.213.174.0/24,91.213.217.0/24,91.213.29.0/24] any <>
$HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type limit, track by_src,
seconds 3600, count 1; classtype:misc-attack; sid:2400011; rev:2084;)
alert tcp
[91.216.122.0/24,91.216.232.0/24,91.216.3.0/24,91.216.73.0/24,91.217.249.0/24,93.120.32.0/19,93.168.18.0/23,93.168.20.0/23,93.168.22.0/23,93.168.24.0/23,93.174.164.0/24,93.175.240.0/20,93.188.160.0/21,93.188.162.0/23,94.126.176.0/21,94.130.0.0/15,94.154.0.0/18,94.154.128.0/18,94.154.64.0/18,94.158.240.0/20] any <>
$HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type limit, track by_src,
seconds 3600, count 1; classtype:misc-attack; sid:2400012; rev:2084;)
alert tcp [94.232.248.0/21,94.63.240.0/24,94.63.247.0/24,95.215.140.0/22,95.215.192.0/22,95.216.0.0/15,95.64.109.0/24,95.64.98.0/23,96.45.160.0/20,96.47.48.0/20]
any <> $HOME_NET any (msg:"ET DROP Spamhaus DROP Listed Traffic Inbound"; flags:S; reference:url,www.spamhaus.org/drop/drop.lasso; threshold: type limit, track
by_src, seconds 3600, count 1; classtype:misc-attack; sid:2400013; rev:2084;)

https://web.archive.org/web/20040103041128/http://www.spamhaus.org/xbl/

INTERNET ARCHIVE WayBackMachine
http://www.spamhaus.org/xbl/
Go
DEC JAN FEB
03
2003 2004 2005
1,344 captures
5 Jan 2004 - 7 Dec 2022
About this capture

SPAMHAUS The Spamhaus Project

Spamhaus | SBL | XBL | ROKSO

Home | News | FAQs | Definition of 'Spam'

IP Lookup

Enter an IP address:

Lookup

Information on this page is for ISPs and mail server administrators.

DNS-based Blocklists (DNSBLs) such as the Spamhaus SBL and XBL are designed only for use on mail servers.

XBL Advisory

Exploits Block List

The Spamhaus Exploits Block List (XBL) is a realtime database of IP addresses of illegal 3rd party exploits, including open proxies (HTTP, socks, AnalogX, wingate, etc), worms/viruses with built-in spam engines, and other types of trojan-horse exploits.

Incorporates CBL data

The XBL wholly incorporates the highly-trusted CBL (Composite Block List) from cbl.abuseat.org, therefore mail servers already using cbl.abuseat.org should NOT also use xbl.spamhaus.org or you will be making 'double' queries to basically the same data source and only one DNSBL will appear to work, the other will appear to not catch anything.

How To Use the XBL

The Exploits Block List can be used by all modern mail servers, by setting your mail server's anti-spam DNSBL feature (sometimes called "Blacklist DNS Servers" or "RBL servers") to query xbl.spamhaus.org. Use of the XBL is free.

We advise you to use xbl.spamhaus.org together with sbl.spamhaus.org, as the SBL and XBL block different types of spam (SBL stops spam from direct spam sources, XBL stops spam from proxies and other exploits)

DNSBL	Zone to query	Returns	Maintainer's URL
email protected by SBL Advisory	sbl.spamhaus.org	127.0.0.2	http://www.spamhaus.org/SBL
email protected by XBL Advisory	xbl.spamhaus.org	127.0.0.2	http://www.spamhaus.org/XBL

For FAQs and additional information please see the CBL pages at http://cbl.abuseat.org/

Zone Transfers (rsync) for ISPs

Large ISPs or large corporate mail services who would be likely to make over 400,000 queries per day should use zone transfers to a local DNS server on your network. Zone transfers of XBL are available as rsync only, please contact us (stating clearly who you are, what size of network you run and what IP address you will be using). The only restriction is that you can not redistribute the zone or its data under a different name (international organizations can however redistribute the zone to their subsidiaries).

The SPAMHAUS Project HOME

email protected by SBL Advisory email protected by XBL Advisory ROKSO

https://web.archive.org/web/20130112044607/http:/rules.emergingthreats.net/



There are the Emerging Threats.net Open rulesets.

More information available at http://www.emergingthreatspro.com and http://www.emergingthreats.net

Name	Last modified	Size	Description
blockrules/	10-Jan-2013 23:30	-	
changelogs/	11-Jan-2013 19:59	-	
fwrules/	11-Nov-2010 10:40	-	
open-nogpl/	25-Sep-2012 20:54	-	
open/	25-Sep-2012 20:54	-	
projects/	17-Jan-2011 12:34	-	
research/	05-Oct-2012 21:48	-	
version.txt	11-Jan-2013 19:59	5	

https://web.archive.org/web/20120106080052/http://doc.emergingthreats.net/

http://doc.emergingthreats.net/ Go DEC JAN MAY 06 2010 2012 2013
90 captures

Emerging Threats Rule Documentation Wiki

This wiki contains all current rules, added as each is put into the main tarball and cvs repository. The rule author, if available, is primarily responsible for the documentation of a rule, however the entire community is encouraged and welcomed to contribute or document any rule. You may attach pcaps, packet text, and even code samples to any entry relevant. This is particularly useful for future troubleshooting. Please document if possible where the sample was captured. If you have a sample that's not suitable for posting publicly please contact emerging@emergingthreats.net and it can be archived privately, available to any vetted researcher.

UserDocs | AllRulesets | EmergingFAQ | AllProjects

Start Here

Want some guidance on using the Emerging Threats Rulesets for the first time? NewUserGuide

Some tips on writing rules? SnortSigs101

Tips on what to add to your local ruleset that's not in the main rulesets: WhatEverySnortUserShouldDo?

OpenInfosec

Feature development discussions around the Open Information Security Foundation's new projects! (http://www.openinfosecfoundation.org)

SidReporter

Try out SidReporter, the newest project at Emerging Threats!! Help improve the accuracy and reliability of the Emerging Threats Rulesets anonymously!

Last 10 Signature Documentation Changes

Results from Main web retrieved at 08:00 (GMT)

2012942 NEW - 06 Jan 2012 - 00:54 TWikiGuest
alert tcp $HOME_NET any $EXTERNAL_NET $HTTP_PORTS (msg:"ET CURRENT EVENTS Phoenix Exploit Kit Printf.pdf"; flow:established,to_server; content:"/printf.pdf"; fast...

2012941 NEW - 06 Jan 2012 - 00:54 TWikiGuest
alert tcp $HOME_NET any $EXTERNAL_NET $HTTP_PORTS (msg:"ET CURRENT EVENTS Phoenix Exploit Kit Newplayer.pdf"; flow:established,to_server; content:"/newplayer.pdf ...

2002080 NEW - 06 Jan 2012 - 00:54 TWikiGuest
alert tcp $HOME_NET any $EXTERNAL_NET $HTTP_PORTS (msg:"ET MALWARE MySearch Products Spyware User Agent (MySearch)"; flow:established,to_server; content:"User Agent ...

2008754 NEW - 06 Jan 2012 - 00:54 TWikiGuest
alert tcp $EXTERNAL_NET $HTTP_PORTS $HOME_NET any (msg:"ET MALWARE Possible Rar'd Malware sent when remote host claims to send an Image"; flow:established,from ...

2011184 NEW - 04 Jan 2012 - 23:20 TWikiGuest
alert tcp $EXTERNAL_NET $HTTP_PORTS $HOME_NET any (msg:"ET TROJAN Phoenix Exploit Kit VBscript download"; flow:established,to_client; content:"Createobject(StrReverse ...

2003496 NEW - 04 Jan 2012 - 22:59 TWikiGuest
alert tcp $HOME_NET any $EXTERNAL_NET $HTTP_PORTS (msg:"ET MALWARE AskSearch Toolbar Spyware User Agent (AskBar)"; flow:to_server,established; content:" 3b AskBar ...

2014082 NEW - 02 Jan 2012 - 22:48 TWikiGuest
alert tcp $EXTERNAL_NET any $HTTP_SERVERS $HTTP_PORTS (msg:"ET WEB_SPECIFIC_APPS SourceBans ajaxargs Parameter Local File Inclusion Attempt"; flow:established,to ...

2014070 NEW - 02 Jan 2012 - 22:48 TWikiGuest
alert tcp $HOME_NET any $EXTERNAL_NET $HTTP_PORTS (msg:"ET TROJAN Trojan Downloader.Bancos Reporting"; flow:established,to_server; content:"GET"; http_method; content ...

2002005 NEW - 02 Jan 2012 - 22:48 TWikiGuest
alert tcp $HOME_NET any $EXTERNAL_NET $HTTP_PORTS (msg:"ET USER_AGENTS Better Internet Spyware User Agent (poller)"; flow:.to_server,established; content:"User ...

2010119 NEW - 31 Dec 2011 - 00:58 TWikiGuest
#alert tcp $EXTERNAL_NET any $HTTP_SERVERS $HTTP_PORTS (msg:"ET WEB_SERVER xp cmdshell Attempt in Cookie"; flow:established,to_server; content:"xp_cmdshell"; nocase ...

Number of topics: **10**

All additions will be reviewed by the documentation team at Emerging Threats, a volunteer group. Please report any inaccuracies or wikispam to emerging@emergingthreats.net.

To post please register. -- Registration

Follow documentation updates via WebRss or WebAtom

Conventions

All rules are available by accessing the following URL format: http://doc.emergingthreats.net/SID

i.e. http://docs.emergingthreats.net/2003434

As a rule is changed the new revision will automatically be placed above the old rule and old comments with an Auto-Added timestamp. This should allow a conversation to be relevant to the revision of the rule at the time. Please post "Yes, that fixed it" comments if a new revision fixes an older issue.

Within each signature entry there is a form to place a comment, suitable for short entries or questions about a rule. For larger posts or formal documentation please use the edit function and place the information below other content. You can use most html tags, recommend using PRE tags with code or packet text to keep it formatted as intended.

Signature authors are informally responsible for initial documentation where necessary. However ANY user may post information they have to contribute, and please do.

Documentation need not be formal. Links to POC code, vulnerability alerts, even mailing list conversations may be added to the rule's documentation. More information is definitely best. The Emerging Documentation team will review and reformat things as required over time.

See the EmergingFAQ

Navigation
WebHome

- AboutEmergingThreats
- AllProjects
- CompromisedHosts
- EmergingFAQ
- GeneralFAQ
- MalwareDocs
- OpenInfosec
- RuleChanges
- RulesBySid
- SignaturesNeeded
- SnortSam
- SubmitSamples
- UserDocs
- WebBasedResearchTools
- WebPreferences
- WebStatistics

Main Utilities

- WelcomeGuest - for first time visitors
- Registration - become a TWiki user
- Search - advanced search
- WebTopicList - all topics in alphabetical order
- WebChanges - recent topic changes in this web
- WebNotify - subscribe to an e-mail alert sent when topics change
- WebRss, WebAtom - RSS and ATOM news feeds of topic changes [http://www.merchantos.com/features/ cloud-based-point-of-sale]

Edit | Attach | Printable | Raw View | Backlinks: Web, All Webs | History: r63 < r62 < r61 < r60 < r59 | More topic actions

Copyright © by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback

TWiki

https://web.archive.org/web/20040627074240/http://www.bleedingsnort.com/

INTERNET ARCHIVE WayBackMachine http://www.bleedingsnort.com/ Go MAY JUN 27 2004 SEP 2003 2005
354 captures
24 Jun 2004 - 7 Dec 2023
About this capture

Current Bleeding Snort Ruleset:

bleeding.rules

bleeding.rules.tar.gz

bleeding-sid-msg-map.txt

Updated every 5 minutes from CVS

CVS Web Interface available Here

The general charter of this project is twofold:

1. To be a clearinghouse for rules for up to the minute threats and ideas. The rules here will be prone to false positives and sometimes not work as expected. But you will find the rules people are experimenting with for each threat, updated VERY often. If you find a flaw in a rule or have a new rule you're using email bleeding@bleedingsnort.com and we'll update it, someone else will tweak that, etc. We skim rules from a number of sources such as the Snort-sigs mailing list. We receive a number of rules by direct submission as well.

2. To be an incubator for the rules that actually turn out to be of long term value and reliably accurate. We hope to feed the rules that stand the test of time into the formal Snort.org rulesets.

This site will in the not too distant future be a php interface to let you submit, tweak, and manage sigs. It'll let you choose what you want available to you in a custom ruleset and give you a link to script to automate downloading the sigs you choose.

That is, when we get the time to finish it up... Volunteers welcome!!

We're also soliciting packet dumps of particularly interesting things. Of recent interest has been adware, spyware and the like. If you run into an infected machine please send up a packet dump and we'll get a rule out there for all to benefit from.

mail bleeding@bleedingsnort.com with packet dumps or rules to add

Thanks to Infotex for providing the resources (and salaries :)) to maintain this project.

https://web.archive.org/web/20041230024004/http:/www.bleedingsnort.com/

http://www.bleedingsnort.com/ Go OCT DEC 30 2004 JAN 2005
354 captures
24 Jun 2004 - 7 Dec 2023
About this capture

Bleeding Snort™

The Aggregation Point for Snort Signatures and Research

Search

Welcome to the Bleeding Edge of Snort — **Wednesday, December 29 2004 @ 09:39 PM EST**

Topics

Home
General News
About Bleeding Snort
New Rules
Documentation

Internet Storm Center

- ISC Poll Results, Port 1433 scans
- Infocon: green
- Top Port: microsoft-ds 445
- Fastest Rising: 1433 ms-sql-s

User Functions

Username:
Password:
Login

Don't have an account yet? Sign up as a New User

US-CERT

- SB04-364: Summary of Security Items from December 22 through December 28, 2004
- SB04-357: Summary of Security Items from December 15 through December 21, 2004
- TA04-356A: Exploitation of phpBB highlight parameter vulnerability
- SB04-350: Summary of Security Items from December 8 through December 14, 2004
- ST04-024: Understanding ISPs
- SB04-343: Summary of Security Items from December 1 through December 7, 2004
- SA04-336A: Update for Microsoft Internet Explorer
- TA04-336A: Update for Microsoft Internet Explorer HTML Elements Vulnerability
- SB04-336: Summary of Security Items from November 24 through November 30, 2004
- ST04-023: Understanding Your Computer: Email Clients

Bleeding Snort Signatures

Current Bleeding Snort Rulesets

All in one file:	Bleeding CVS Web Interface
bleeding-all.rules	SPADE CVS Web Interface
bleeding.rules.tar.gz	Daily Change Summary
bleeding-sid-msg-map.txt	Bleeding for the Impatient
Individual Rulesets	How to Submit a Signature or Idea

Latest Forum Posts

Forum	Topic	Last Post
New Signatures	squ1rt Apache DoS sig	12/29/04 15:06
New Signatures	PHP outbound connection detect..	12/29/04 00:10
New Ideas	bogons	12/28/04 22:51
New Signatures	Winhelp32 Trojan Sigs	12/28/04 10:26
New Signatures	Santy Worm Sigs	12/25/04 15:31
General	Merry Christmas / Feliz Navida..	12/25/04 12:51
Tweaks to Existing Signatures	[Q] Joseph Gama's nmap -f -sN ..	12/23/04 17:50

Today's Featured Article

Ruleset Rearranging

Wednesday, December 29 2004 @ 04:06 PM EST
Contributed by: jonkman
Views: 38

We've made some organizational moves to the rulesets. The Scan, DOS, EXPLOIT, ATTACK RESPONSE, and WEB rules are now in their own rulesets. There were enough of each to justify it for roganizational reasons.

What this means to you:
If you're not just using the bleeding-all ruleset you need to add the following to your snort.conf...

read more (42 words) 2 comments
Most Recent Post: 12/29 06:37PM by jonkman

Thanks to Incidents.org

Monday, December 27 2004 @ 05:04 PM EST
Contributed by: jonkman
Views: 39

Incidents.org mentioned the Bleeding Snort community as being a great contributor to their investigations this year in today's diary entry:
http://isc.sans.org/diary.php?date=2004-12-26

We'd like to thank the ISC for their efforts. A lot of the signatures we have are due to their work. If you have the opportunity we encourage you to support that organization through their excellent training programs and the like.

More...

read more (280 words) 0 comments
Post a comment

Trojan Active that Exploits Unpatched XPSP2 Vulnerabilities

Monday, December 27 2004 @ 01:47 PM EST
Contributed by: jonkman
Views: 192

A worm has been identified in the wild that exploits the previously mentioned vulnerabilities in Internet Explorer/XP SP2. There is no patch, these vulnerabilities have been known for a long time and remain unpatched. Your only protection is to not use IE. More...

read more (95 words) 0 comments
Post a comment

Santy.B On the Loose Using AOL/Yahoo Search

Saturday, December 25 2004 @ 12:40 PM EST
Contributed by: jonkman
Views: 86

One of a couple variants of Santy (the PhpBB worm) is using AOL and Yahoo to search. Another as mentioned at Incidents.org is setting up a command connection to an IRC channel.

read more (39 words) 1 comments
Most Recent Post: 12/25 03:20PM by jonkman

Working SPADE/Snort 2.3.0RC2 Tarball Available.

Tuesday, December 21 2004 @ 01:15 AM EST
Contributed by: jonkman
Views: 147

Steve Donegan has sent us a compilable tarball with spade integrated into snort 2.3.0RC2. This represents a good deal of work on Steve's part, many thanks.

read more (91 words) 0 comments

What's New

STORIES
1 story in last 24 hours

COMMENTS last 48 hrs
- Ruleset Rearranging [+2]

LINKS last 2 wks
- TLD's by Country

Forumposts

Order By: New Views Posts
Latest 5 Forum Posts

- **squ1rt Apache DoS ..**
  By: nnachefski
  On: 12/29/04 15:06 PM
  Views 36 Replies 2
- **PHP outbound conne..**
  By: cooljay
  On: 12/29/04 00:10 AM
  Views 151 Replies 8
- **bogons**
  By: emf
  On: 12/28/04 22:49 PM
  Views 238 Replies 7
- **Winhelp32 Trojan S..**
  By: jonkman
  On: 12/28/04 10:26 AM
  Views 185 Replies 2
- **Santy Worm Sigs**
  By: jonkman
  On: 12/25/04 15:30 PM
  Views 423 Replies 4

BOFH

- BOFH gets into the Xmas spirit
- BOFH and the serial killer
- BOFH: Et tu, PFY-us?
- Introducing the big, bad BOFH polo shirt
- BOFH: The hostage's guide to lift imprisonment
- BOFH: A little Ray of sunshine
- BOFH: When sorry seems to be the hardest word
- BOFH: How to survive a Boring Geek Alert
- The BOFH mobile comms quiz
- BOFH: Seek, locate and destroy
- BOFH: A ringside seat at the Boss-baiting pit
- BOFH tests the law of redundant supply
- BOFH 2004: The whole shooting match
- BOFH: What to do when the Boss gets touchy-feely
- PFY proves self abuse cures male-pattern baldness

Security Tracker

- QNX crttrap '-c' Lets Local Users Read or Write Arbitrary Files
- Moodle 'file.php' Discloses Session ID Files and 'view.php' Permits Cross-Site Scripting Attacks
- Owl Intranet Engine Has Unspecified Input Validation Holes That Permit SQL Injection and Cross-Site Scripting Attacks
- PHProjekt 'authform.inc.php' Include File Flaw Lets Remote Users Execute Arbitrary Command
- WHM AutoPilot

Post a comment

CVS Updates Available via RSS

Sunday, December 19 2004 @ 11:50 AM EST
Contributed by: jonkman
Views: 86

By request we've made the CVS changes available via RSS Feed. The rss feed will be updated with each push made into the new rulesets.

We think this will solve the issue of getting timely updates of rule changes without making a mailing list that'd be just too much traffic.

We've also as a side note added a text ChangeLog for review.

read more (42 words) 0 comments
Post a comment

Honeynet Security Console 1.2 Released

Thursday, December 16 2004 @ 10:48 AM EST
Contributed by: Jeff Dell
Views: 147

Honeynet Security Console is a free analysis tool to view events on your personal network or honeynet. It gives you the power to view events from Snort, TCPDump, Firewall, Syslog and Sebek logs. It also allows you to correlate events between each of these data types to have a full view of the attackers' actions.

read more (8 words) 0 comments
Post a comment

ISC Needs Your Help!!

Saturday, December 11 2004 @ 12:16 AM EST
Contributed by: jonkman
Views: 310

The intrepid Incident Handlers at http://isc.sans.org are on the trail of the recent UDP strangeness noted in the last couple days Diary Entries.

http://isc.sans.org/diary.php?date=2004-12-10

Erik Fichtner put great effort into writing signatures to detect this very complex traffic they're investigating. They've asked that we distribute these rules and ask that you report the details of any hits.

read more (61 words) 1 comments
Most Recent Post: 12/14 09:22AM by jonkman

Vulnerability in All Major Browsers

Wednesday, December 08 2004 @ 10:59 PM EST
Contributed by: jonkman
Views: 215

Yes, "All" includes the free browsers as well. The vulnerability is a popup window injection issue. These have never been considered that much of a threat by the security community because the attacker has to know something about where you're going to be browsing, and have their page open at the same time or get you to follow link on their page.

This is a very interesting set of issues for other reasons. Those reasons and a link to a demonstration:

read more (252 words) 0 comments
Post a comment

Snort Installation Guide with BASE and MySQL

Wednesday, December 08 2004 @ 08:38 PM EST
Contributed by: jonkman
Views: 281

Ghaith Nasrawi has just posted a very detailed and useful installation guide to get you going with BASE and snort on Mysql. Very useful if you're trying to get into the IDS arena and need a little push.

read more (48 words) 0 comments
Post a comment

First | Previous | **1** 2 3 4 5 6 | Next | Last

Server Inc Include File Flaw Lets Remote Users Execute Arbitrary Commands

Sophos Virus Alerts

- 29 Dec W32/Leebad-B
- 29 Dec W32/Dedler-H
- 28 Dec W32/Forbot-DH
- 27 Dec Perl/Santy-Fam
- 24 Dec Troj/Agent-ZC
- 24 Dec Troj/Bancos-AS
- 23 Dec Troj/Multidr-BG
- 23 Dec W32/Rembot-A
- 23 Dec W32/Rbot-SD
- 22 Dec W32/Agobot-OR
- Source: Sophos

WindowsSecurity.com

- Happy Holidays from WindowSecurity.com!
- Quick Check: Is Internet Explorer Safe?
- Enforcing GPO Security Settings
- Web Server Security Issues and Front Page Server Extensions
- Is it Time to Start Encrypting Your E-mail?
- Windows Server 2003 Hardening List (Part 1)
- Social Engineering meets the Bot (Part 3) - All is Revealed
- Windows Update Services Review

Copyright © 2004 the Bleeding Edge of Snort
All trademarks and copyrights on this page are owned by their respective owners.

Thanks to infotex! for providing the material resources to maintain this project.
http://www.infotex.com

All content released under the BSD License
Created this page in 0.25 seconds

https://web.archive.org/web/20110430194135/https:/support.sourcefire.com/

INTERNET ARCHIVE
WayBackMachine
https://support.sourcefire.com/
Go
JAN APR SEP
30
2007 2011 2012
93 captures
About this capture

Sourcefire: Security for the real world.

- SUPPORT HOME
- DOWNLOADS
- LICENSE
- VRT NOTICES
- RULES
- ACCOUNT
- SUPPORT TICKET
- CONTACT US

4.10 Is here

Login

Password

Register | Forgot password?

Log in

Current Release

DC : 4.10

3D Sensor : 4.10

RNA Software : 4.9.1.7

Latest Updates

2011-04-27

SEU Update 2011-04-27: SEU 448

2011-04-28

VDB Update 2011-04-28: 66

Certifications

2009-09-14

ICSA Labs Network IPS Certification for Sourcefire 3D Sensor Family, version 4.8.0.2. Contact Technical Support for details.

Visio Stencil

Sourcefire 3D System.zip

Product Lifecycle

It is Sourcefire's policy to provide support for the current and one minor version back of all products.

Click here for a list of currently supported products.

End-of-Life (EOL) Notice for 4.8.x:

With the release of version 4.10 of the 3D System, Sourcefire will no longer support Defense Center 4.8.x, 3D Sensor 4.8.x, and prior versions.

Click here for a copy of Sourcefire's Product Lifecycle Policy.

Hardware EOL Announcements

Sourcefire periodically transitions its oldest supported appliances into an End-of-Life (EOL) phase.

Click here to view the March 2010 Hardware EOL Announcement.

Click here to view the October 2009 Hardware EOL Announcement.

Click here to view the February 2009 Hardware EOL Announcement.

Sourcefire Visualizer EOL Announcement

Effective February 2, 2009, Sourcefire Visualizer was transitioned into an EOL phase.

Click here for more information.

©2010 Sourcefire, Inc.

Terms of Use | Privacy Policy

https://web.archive.org/web/19970605071155/https://www.w3.org/Protocols/HTTP/AsImplemented.html

INTERNET ARCHIVE
Wayback Machine
https://www.w3.org/Protocols/HTTP/AsImplemented.html
Go
585 captures
MAY JUN JUL
05
1996 1997 1999
About this capture



1991

The Original HTTP as defined in 1991

This document defines the Hypertext Transfer protocol (HTTP) as originally implemented by the World Wide Web initaitive software in the prototype released. This is a subset of the full HTTP protocol, and is known as HTTP 0.9.

No client profile information is transferred with the query. Future HTTP protocols will be back-compatible with this protocol.

This restricted protocol is very simple and may always be used when you do not need the capabilities of the full protocol which is backwards compatible.

The definition of this protocol is in the public domain (see policy).

The protocol uses the normal internet-style telnet protocol style on a TCP-IP link. The following describes how a client acquires a (hypertext) document from an HTTP server, given an HTTP document address .

Connection

The client makes a TCP-IP connection to the host using the domain name or IP number , and the port number given in the address.

If the port number is not specified, 80 is always assumed for HTTP.

The server accepts the connection.

Note: HTTP currently runs over TCP, but could run over any connection-oriented service. The interpretation of the protocol below in the case of a sequenced packet service (such as DECnet(TM) or ISO TP4) is that that the request should be one TPDU, but the response may be many.

Request

The client sends a document request consisting of a line of ASCII characters terminated by a CR LF (carriage return, line feed) pair. A well-behaved server will not require the carriage return character.

This request consists of the word "GET", a space, the document address , omitting the "http:, host and port parts when they are the coordinates just used to make the connection. (If a gateway is being used, then a full document address may be given specifying a different naming scheme).

The document address will consist of a single word (ie no spaces). If any further words are found on the request line, they MUST either be ignored, or else treated according to the full HTTP spec .

The search functionality of the protocol lies in the ability of the addressing syntax to describe a search on a named index .

A search should only be requested by a client when the index document itself has been descibed as an index using the ISINDEX tag .

Response

The response to a simple GET request is a message in hypertext mark-up language (HTML). This is a byte stream of ASCII characters.

Lines shall be delimited by an optional carriage return followed by a mandatory line feed chararcter. The client should not assume that the carriage return will be present. Lines may be of any length. Well-behaved servers should retrict line length to 80 characters excluding the CR LF pair.

The format of the message is HTML - that is, a trimmed SGML document. Note that this format allows for menus and hit lists to be returned as hypertext. It also allows for plain ASCII text to be returned following the PLAINTEXT tag .

The message is terminated by the closing of the connection by the server.

Well-behaved clients will read the entire document as fast as possible. The client shall not wait for user action (output paging for example) before reading the whole of the document. The server may impose a timeout of the order of 15 seconds on inactivity.

Error responses are supplied in human readable text in HTML syntax. There is no way to distinguish an error response from a satisfactory response except for the content of the text.

Disconnection

The TCP-IP connection is broken by the server when the whole document has been transferred.

The client may abort the transfer by breaking the connection before this, in which case the server shall not record any error condition.

Requests are idempotent . The server need not store any information about the request after disconnection.

Tim BL

https://web.archive.org/web/20070206005843/https://tools.ietf.org/html/rfc765

https://tools.ietf.org/html/rfc765 Go JAN FEB APR 06 2007 2008 208 captures 6 Feb 2007 - 2 Oct 2023 About this capture

Obsoleted by: 959

IEN 149
RFC 765

J. Postel
ISI
June 1980

FILE TRANSFER PROTOCOL

INTRODUCTION

The objectives of FTP are 1) to promote sharing of files (computer programs and/or data), 2) to encourage indirect or implicit (via programs) use of remote computers, 3) to shield a user from variations in file storage systems among Hosts, and 4) to transfer data reliably and efficiently. FTP, though usable directly by a user at a terminal, is designed mainly for use by programs.

The attempt in this specification is to satisfy the diverse needs of users of maxi-Hosts, mini-Hosts, and TIPs, with a simple, and easily implemented protocol design.

This paper assumes knowledge of the following protocols described in the ARPA Internet Protocol Handbook.

The Transmission Control Protocol

The TELNET Protocol

DISCUSSION

In this section, the terminology and the FTP model are discussed. The terms defined in this section are only those that have special significance in FTP. Some of the terminology is very specific to the FTP model; some readers may wish to turn to the section on the FTP model while reviewing the terminology.

TERMINOLOGY

ASCII

The ASCII character set as defined in the ARPA Internet Protocol Handbook. In FTP, ASCII characters are defined to be the lower half of an eight-bit code set (i.e., the most significant bit is zero).

access controls

Access controls define users' access privileges to the use of a system, and to the files in that system. Access controls are necessary to prevent unauthorized or accidental use of files. It is the prerogative of a server-FTP process to invoke access controls.

byte size

There are two byte sizes of interest in FTP: the logical byte size of the file, and the transfer byte size used for the transmission of the data. The transfer byte size is always 8 bits. The transfer byte size is not necessarily the byte size in which data is to be stored in a system, nor the logical byte size for interpretation of the structure of the data.

data connection

A simplex connection over which data is transferred, in a specified mode and type. The data transferred may be a part of a file, an entire file or a number of files. The path may be between a server-DTP and a user-DTP, or between two server-DTPs.

data port

The passive data transfer process "listens" on the data port for a connection from the active transfer process in order to open the data connection.

EOF

The end-of-file condition that defines the end of a file being

IPR2023-01353 Page 00025

transferred.

EOR

The end-of-record condition that defines the end of a record being transferred.

error recovery

A procedure that allows a user to recover from certain errors such as failure of either Host system or transfer process. In FTP, error recovery may involve restarting a file transfer at a given checkpoint.

FTP commands

A set of commands that comprise the control information flowing from the user-FTP to the server-FTP process.

file

An ordered set of computer data (including programs), of arbitrary length, uniquely identified by a pathname.

mode

The mode in which data is to be transferred via the data connection. The mode defines the data format during transfer including EOR and EOF. The transfer modes defined in FTP are described in the Section on Transmission Modes.

NVT

The Network Virtual Terminal as defined in the TELNET Protocol.

NVFS

The Network Virtual File System. A concept which defines a standard network file system with standard commands and pathname conventions. FTP only partially implements the NVFS concept at this time.

page

A file may be structured as a set of independent parts called pages. FTP supports the transmission of discontinuous files as independent indexed pages.

pathname

Pathname is defined to be the character string which must be input to a file system by a user in order to identify a file. Pathname normally contains device and/or directory names, and file name specification. FTP does not yet specify a standard pathname convention. Each user must follow the file naming conventions of the file systems involved in the transfer.

record

A sequential file may be structured as a number of contiguous parts called records. Record structures are supported by FTP but a file need not have record structure.

reply

A reply is an acknowledgment (positive or negative) sent from server to user via the TELNET connections in response to FTP commands. The general form of a reply is a completion code (including error codes) followed by a text string. The codes are for use by programs and the text is usually intended for human users.

IPR2023-01353 Page 00026

human users.

server-DTP

The data transfer process, in its normal "active" state, establishes the data connection with the "listening" data port, sets up parameters for transfer and storage, and transfers data on command from its PI. The DTP can be placed in a "passive" state to listen for, rather than initiate a, connection on the data port.

server-FTP process

A process or set of processes which perform the function of file transfer in cooperation with a user-FTP process and, possibly, another server. The functions consist of a protocol interpreter (PI) and a data transfer process (DTP).

server-PI

The protocol interpreter "listens" on Port L for a connection from a user-PI and establishes a TELNET communication connection. It receives standard FTP commands from the user-PI, sends replies, and governs the server-DTP.

TELNET connections

The full-duplex communication path between a user-PI and a server-PI, operating according to the TELNET Protocol.

type

The data representation type used for data transfer and storage. Type implies certain transformations between the time of data storage and data transfer. The representation types defined in FTP are described in the Section on Establishing Data Connections.

user

A human being or a process on behalf of a human being wishing to obtain file transfer service. The human user may interact directly with a server-FTP process, but use of a user-FTP process is preferred since the protocol design is weighted towards automata.

user-DTP

The data transfer process "listens" on the data port for a connection from a server-FTP process. If two servers are transferring data between them, the user-DTP is inactive.

user-FTP process

A set of functions including a protocol interpreter, a data transfer process and a user interface which together perform the function of file transfer in cooperation with one or more server-FTP processes. The user interface allows a local language to be used in the command-reply dialogue with the user.

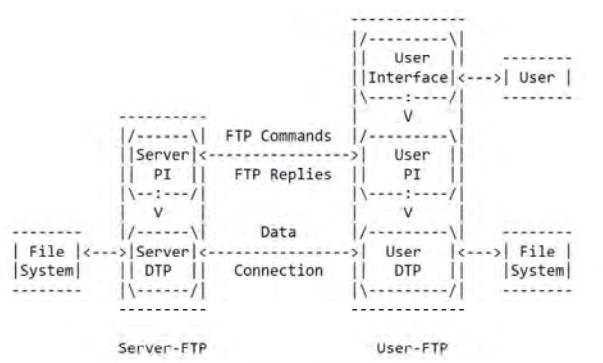
user-PI

The protocol interpreter initiates the TELNET connection from its port U to the server-FTP process, initiates FTP commands, and governs the user-DTP if that process is part of the file transfer.

IPR2023-01353 Page 00027

THE FTP MODEL

With the above definitions in mind, the following model (shown in Figure 1) may be diagrammed for an FTP service.



NOTES: 1. The data connection may be used in either direction.
2. The data connection need not exist all of the time.

Figure 1 Model for FTP Use

In the model described in Figure 1, the user-protocol interpreter initiates the TELNET connection. At the initiation of the user, standard FTP commands are generated by the user-PI and transmitted to the server process via the TELNET connection. (The user may establish a direct TELNET connection to the server-FTP, from a TIP terminal for example, and generate standard FTP commands himself, bypassing the user-FTP process.) Standard replies are sent from the server-PI to the user-PI over the TELNET connection in response to the commands.

The FTP commands specify the parameters for the data connection (data port, transfer mode, representation type, and structure) and the nature of file system operation (store, retrieve, append, delete, etc.). The user-DTP or its designate should "listen" on the specified data port, and the server initiate the data connection and data transfer in accordance with the specified parameters. It should be noted that the data port need not be in

the same Host that initiates the FTP commands via the TELNET connection, but the user or his user-FTP process must ensure a "listen" on the specified data port. It should also be noted that the data connection may be used for simultaneous sending and receiving.

In another situation a user might wish to transfer files between two Hosts, neither of which is his local Host. He sets up TELNET connections to the two servers and then arranges for a data connection between them. In this manner control information is passed to the user-PI but data is transferred between the server data transfer processes. Following is a model of this server-server interaction.

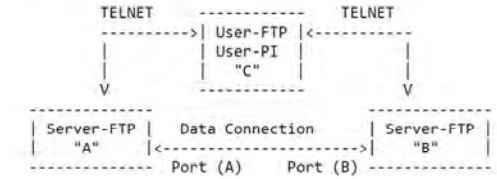


Figure 2

The protocol requires that the TELNET connections be open while

IPR2023-01353 Page 00028

data transfer is in progress. It is the responsibility of the user to request the closing of the TELNET connections when finished using the FTP service, while it is the server who takes the action. The server may abort data transfer if the TELNET connections are closed without command.

DATA TRANSFER FUNCTIONS

Files are transferred only via the data connection. The TELNET connection is used for the transfer of commands, which describe the functions to be performed, and the replies to these commands (see the Section on FTP Replies). Several commands are concerned with the transfer of data between Hosts. These data transfer commands include the MODE command which specify how the bits of the data are to be transmitted, and the STRUcture and TYPE commands, which are used to define the way in which the data are to be represented. The transmission and representation are basically independent but

"Stream" transmission mode is dependent on the file structure attribute and if "Compressed" transmission mode is used the nature of the filler byte depends on the representation type.

DATA REPRESENTATION AND STORAGE

Data is transferred from a storage device in the sending Host to a storage device in the receiving Host. Often it is necessary to perform certain transformations on the data because data storage representations in the two systems are different. For example, NVT-ASCII has different data storage representations in different systems. PDP-10's generally store NVT-ASCII as five 7-bit ASCII characters, left-justified in a 36-bit word. 360's store NVT-ASCII as 8-bit EBCDIC codes. Multics stores NVT-ASCII as four 9-bit characters in a 36-bit word. It may be desirable to convert characters into the standard NVT-ASCII representation when transmitting text between dissimilar systems. The sending and receiving sites would have to perform the necessary transformations between the standard representation and their internal representations.

A different problem in representation arises when transmitting binary data (not character codes) between Host systems with different word lengths. It is not always clear how the sender should send data, and the receiver store it. For example, when transmitting 32-bit bytes from a 32-bit word-length system to a 36-bit word-length system, it may be desirable (for reasons of efficiency and usefulness) to store the 32-bit bytes right-justified in a 36-bit word in the latter system. In any case, the user should have the option of specifying data representation and transformation functions. It should be noted that FTP provides for very limited data type representations. Transformations desired beyond this limited capability should be performed by the user directly.

Data representations are handled in FTP by a user specifying a representation type. This type may implicitly (as in ASCII or EBCDIC) or explicitly (as in Local byte) define a byte size for interpretation which is referred to as the "logical byte size." This has nothing to do with the byte size used for transmission over the data connection, called the "transfer byte size", and the two should not be confused. For example, NVT-ASCII has a logical byte size of 8 bits. If the type is Local byte, then the TYPE command has an obligatory second parameter specifying the logical byte size. The transfer byte size is always 8 bits.

The types ASCII and EBCDIC also take a second (optional) parameter; this is to indicate what kind of vertical format control, if any, is associated with a file. The following data representation types are defined in FTP:

ASCII Format

This is the default type and must be accepted by all FTP implementations. It is intended primarily for the transfer of text files, except when both Hosts would find the EBCDIC

IPR2023-01353 Page 00029

type more convenient.

The sender converts the data from his internal character representation to the standard 8-bit NVT-ASCII representation (see the TELNET specification). The receiver will convert the data from the standard form to his own internal form.

In accordance with the NVT standard, the <CRLF> sequence should be used, where necessary, to denote the end of a line of text. (See the discussion of file structure at the end of the Section on Data Representation and Storage).

Using the standard NVT-ASCII representation means that data must be interpreted as 8-bit bytes.

The Format parameter for ASCII and EBCDIC types is discussed below.

EBCDIC Format

This type is intended for efficient transfer between Hosts which use EBCDIC for their internal character representation.

For transmission the data are represented as 8-bit EBCDIC characters. The character code is the only difference between the functional specifications of EBCDIC and ASCII types.

End-of-line (as opposed to end-of-record--see the discussion of structure) will probably be rarely used with EBCDIC type for purposes of denoting structure, but where it is necessary the <NL> character should be used.

A character file may be transferred to a Host for one of three purposes: for printing, for storage and later retrieval, or for processing. If a file is sent for printing, the receiving Host must know how the vertical format control is represented. In the second case, it must be possible to store a file at a Host and then retrieve it later in exactly the same form. Finally, it ought to be possible to move a file from one Host to another and process the file at the second Host without undue trouble. A single ASCII or EBCDIC format does not satisfy all these conditions and so these types have a second parameter specifying one of the following three formats:

Non-print

This is the default format to be used if the second (format) parameter is omitted. Non-print format must be accepted by all FTP implementations.

The file need contain no vertical format information. If it is passed to a printer process, this process may assume standard values for spacing and margins.

Normally, this format will be used with files destined for processing or just storage.

TELNET Format Controls

The file contains ASCII/EBCDIC vertical format controls (i.e., <CR>, <LF>, <NL>, <VT>, <FF>) which the printer process will interpret appropriately. <CRLF>, in exactly this sequence, also denotes end-of-line.

Carriage Control (ASA)

The file contains ASA (FORTRAN) vertical format control characters. (See RFC 740 Appendix C and Communications of the ACM, Vol. 7, No. 10, 606 (Oct. 1964)). In a line or a record, formatted according to the ASA Standard, the first character is not to be printed. Instead it should be used to determine the vertical movement of the paper which should take place before the rest of the record is printed.

IPR2023-01353 Page 00030

The ASA Standard specifies the following control characters:

Character	Vertical Spacing
blank	Move paper up one line
0	Move paper up two lines
1	Move paper to top of next page
+	No movement, i.e., overprint

Clearly there must be some way for a printer process to distinguish the end of the structural entity. If a file has record structure (see below) this is no problem; records will be explicitly marked during transfer and storage. If the file has no record structure, the <CRLF> end-of-line sequence is used to separate printing lines, but these format effectors are overridden by the ASA controls.

Image

The data are sent as contiguous bits which, for transfer, are packed into the 8-bit transfer bytes. The receiving site must store the data as contiguous bits. The structure of the storage system might necessitate the padding of the file (or of each record, for a record-structured file) to some convenient boundary (byte, word or block). This padding, which must be all zeros, may occur only at the end of the file (or at the end of each record) and there must be a way of identifying the padding bits so that they may be stripped off if the file is retrieved. The padding transformation should be well publicized to enable a user to process a file at the storage site.

Image type is intended for the efficient storage and retrieval of files and for the transfer of binary data. It is recommended that this type be accepted by all FTP implementations.

Local byte Byte size

The data is transferred in logical bytes of the size specified by the obligatory second parameter, Byte size. The value of Byte size must be a decimal integer; there is no default value. The logical byte size is not necessarily the same as the transfer byte size. If there is a

difference in byte sizes, then the logical bytes should be packed contiguously, disregarding transfer byte boundaries and with any necessary padding at the end.

When the data reaches the receiving Host it will be transformed in a manner dependent on the logical byte size and the particular Host. This transformation must be invertible (that is an identical file can be retrieved if the same parameters are used) and should be well publicized by the FTP implementors.

For example, a user sending 36-bit floating-point numbers to a Host with a 32-bit word could send his data as Local byte with a logical byte size of 36. The receiving Host would then be expected to store the logical bytes so that they could be easily manipulated; in this example putting the 36-bit logical bytes into 64-bit double words should suffice.

Another example, a pair of hosts with a 36-bit word size may send data to one another in words by using TYPE L 36. The data would be sent in the 8-bit transmission bytes packed so that 9 transmission bytes carried two host words.

A note of caution about parameters: a file must be stored and retrieved with the same parameters if the retrieved version is to be identical to the version originally transmitted. Conversely, FTP implementations must return a file identical to the original if the parameters used to store and retrieve a file are the same.

In addition to different representation types, FTP allows the structure of a file to be specified. Three file structures are

IPR2023-01353 Page 00031

defined in FTP:

   file-structure, where there is no internal structure and the
                   file is considered to be a continuous
                   sequence of data bytes,

   record-structure, where the file is made up of sequential
                   records,

   and page-structure, where the file is made up of independent
                   indexed pages.

File-structure is the default, to be assumed if the STRUcture command has not been used but both file and record structures must

be accepted for "text" files (i.e., files with TYPE ASCII or EBCDIC) by all FTP implementations. The structure of a file will affect both the transfer mode of a file (see the Section on Transmission Modes) and the interpretation and storage of the file.

The "natural" structure of a file will depend on which Host stores the file. A source-code file will usually be stored on an IBM 360 in fixed length records but on a PDP-10 as a stream of characters partitioned into lines, for example by <CRLF>. If the transfer of files between such disparate sites is to be useful, there must be some way for one site to recognize the other's assumptions about the file.

With some sites being naturally file-oriented and others naturally record-oriented there may be problems if a file with one structure is sent to a Host oriented to the other. If a text file is sent with record-structure to a Host which is file oriented, then that Host should apply an internal transformation to the file based on the record structure. Obviously this transformation should be useful but it must also be invertible so that an identical file may be retrieved using record structure.

In the case of a file being sent with file-structure to a record-oriented Host, there exists the question of what criteria the Host should use to divide the file into records which can be processed locally. If this division is necessary the FTP implementation should use the end-of-line sequence, <CRLF> for ASCII, or <NL> for EBCDIC text files, as the delimiter. If an FTP implementation adopts this technique, it must be prepared to reverse the transformation if the file is retrieved with file-structure.

Page Structure

   To transmit files that are discontinuous FTP defines a page structure. Files of this type are sometimes know as "random access files" or even as "holey files". In these files there is sometimes other information associated with the file as a whole (e.g., a file descriptor), or with a section of the file (e.g., page access controls), or both. In FTP, the sections of the file are called pages.

   To provide for various page sizes and associated information each page is sent with a page header. The page header has the following defined fields:

      Header Length

         The number of logical bytes in the page header including this byte. The minimum header length is 4.

      Page Index

         The logical page number of this section of the file. This is not the transmission sequence number of this page, but the index used to identify this page of the file.

      Data Length

IPR2023-01353 Page 00032

The number of logical bytes in the page data. The minimum data length is 0.

Page Type

The type of page this is. The following page types are defined:

0 = Last Page

This is used to indicate the end of a paged structured transmission. The header length must be 4, and the data length must be 0.

1 = Simple Page

This is the normal type for simple paged files with no page level associated control information. The header length must be 4.

2 = Descriptor Page

This type is used to transmit the descriptive information for the file as a whole.

3 = Access Controled Page

This is type includes an additional header field for paged files with page level access control information. The header length must be 5.

Optional Fields

Further header fields may be used to supply per page control information, for example, per page access control.

All fields are one logical byte in length. The logical byte size is specified by the TYPE command.

ESTABLISHING DATA CONNECTIONS

The mechanics of transferring data consists of setting up the data connection to the appropriate ports and choosing the parameters for transfer. Both the user and the server-DTPs have a default data port. The user-process default data port is the same as the control connection port, i.e., U. The server-process default data port is the port adjacent to the control connection port, i.e., L-1.

The transfer byte size is 8-bit bytes. This byte size is relevant only for the actual transfer of the data; it has no bearing on representation of the data within a Host's file system.

The passive data transfer process (this may be a user-DTP or a second server-DTP) shall "listen" on the data port prior to sending a transfer request command. The FTP request command determines the direction of the data transfer. The server, upon receiving the transfer request, will initiate the data connection to the port. When the connection is established, the data transfer begins between DTP's, and the server-PI sends a confirming reply to the user-PI.

It is possible for the user to specify an alternate data port by use of the PORT command. He might want a file dumped on a TIP line printer or retrieved from a third party Host. In the latter case the user-PI sets up TELNET connections with both server-PI's. One server is then told (by an FTP command) to "listen" for a connection which the other will initiate. The user-PI sends one server-PI a PORT command indicating the data port of the other. Finally both are sent the appropriate transfer commands. The exact sequence of commands and replies sent between the user-controller and the servers is defined in the Section on FTP Replies.

In general it is the server's responsibility to maintain the data connection--to initiate it and to close it. The exception to this

IPR2023-01353 Page 00033

is when the user-DTP is sending the data in a transfer mode that requires the connection to be closed to indicate EOF. The server MUST close the data connection under the following conditions:

1. The server has completed sending data in a transfer mode that requires a close to indicate EOF.

2. The server receives an ABORT command from the user.

3. The port specification is changed by a command from the user.

4. The TELNET connection is closed legally or otherwise.

5. An irrecoverable error condition occurs.

Otherwise the close is a server option, the exercise of which he must indicate to the user-process by an appropriate reply.

TRANSMISSION MODES

The next consideration in transferring data is choosing the appropriate transmission mode. There are three modes: one which formats the data and allows for restart procedures; one which also compresses the data for efficient transfer; and one which passes the data with little or no processing. In this last case the mode interacts with the structure attribute to determine the type of processing. In the compressed mode the representation type determines the filler byte.

All data transfers must be completed with an end-of-file (EOF) which may be explicitly stated or implied by the closing of the data connection. For files with record structure, all the end-of-record markers (EOR) are explicit, including the final one. For files transmitted in page structure a "last-page" page type is used.

NOTE: In the rest of this section, byte means "transfer byte" except where explicitly stated otherwise.

For the purpose of standardized transfer, the sending Host will translate his internal end of line or end of record denotation into the representation prescribed by the transfer mode and file structure, and the receiving Host will perform the inverse translation to his internal denotation. An IBM 360 record count field may not be recognized at another Host, so the end of record

information may be transferred as a two byte control code in Stream mode or as a flagged bit in a Block or Compressed mode descriptor. End of line in an ASCII or EBCDIC file with no record structure should be indicated by <CRLF> or <NL>, respectively. Since these transformations imply extra work for some systems, identical systems transferring non-record structured text files might wish to use a binary representation and stream mode for the transfer.

The following transmission modes are defined in FTP:

STREAM

The data is transmitted as a stream of bytes. There is no restriction on the representation type used; record structures are allowed.

In a record structured file EOR and EOF will each be indicated by a two-byte control code. The first byte of the control code will be all ones, the escape character. The second byte will have the low order bit on and zeros elsewhere for EOR and the second low order bit on for EOF; that is, the byte will have value 1 for EOR and value 2 for EOF. EOR and EOF may be indicated together on the last byte transmitted by turning both low order bits on, i.e., the value 3. If a byte of all ones was intended to be sent as data, it should be repeated in the second byte of the control code.

If the structure is file structure, the EOF is indicated by the sending Host closing the data connection and all bytes are data bytes.

BLOCK

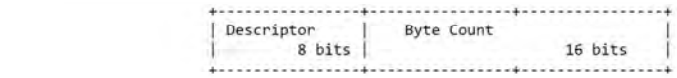
IPR2023-01353 Page 00034

The file is transmitted as a series of data blocks preceded by one or more header bytes. The header bytes contain a count field, and descriptor code. The count field indicates the total length of the data block in bytes, thus marking the beginning of the next data block (there are no filler bits). The descriptor code defines: last block in the file (EOF) last block in the record (EOR), restart marker (see the Section on Error Recovery and Restart) or suspect data (i.e., the data being transferred is suspected of errors and is not reliable). This last code is NOT intended for error control within FTP. It is motivated by the desire of sites

exchanging certain types of data (e.g., seismic or weather data) to send and receive all the data despite local errors (such as "magnetic tape read errors"), but to indicate in the transmission that certain portions are suspect). Record structures are allowed in this mode. and any representation type may be used.

The header consists of the three bytes. Of the 24 bits of header information, the 16 low order bits shall represent byte count, and the 8 high order bits shall represent descriptor codes as shown below.

Block Header



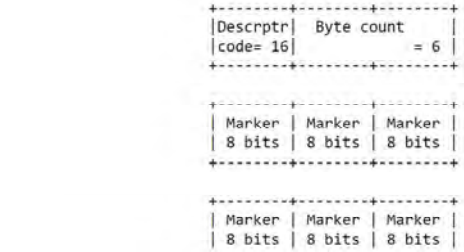
The descriptor codes are indicated by bit flags in the descriptor byte. Four codes have been assigned, where each code number is the decimal value of the corresponding bit in the byte.

Code	Meaning
128	End of data block is EOR
64	End of data block is EOF
32	Suspected errors in data block
16	Data block is a restart marker

With this encoding more than one descriptor coded condition may exist for a particular block. As many bits as necessary may be flagged.

The restart marker is embedded in the data stream as an integral number of 8-bit bytes representing printable characters in the language being used over the TELNET connection (e.g., default--NVT-ASCII). <SP> (Space, in the appropriate language) must not be used WITHIN a restart marker.

For example, to transmit a six-character marker, the following would be sent:

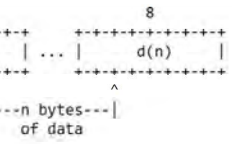


IPR2023-01353 Page 00035

COMPRESSED

There are three kinds of information to be sent: regular data, sent in a byte string; compressed data, consisting of replications or filler; and control information, sent in a two-byte escape sequence. If n>0 bytes (up to 127) of regular data are sent, these n bytes are preceded by a byte with the left-most bit set to 0 and the right-most 7 bits containing the number n.

Byte string:



String of n data bytes d(1),..., d(n)
Count n must be positive.

To compress a string of n replications of the data byte d, the following 2 bytes are sent:

Replicated Byte:



A string of n filler bytes can be compressed into a single byte, where the filler byte varies with the representation type. If the type is ASCII or EBCDIC the filler byte is <SP> (Space, ASCII code 32., EBCDIC code 64). If the type is Image or Local byte the filler is a zero byte.

Filler String:

The escape sequence is a double byte, the first of which is the escape byte (all zeros) and the second of which contains descriptor codes as defined in Block mode. The descriptor codes have the same meaning as in Block mode and apply to the succeeding string of bytes.

Compressed mode is useful for obtaining increased bandwidth on very large network transmissions at a little extra CPU cost. It can be most effectively used to reduce the size of printer files such as those generated by RJE Hosts.

ERROR RECOVERY AND RESTART

There is no provision for detecting bits lost or scrambled in data transfer; this level of error control is handled by the TCP. However, a restart procedure is provided to protect users from gross system failures (including failures of a Host, an FTP-process, or the underlying network).

The restart procedure is defined only for the block and compressed modes of data transfer. It requires the sender of data to insert a special marker code in the data stream with some marker information. The marker information has meaning only to the sender, but must consist of printable characters in the default or negotiated language of the TELNET connection (ASCII or EBCDIC). The marker could represent a bit-count, a record-count, or any

IPR2023-01353 Page 00036

other information by which a system may identify a data checkpoint. The receiver of data, if it implements the restart procedure, would then mark the corresponding position of this marker in the receiving system, and return this information to the user.

In the event of a system failure, the user can restart the data transfer by identifying the marker point with the FTP restart procedure. The following example illustrates the use of the restart procedure.

The sender of the data inserts an appropriate marker block in the data stream at a convenient point. The receiving Host marks the corresponding data point in its file system and conveys the last known sender and receiver marker information to the user, either directly or over the TELNET connection in a 110 reply (depending on who is the sender). In the event of a system failure, the user or controller process restarts the server at the last server marker by sending a restart command with server's marker code as its argument. The restart command is transmitted over the TELNET connection and is immediately followed by the command (such as RETR, STOR or LIST) which was being executed when the system failure occurred.

FILE TRANSFER FUNCTIONS

The communication channel from the user-PI to the server-PI is established by a TCP connection from the user to a standard server port. The user protocol interpreter is responsible for sending FTP commands and interpreting the replies received; the server-PI interprets commands, sends replies and directs its DTP to set up the data connection and transfer the data. If the second party to the data transfer (the passive transfer process) is the user-DTP then it is governed through the internal protocol of the user-FTP Host; if it is a second server-DTP then it is governed by its PI on command from the user-PI. The FTP replies are discussed in the next section. In the description of a few of the commands in this section it is helpful to be explicit about the possible replies.

FTP COMMANDS

ACCESS CONTROL COMMANDS

The following commands specify access control identifiers (command codes are shown in parentheses).

USER NAME (USER)

The argument field is a TELNET string identifying the user. The user identification is that which is required by the server for access to its file system. This command will normally be the first command transmitted by the user after the TELNET connections are made (some servers may require this). Additional identification information in the form of a password and/or an account command may also be required by some servers. Servers may allow a new USER command to be entered at any point in order to change the access control and/or accounting information. This has the effect of flushing any user, password, and account information already supplied and beginning the login sequence again. All transfer parameters are unchanged and any file transfer in progress is completed under the old account.

PASSWORD (PASS)

The argument field is a TELNET string identifying the user's password. This command must be immediately preceded by the user name command, and, for some sites, completes the user's identification for access control. Since password information is quite sensitive, it is desirable in general to "mask" it or suppress typeout. It appears that the server has no foolproof way to achieve this. It is therefore the responsibility of the user-FTP process to hide the sensitive password information.

ACCOUNT (ACCT)

The argument field is a TELNET string identifying the user's account. The command is not necessarily related to the USER command, as some sites may require an account for login and others only for specific access, such as storing files. In the latter case the command may arrive at any time.

There are reply codes to differentiate these cases for the

IPR2023-01353 Page 00037

automaton: when account information is required for login, the response to a successful PASSword command is reply code 332. On the other hand, if account information is NOT required for login, the reply to a successful PASSword command is 230; and if the account information is needed for a command issued later in the dialogue, the server should

return a 332 or 532 reply depending on whether he stores (pending receipt of the ACCounT command) or discards the command, respectively.

REINITIALIZE (REIN)

This command terminates a USER, flushing all I/O and account information, except to allow any transfer in progress to be completed. All parameters are reset to the default settings and the TELNET connection is left open. This is identical to the state in which a user finds himself immediately after the TELNET connection is opened. A USER command may be expected to follow.

LOGOUT (QUIT)

This command terminates a USER and if file transfer is not in progress, the server closes the TELNET connection. If file transfer is in progress, the connection will remain open for result response and the server will then close it. If the user-process is transferring files for several USERs but does not wish to close and then reopen connections for each, then the REIN command should be used instead of QUIT.

An unexpected close on the TELNET connection will cause the server to take the effective action of an abort (ABOR) and a logout (QUIT).

TRANSFER PARAMETER COMMANDS

All data transfer parameters have default values, and the commands specifying data transfer parameters are required only if the default parameter values are to be changed. The default value is the last specified value, or if no value has been specified, the standard default value as stated here. This implies that the server must "remember" the applicable default values. The commands may be in any order except that they must precede the FTP service request. The following commands specify data transfer parameters.

DATA PORT (PORT)

The argument is a HOST-PORT specification for the data port to be used in data connection. There defaults for both the user and server data ports, and under normal circumstances this command and its reply are not needed. If this command

is used the argument is the concatenation of a 32-bit internet host address and a 16-bit TCP port address. This address information is broken into 8-bit fields and the value of each field is transmitted as a decimal number (in character string representation). The fields are separated by commas. A port command would be:

PORT h1,h2,h3,h4,p1,p2

where, h1 is the high order 8 bits of the internet host address.

PASSIVE (PASV)

This command requests the server-DTP to "listen" on a data port (which is not its default data port) and to wait for a connection rather than initiate one upon receipt of a transfer command. The response to this command includes the host and port address this server is listening on.

IPR2023-01353 Page 00038

REPRESENTATION TYPE (TYPE)

The argument specifies the representation type as described in the Section on Data Representation and Storage. Several types take a second parameter. The first parameter is denoted by a single TELNET character, as is the second Format parameter for ASCII and EBCDIC; the second parameter for local byte is a decimal integer to indicate Bytesize. The parameters are separated by a <SP> (Space, ASCII code 32.).

The following codes are assigned for type:



L <byte size> - Local byte Byte size

The default representation type is ASCII Non-print. If the Format parameter is changed, and later just the first argument is changed, Format then returns to the Non-print default.

FILE STRUCTURE (STRU)

The argument is a single TELNET character code specifying file structure described in the Section on Data Representation and Storage.

The following codes are assigned for structure:

F - File (no record structure)
R - Record structure
P - Page structure

The default structure is File.

TRANSFER MODE (MODE)

The argument is a single TELNET character code specifying the data transfer modes described in the Section on Transmission Modes.

The following codes are assigned for transfer modes:

S - Stream
B - Block
C - Compressed

The default transfer mode is Stream.

FTP SERVICE COMMANDS

The FTP service commands define the file transfer or the file system function requested by the user. The argument of an FTP service command will normally be a pathname. The syntax of pathnames must conform to server site conventions (with standard defaults applicable), and the language conventions of the TELNET connection. The suggested default handling is to use the last specified device, directory or file name, or the standard default defined for local users. The commands may be in any order except that a "rename from" command must be followed by a "rename to" command and the restart command must be followed by the interrupted service command. The data, when transferred in response to FTP service commands, shall always be sent over the data connection, except for certain informative replies. The following commands specify FTP service requests:

RETRIEVE (RETR)

IPR2023-01353 Page 00039

RETRIEVE (RETR)

This command causes the server-DTP to transfer a copy of the file, specified in the pathname, to the server- or user-DTP at the other end of the data connection. The status and contents of the file at the server site shall be unaffected.

STORE (STOR)

This command causes the server-DTP to accept the data transferred via the data connection and to store the data as a file at the server site. If the file specified in the pathname exists at the server site then its contents shall be replaced by the data being transferred. A new file is created at the server site if the file specified in the pathname does not already exist.

APPEND (with create) (APPE)

This command causes the server-DTP to accept the data transferred via the data connection and to store the data in a file at the server site. If the file specified in the pathname exists at the server site, then the data shall be appended to that file; otherwise the file specified in the pathname shall be created at the server site.

MAIL FILE (MLFL)

The intent of this command is to enable a user at the user site to mail data (in form of a file) to another user at the server site. It should be noted that the files to be mailed are transmitted via the data connection in ASCII or EBCDIC type. (It is the user's responsibility to ensure that the type is correct.) These files should be inserted into the destination user's mailbox by the server in accordance with serving Host mail conventions. The mail may be marked as sent from the particular user HOST and the user specified by the 'USER' command. The argument field may contain a Host system ident, or it may be empty. If the argument field is empty or blank (one or more spaces), then the mail is destined for a printer or other designated place for general delivery site mail.

MAIL (MAIL)

This command allows a user to send mail that is NOT in a file over the TELNET connection. The argument field may contain system ident, or it may be empty. The ident is defined as above for the MLFL command. After the 'MAIL' command is received, the server is to treat the following lines as text of the mail sent by the user. The mail text is to be terminated by a line containing only a single period, that is, the character sequence "CRLF.CRLF". It is suggested that a modest volume of mail service should be free; i.e., it may be entered before a USER command.

MAIL SEND TO TERMINAL (MSND)

This command is like the MAIL command, except that the data is displayed on the addressed user's terminal, if such access is currently allowed, otherwise an error is returned.

MAIL SEND TO TERMINAL OR MAILBOX (MSOM)

This command is like the MAIL command, except that the data is displayed on the addressed user's terminal, if such access is currently allowed, otherwise the data is placed in the user's mailbox.

MAIL SEND TO TERMINAL AND MAILBOX (MSAM)

This command is like the MAIL command, except that the data is displayed on the addressed user's terminal, if such access is currently allowed, and, in any case, the data is placed in the user's mailbox.

MAIL RECIPIENT SCHEME QUESTION (MRSQ)

This FTP command is used to select a scheme for the transmission of mail to several users at the same host. The schemes are to list the recipients first, or to send the mail first.

MAIL RECIPIENT (MRCP)

IPR2023-01353 Page 00040

MAIL RECIPIENT (MRCP)

This command is used to identify the individual recipients of the mail in the transmission of mail for multiple users at one host.

ALLOCATE (ALLO)

This command may be required by some servers to reserve sufficient storage to accommodate the new file to be transferred. The argument shall be a decimal integer representing the number of bytes (using the logical byte size) of storage to be reserved for the file. For files sent with record or page structure a maximum record or page size (in logical bytes) might also be necessary; this is indicated by a decimal integer in a second argument field of the command. This second argument is optional, but when present should be separated from the first by the three TELNET characters <SP> R <SP>. This command shall be followed by a STORe or APPEnd command. The ALLO command should be treated as a NOOP (no operation) by those servers which do not require that the maximum size of the file be declared beforehand, and those servers interested in only the maximum record or page size should accept a dummy value in the first argument and ignore it.

RESTART (REST)

The argument field represents the server marker at which file transfer is to be restarted. This command does not cause file transfer but "spaces" over the file to the specified data checkpoint. This command shall be immediately followed by the appropriate FTP service command which shall cause file transfer to resume.

RENAME FROM (RNFR)

This command specifies the file which is to be renamed. This command must be immediately followed by a "rename to" command specifying the new file pathname.

RENAME TO (RNTO)

This command specifies the new pathname of the file specified in the immediately preceding "rename from" command. Together the two commands cause a file to be renamed.

ABORT (ABOR)

This command tells the server to abort the previous FTP service command and any associated transfer of data. The

abort command may require "special action", as discussed in the Section on FTP Commands, to force recognition by the server. No action is to be taken if the previous command has been completed (including data transfer). The TELNET connection is not to be closed by the server, but the data connection must be closed.

There are two cases for the server upon receipt of this command: (1) the FTP service command was already completed, or (2) the FTP service command is still in progress.

In the first case, the server closes the data connection (if it is open) and responds with a 226 reply, indicating that the abort command was successfully processed.

In the second case, the server aborts the FTP service in progress and closes the data connection, returning a 426 reply to indicate that the service request terminated in abnormally. The server then sends a 226 reply, indicating that the abort command was successfully processed.

IPR2023-01353 Page 00041

DELETE (DELE)

This command causes the file specified in the pathname to be deleted at the server site. If an extra level of protection is desired (such as the query, "DO you really wish to delete?"), it should be provided by the user-FTP process.

CHANGE WORKING DIRECTORY (CWD)

This command allows the user to work with a different directory or dataset for file storage or retrieval without altering his login or accounting information. Transfer parameters are similarly unchanged. The argument is a pathname specifying a directory or other system dependent file group designator.

LIST (LIST)

This command causes a list to be sent from the server to the passive DTP. If the pathname specifies a directory, the server should transfer a list of files in the specified directory. If the pathname specifies a file then the server should send current information on the file. A null argument implies the user's current working or default

directory. The data transfer is over the data connection in type ASCII or type EBCDIC. (The user must ensure that the TYPE is appropriately ASCII or EBCDIC).

NAME-LIST (NLST)

This command causes a directory listing to be sent from server to user site. The pathname should specify a directory or other system-specific file group descriptor; a null argument implies the current directory. The server will return a stream of names of files and no other information. The data will be transferred in ASCII or EBCDIC type over the data connection as valid pathname strings separated by <CRLF> or <NL>. (Again the user must ensure that the TYPE is correct.)

SITE PARAMETERS (SITE)

This command is used by the server to provide services specific to his system that are essential to file transfer but not sufficiently universal to be included as commands in the protocol. The nature of these services and the specification of their syntax can be stated in a reply to the HELP SITE command.

STATUS (STAT)

This command shall cause a status response to be sent over the TELNET connection in the form of a reply. The command may be sent during a file transfer (along with the TELNET IP and Synch signals--see the Section on FTP Commands) in which case the server will respond with the status of the operation in progress, or it may be sent between file transfers. In the latter case the command may have an argument field. If the argument is a pathname, the command is analogous to the "list" command except that data shall be transferred over the TELNET connection. If a partial pathname is given, the server may respond with a list of file names or attributes associated with that specification. If no argument is given, the server should return general status information about the server FTP process. This should include current values of all transfer parameters and the status of connections.

HELP (HELP)

This command shall cause the server to send helpful

IPR2023-01353 Page 00042

information regarding its implementation status over the TELNET connection to the user. The command may take an argument (e.g., any command name) and return more specific information as a response. The reply is type 211 or 214. It is suggested that HELP be allowed before entering a USER command. The server may use this reply to specify site-dependent parameters, e.g., in response to HELP SITE.

NOOP (NOOP)

This command does not affect any parameters or previously entered commands. It specifies no action other than that the server send an OK reply.

The File Transfer Protocol follows the specifications of the TELNET protocol for all communications over the TELNET connection. Since, the language used for TELNET communication may be a negotiated option, all references in the next two sections will be to the "TELNET language" and the corresponding "TELNET end of line code". Currently one may take these to mean NVT-ASCII and <CRLF>. No other specifications of the TELNET protocol will be cited.

FTP commands are "TELNET strings" terminated by the "TELNET end of line code". The command codes themselves are alphabetic characters terminated by the character <SP> (Space) if parameters follow and TELNET-EOL otherwise. The command codes and the semantics of commands are described in this section; the detailed syntax of commands is specified in the Section on Commands, the reply sequences are discussed in the Section on Sequencing of Commands and Replies, and scenarios illustrating the use of commands are provided in the Section on Typical FTP Scenarios.

FTP commands may be partitioned as those specifying access-control identifiers, data transfer parameters, or FTP service requests. Certain commands (such as ABOR, STAT, QUIT) may be sent over the TELNET connection while a data transfer is in progress. Some servers may not be able to monitor the TELNET and data connections simultaneously, in which case some special action will be necessary to get the server's attention. The exact form of the "special action" is undefined; but the following ordered format is tentatively recommended:

1. User system inserts the TELNET "Interrupt Process" (IP) signal in the TELNET stream.

2. User system sends the TELNET "Synch" signal

3. User system inserts the command (e.g., ABOR) in the TELNET stream.

4. Server PI,, after receiving "IP", scans the TELNET stream for EXACTLY ONE FTP command.

(For other servers this may not be necessary but the actions listed above should have no unusual effect.)

FTP REPLIES

Replies to File Transfer Protocol commands are devised to ensure the synchronization of requests and actions in the process of file transfer, and to guarantee that the user process always knows the state of the Server. Every command must generate at least one reply, although there may be more than one; in the latter case, the multiple replies must be easily distinguished. In addition, some commands occur in sequential groups, such as USER, PASS and ACCT, or RNFR and RNTO. The replies show the existence of an intermediate state if all preceding commands have been successful. A failure at any point in the sequence necessitates the repetition of the entire sequence from the beginning.

The details of the command-reply sequence are made explicit in a set of state diagrams below.

An FTP reply consists of a three digit number (transmitted as three alphanumeric characters) followed by some text. The number is intended for use by automata to determine what state to enter next; the text is intended for the human user. It is intended that the three digits contain enough encoded information that the user-process (the User-PI) will not need to examine the text and may either discard it or pass it on to the user, as appropriate. In particular, the text may be server-dependent, so there are likely to be varying texts for each reply code.

Formally, a reply is defined to contain the 3-digit code, followed by Space <SP>, followed by one line of text (where some maximum line length has been specified), and terminated by the TELNET

IPR2023-01353 Page 00043

line length has been specified), and terminated by the TELNET
end-of-line code. There will be cases, however, where the text is
longer than a single line. In these cases the complete text must

be bracketed so the User-process knows when it may stop reading the reply (i.e. stop processing input on the TELNET connection) and go do other things. This requires a special format on the first line to indicate that more than one line is coming, and another on the last line to designate it as the last. At least one of these must contain the appropriate reply code to indicate the state of the transaction. To satisfy all factions it was decided that both the first and last line codes should be the same.

Thus the format for multi-line replies is that the first line will begin with the exact required reply code, followed immediately by a Hyphen, "-" (also known as Minus), followed by text. The last line will begin with the same code, followed immediately by Space <SP>, optionally some text, and the TELNET end-of-line code.

For example:

```
123-First line
Second line
  234 A line beginning with numbers
123 The last line
```

The user-process then simply needs to search for the second occurrence of the same reply code, followed by <SP> (Space), at the beginning of a line, and ignore all intermediary lines. If an intermediary line begins with a 3-digit number, the Server must pad the front to avoid confusion.

This scheme allows standard system routines to be used for reply information (such as for the STAT reply), with "artificial" first and last lines tacked on. In the rare cases where these routines are able to generate three digits and a Space at the beginning of any line, the beginning of each text line should be offset by some neutral text, like Space.

This scheme assumes that multi-line replies may not be nested. We have found that, in general, nesting of replies will not occur, except for random system messages (also called spontaneous replies) which may interrupt another reply. System messages (i.e. those not processed by the FTP server) will NOT carry reply codes and may occur anywhere in the command-reply sequence. They may be ignored by the User-process as they are only information for the human user.

The three digits of the reply each have a special significance. This is intended to allow a range of very simple to very sophisticated response by the user-process. The first digit denotes whether the response is good, bad or incomplete. (Referring to the state diagram) an unsophisticated user-process will be able to determine its next action (proceed as planned, redo, retrench, etc.) by simply examining this first digit. A user-process that wants to know approximately what kind of error occurred (e.g. file system error, command syntax error) may examine the second digit, reserving the third digit for the finest gradation of information (e.g. RNTO command without a preceding RNFR.)

There are five values for the first digit of the reply code:

1yz Positive Preliminary reply

The requested action is being initiated; expect another reply before proceeding with a new command. (The user-process sending another command before the completion reply would be in violation of protocol; but server-FTP processes should queue any commands that arrive while a preceding command is in progress.) This type of reply can be used to indicate that the command was accepted and the user-process may now pay attention

IPR2023-01353 Page 00044

to the data connections, for implementations where simultaneous monitoring is difficult.

2yz Positive Completion reply

The requested action has been successfully completed. A new request may be initiated.

3yz Positive Intermediate reply

The command has been accepted, but the requested action is being held in abeyance, pending receipt of further information. The user should send another command specifying this information. This reply is used in command sequence groups.

4yz Transient Negative Completion reply

The command was not accepted and the requested action did not take place, but the error condition is temporary and the action may be requested again. The user should

return to the beginning of the command sequence, if any. It is difficult to assign a meaning to "transient", particularly when two distinct sites (Server and User-processes) have to agree on the interpretation. Each reply in the 4yz category might have a slightly different time value, but the intent is that the user-process is encouraged to try again. A rule of thumb in determining if a reply fits into the 4yz or the 5yz (Permanent Negative) category is that replies are 4yz if the commands can be repeated without any change in command form or in properties of the User or Server (e.g. the command is spelled the same with the same arguments used; the user does not change his file access or user name; the server does not put up a new implementation.)

5yz Permanent Negative Completion reply

The command was not accepted and the requested action did not take place. The User-process is discouraged from repeating the exact request (in the same sequence). Even some "permanent" error conditions can be corrected, so the human user may want to direct his User-process to reinitiate the command sequence by direct action at some point in the future (e.g. after the spelling has been changed, or the user has altered his directory status.)

The following function groupings are encoded in the second digit:

x0z Syntax - These replies refer to syntax errors, syntactically correct commands that don't fit any functional category, unimplemented or superfluous commands.

x1z Information - These are replies to requests for information, such as status or help.

x2z Connections - Replies referring to the TELNET and data connections.

x3z Authentication and accounting - Replies for the login process and accounting procedures.

x4z Unspecified as yet

x5z File system - These replies indicate the status of the Server file system vis-a-vis the requested transfer or other file system action.

The third digit gives a finer gradation of meaning in each of the function categories, specified by the second digit. The

IPR2023-01353 Page 00045

list of replies below will illustrate this. Note that the text associated with each reply is recommended, rather than mandatory, and may even change according to the command with which it is associated. The reply codes, on the other hand, must strictly follow the specifications in the last section; that is, Server implementations should not invent new codes for situations that are only slightly different from the ones described here, but rather should adapt codes already defined.

A command such as TYPE or ALLO whose successful execution does not offer the user-process any new information will cause a 200 reply to be returned. If the command is not implemented by a particular Server-FTP process because it has no relevance to that computer system, for example ALLO at a TOPS20 site, a Positive Completion reply is still desired so that the simple User-process knows it can proceed with its course of action. A 202 reply is used in this case with, for example, the reply text: "No storage allocation necessary." If, on the other hand, the command requests a non-site-specific action and is unimplemented, the response is 502. A refinement of that is the 504 reply for a command that IS implemented, but that requests an unimplemented parameter.

Reply Codes by Function Groups

200 Command okay
500 Syntax error, command unrecognized
[This may include errors such as command line too long.]
501 Syntax error in parameters or arguments
202 Command not implemented, superfluous at this site.
502 Command not implemented
503 Bad sequence of commands
504 Command not implemented for that parameter

110 Restart marker reply.

In this case the text is exact and not left to the particular implementation; it must read:
MARK yyyy = mmmm
where yyyy is User-process data stream marker, and mmmm server's equivalent marker. (note the spaces between markers and "=".)
119 Terminal not available, will try mailbox.
211 System status, or system help reply
212 Directory status
213 File status
214 Help message
(on how to use the server or the meaning of a particular non-standard command. This reply is useful only to the human user.)
215 <scheme> is the preferred scheme.

120 Service ready in nnn minutes
220 Service ready for new user
221 Service closing TELNET connection
(logged out if appropriate)
421 Service not available, closing TELNET connection.
This may be a reply to any command if the service knows it must shut down.]
125 Data connection already open; transfer starting
225 Data connection open; no transfer in progress
425 Can't open data connection
226 Closing data connection;
requested file action successful (for example, file transfer or file abort.)
426 Connection closed; transfer aborted.
227 Entering Passive Mode. h1,h2,h3,h4,p1,p2

230 User logged in, proceed
530 Not logged in
331 User name okay, need password
332 Need account for login
532 Need account for storing files

150 File status okay; about to open data connection.
151 User not local; Will forward to <user>@<host>.
152 User Unknown; Mail will be forwarded by the operator.
250 Requested file action okay, completed.
350 Requested file action pending further information
450 Requested file action not taken:
file unavailable (e.g. file busy)
550 Requested action not taken:

IPR2023-01353 Page 00046

```
      file unavailable (e.g. file not found, no access)
   451 Requested action aborted: local error in processing
   551 Requested action aborted: page type unknown
   452 Requested action not taken:
      insufficient storage space in system
   552 Requested file action aborted:
      exceeded storage allocation (for current directory or
      dataset)
   553 Requested action not taken:
      file name not allowed
   354 Start mail input; end with <CR><LF>.<CR><LF>
```

Numeric Order List of Reply Codes

```
   110 Restart marker reply.
      In this case the text is exact and not left to the
      particular implementation; it must read:
           MARK yyyy = mmmm
      where yyyy is User-process data stream marker, and mmmm
      server's equivalent marker.  (note the spaces between
      markers and "=".)
   119 Terminal not available, will try mailbox.
   120 Service ready in nnn minutes
   125 Data connection already open; transfer starting
   150 File status okay; about to open data connection.
   151 User not local; Will forward to <user>@<host>.
   152 User Unknown; Mail will be forwarded by the operator.
   200 Command okay
   202 Command not implemented, superfluous at this site.
   211 System status, or system help reply
   212 Directory status
   213 File status
   214 Help message
      (on how to use the server or the meaning of a particular
      non-standard command.  This reply is useful only to the
      human user.)
   215 <scheme> is the preferred scheme.
   220 Service ready for new user
   221 Service closing TELNET connection
      (logged out if appropriate)
   225 Data connection open; no transfer in progress
   226 Closing data connection;
      requested file action successful (for example, file transfer
      or file abort.)
   227 Entering Passive Mode.  h1,h2,h3,h4,p1,p2
```



```
   230 User logged in, proceed
   250 Requested file action okay, completed.
   331 User name okay, need password
   332 Need account for login
   350 Requested file action pending further information
   354 Start mail input; end with <CR><LF>.<CR><LF>
   421 Service not available, closing TELNET connection.
      This may be a reply to any command if the service knows it
      must shut down.]
   425 Can't open data connection
   426 Connection closed; transfer aborted.
   450 Requested file action not taken:
      file unavailable (e.g. file busy)
   451 Requested action aborted: local error in processing
   452 Requested action not taken:
      insufficient storage space in system
   500 Syntax error, command unrecognized
      [This may include errors such as command line too long.]
   501 Syntax error in parameters or arguments
   502 Command not implemented
   503 Bad sequence of commands
   504 Command not implemented for that parameter
   530 Not logged in
   532 Need account for storing files
   550 Requested action not taken:
      file unavailable (e.g. file not found, no access)
   551 Requested action aborted: page type unknown
   552 Requested file action aborted:
```

IPR2023-01353 Page 00047

552 Requested file action aborted:
exceeded storage allocation (for current directory or dataset)
553 Requested action not taken:
file name not allowed

DECLARATIVE SPECIFICATIONS

MINIMUM IMPLEMENTATION

In order to make FTP workable without needless error messages, the following minimum implementation is required for all servers:

TYPE - ASCII Non-print
MODE - Stream
STRUCTURE - File, Record
COMMANDS - USER, QUIT, PORT,
TYPE, MODE, STRU,
for the default values
RETR, STOR,
NOOP.

The default values for transfer parameters are:

TYPE - ASCII Non-print
MODE - Stream
STRU - File

All Hosts must accept the above as the standard defaults.

CONNECTIONS

The server protocol interpreter shall "listen" on Port L. The user or user protocol interpreter shall initiate the full-duplex TELNET connection. Server- and user- processes should follow the conventions of the TELNET protocol as specified in the ARPA Internet Protocol Handbook. Servers are under no obligation to provide for editing of command lines and may specify that it be done in the user Host. The TELNET connection shall be closed by the server at the user's request after all transfers and replies are completed.

The user-DTP must "listen" on the specified data port; this may be the default user port (U) or a port specified in the PORT command. The server shall initiate the data connection from his own default data port (L-1) using the specified user data port. The direction of the transfer and the port used will be determined by the FTP service command.

When data is to be transferred between two servers, A and B (refer to Figure 2), the user-PI, C, sets up TELNET connections with both server-PI's. One of the servers, say A, is then sent a PASV command telling him to "listen" on his data port rather than initiate a connection when he receives a transfer service command. When the user-PI receives an acknowledgment to the PASV command, which includes the identity of the host and port being listened on, the user-PI then sends A's port, a, to B in a PORT command; a reply is returned. The user-PI may then send the corresponding

IPR2023-01353 Page 00048