

[TOOL] LogAgent, ASCII Log Monitor

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-06/0075.html>

From: support@securiteam.com

Date: 06/18/02

From: support@securiteam.com

To: list@securiteam.com

Date: Tue, 18 Jun 2002 20:32:50 +0200 (CEST)

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

LogAgent, ASCII Log Monitor

DETAILS

LogAgent is a piece of software made in Perl designed to monitor ASCII log files and redirect any change made to it to a central location. The purpose of this is to add flexibility in some security (or other) applications on the choice of destination folder for the log files. The ability to specify your own destination folder for log files could be a crucial requirement in your specification for a security software, and good products can be overlooked simply because they lack this single feature. LogAgent tries to fill that gap by monitoring the log files on the local machine, and then redirects the last line of the log file (as a modification is assumed to be an addition to the file made by the associated software, more about this later) to the destination of your choice, either on another folder on the same machine or to a remote server for network-wide log file centralization.

Source code:

log20en.pl

#! C:\perl\bin\perl.exe

LogAgent 2.0 beta

#####

LogAgent 2.0 beta

by Floydman floydian_99@yahoo.com

Securiteam: [TOOL] LogAgent, ASCII Log Monitor

```
# Copyright 2002 SecurIT Informatique Inc. http://securit.iquebec.com #
##
# This program gets its configuration from the file config.txt, and the
list of #
# directories to be monitored from the file mondir.txt. These two files
have to be in #
# the same directory as LogAgent. The config file lets you specify if you
want to #
# include the IP of the machine, the hostname and the username in the log
files, in #
# these cases where the software generating the log doesn't provide these
credentials. #
# You can also specify if you want to display entries captured by LogAgent
to be #
# displayed on the console or not. Then, the program starts the monitoring
threads for #
# each entry in mondir.txt, and then enters in an infinite-loop, waiting
for signals #
# from the threads. When a signal is triggered (ie: a file as changed in
the directory #
# you are monitoring), it gets the last line from the log file, and sends
it to the #
# specified outputs. Output dirs can be remote or local, and as many as
you want. #
##
# note about config.txt: Do not modify the headers LOGIP, LOGHOST, LOGUSER
and #
# SHOWCONSOLE, or the program will stop working. Only change the Y or N at
the end of #
# the line. #
#####

#####
# LICENSE #
# This software is Open Source. This means that its source code is open,
free and avai-#
# lable for anyone to look into, make modifications, correct bugs (let me
know, please) #
# and use for their personal use. This is a beta version, so this software
is NOT for #
# commercial use. You can create your own binaries, provided you
rightfully own a #
# compiler (if you don't, then you are stealing them, not me), and to
distribute it in- #
# side your organisation for internal usage only. DO NOT distribute
compiled copies of #
# this software to external parties other than the one you work for. If
you wish to be #
# a licensed distributor for the final version, send an e-mail to
securit.iquebec.com. #
#####
```

Securiteam: [TOOL] LogAgent, ASCII Log Monitor

```
#####  
# Main Program #  
# This is the main structure of LogAgent. #  
# This procedure takes note of the machine credentials, LogAgent's  
configuration and #  
# the list of directories to monitor. #  
# Then, we start a thread for each entry in mondir.txt and we enter in the  
main loop. #  
# This loop waits for signals from the threads, and when a signal is  
received, it #  
# captures the last line of the modified log file. This line is then sent  
to the #  
# various outputs specified in config.txt. #  
# At the end of the loop (CTRL-C) we destroy our threads and memory  
objects, for clean #  
# programming purposes. #  
#####  
  
# Using Win32::AdvNotify  
# By Amine Moulay Ramdane <aminergeneration.net>  
# Website: http://www.generation.net/~aminergeneration.net/Perl/  
# This Perl module is the core engine of LogAgent. This module contains  
all the functionalities  
# For monitoring the changes made to files and folders on the system  
# You will also need to install the Win32 API Perl module in order to use  
AdvNotify  
  
use Win32::AdvNotify qw(FILE_NAME SIZE INFINITE Yes No  
All %ActionName %ActionColor);  
  
# Declaration of needed components for machine identification  
use Socket;  
use Sys::Hostname;  
my $element;  
  
# Creation of the AdvNotify object  
my $obj = new Win32::AdvNotify() || die "Can't create object\n";  
  
# Creation of machine ID table  
@id = getid();  
  
# Creation of config table  
my @config = getconfig();  
  
# Creation of mondir table  
my @mondir = getmondir();  
  
# Creation of threads table. Threads are started, and then launched, this  
is the way the AdvNotify module works  
my $index=0;  
foreach $element (@mondir)
```

Securiteam: [TOOL] LogAgent, ASCII Log Monitor

```
{
  $threads[$index] = $obj->StartThread(Directory => $mondir[$index],
    Filter => All ,
    WatchSubtree => No ) || die "Can't start
thread\n";
  $threads[$index]->EnableWatch() || die "Problem starting
EnableWatch()\n";
  $index++;
}

print "Log Agent 2.0, brought to you by Floydman\n";
print "Copyright 2002 SecurIT Informatique Inc.\n";
print "http://securit.iquebec.com\n";

# Enters the main monitoring loop
startmonitoringloop();

# termination of the threads.
for ($a; $a<$index; $a++)
  { $threads[$a]->Terminate(); }

# destruction of the object
undef $obj;

# End of program#

#####
# procedure getid() #
# This procedure gets the IP address, the host name and the username of
the machine. #
#####

sub getid
{
  # Define username, IP address and hostname of the local machine
  my $addr = inet_ntoa(scalar(gethostbyname($name)) || 'localhost');
  my $host = hostname() || "hostname not defined";
  my $login = getlogin || getpwuid($<) || "not logged";
  my @id_table = ($addr, $host, $login);
  return (@id_table);
}

#####
# procedure getconfig() #
# This procedure gets the configuration file config.txt. #
#####

sub getconfig
{
  my @configtable;
```

Securiteam: [TOOL] LogAgent, ASCII Log Monitor

```
my @dirtable;
$index = 0;
$numarg = 0;

open(CONFIGFILE,"<config.txt") || die "Can't open config.txt";
$logip = <CONFIGFILE> || die "Can't read logip from config.txt";
($logip=~m/LOGIP/i) || die "LOGIP entry missing in config.txt";

$loghost = <CONFIGFILE> || die "Can't read loghost from config.txt";
($loghost=~m/LOGHOST/i) || die "LOGHOST entry missing in config.txt";

$loguser = <CONFIGFILE> || die "Can't read loguser from config.txt";
($loguser=~m/LOGUSER/i) || die "LOGUSER entry missing in config.txt";

$showconsole = <CONFIGFILE> || die "Can't showconsole read from
config.txt";
($showconsole=~m/SHOWCONSOLE/i) || die "SHOWCONSOLE entry missing in
config.txt";

while (defined($dir = <CONFIGFILE>))
{
    $dirtable[$index]=$dir;
    $index++;
}
($index==0) && die "No destination directory specified in config.txt.";
close (CONFIGFILE) || die "Can't close config.txt";

@configtable = ($logip, $loghost, $loguser, $showconsole, @dirtable);
@configtable = parse(@configtable);

(($numarg=@configtable)<5) && die "Not enough parameters in config.txt.
Check file for errors.";

# Transformation of the first 4 lines of configtable to boolean value
$configtable[0]=$configtable[0]=~m/Y/i;
$configtable[1]=$configtable[1]=~m/Y/i;
$configtable[2]=$configtable[2]=~m/Y/i;
$configtable[3]=$configtable[3]=~m/Y/i;

return (@configtable);
}

#####
# procedure getmondir() #
# This procedure gets the configuration file mondir.txt. #
#####

sub getmondir
{
my @dirtable;
my $index = 0;
```

Securiteam: [TOOL] LogAgent, ASCII Log Monitor

```
open(MONDIRFILE,"<mondir.txt") || die "Can't open mondir.txt";
while (defined($dir = <MONDIRFILE>))
{
    $dirtable[$index]=$dir;
    $index++;
}
($index==0) && die "No destination directory specified in config.txt.";
close (MONDIRFILE) || die "Can't close mondir.txt";

@dirtable = parse(@dirtable);

return (@dirtable);
}

#####
# procedure parse(table_file) #
# This procedure cleans the files from non-valid and blank characters that
# could be #
# placed in the config files. The procedure returns the file as a table. #
#####

sub parse
{ my (@table) = @_ ;

#check for invalid characters in table_file
chomp @table;

foreach $element (@table)
{
    $element=~s%^\s+% ;
    @char = split (//, $element);

    foreach $char (@char)
    { $char=~s%\\%/% ; }
    $element = join ("",@char);
}

my @tabletemp;
my $index = 0;

foreach $element (@table)
{
    if ($element ne "") {
    $tabletemp[$index]=$element;
    $index++;}
}

@table = @tabletemp;
return (@table);
}
```

Securiteam: [TOOL] LogAgent, ASCII Log Monitor

```
#####
# procedure startmonitoringloop() #
# This procedure is the main monitoring loop. When a change is detected in
# a file #
# located in a monitored directory (preferably ASCII files), the procedure
# calls #
# getlastline() with the name of the modified file. The captured line is
# then sent #
# via the procedure sendoutput(), along with LogAgent's configuration
# table. #
#####

sub startmonitoringloop
{

while($threads[0]->Wait(INFINITE))# exit with [Ctrl-C] signal
{
  while($threads[0]->Read(\@data))# exit when the list is empty
  {
    for($i=0;$i<=$#data;$i++)
    {
      @line=getlastline($data[$i]);
      sendoutput(@line,@config);
    }
  }
}
}

#####
# procedure getlastline(filename) #
# This procedure gets the last line (non-blank) of the file received as
# the argument. #
# It returns the filename (whitout path) and the last line of the file. #
#####

sub getlastline
{my ($data) = @_;

open (LOGFILE, $data->{Directory}.$data->{FileName}) or die "Can't open
log file";
flock (LOGFILE, 1) or die "Can't lock file";
@lines = <LOGFILE>;
close (LOGFILE) or die "Can't close file"; # To unlock the file as fast as
possible for new entries

@lines = parse(@lines);
$lastline = $lines[-1];

return ($data->{FileName}, $lastline);
}
```

Securiteam: [TOOL] LogAgent, ASCII Log Monitor

```
#####  
# procedure sendoutput(line, config) #  
# This procedure receives as arguments: the name of the modified file, the  
last line #  
# of the logfile, and then the config table (LOGIP, LOGHOST, LOGUSER,  
SHOWCONSOLE, and #  
# the various destination directories). The procedure checks the  
configuration to see #  
# if it has to append any information to the original line or not. If  
SHOWCONSOLE in #  
# enabled, then the line is printed on the screen, if not it simply passes  
to the next #  
# step which is to forward this line to all mentioned destinations in  
config.txt. #  
#####
```

```
sub sendoutput  
{ my ($filename, $line, $logip, $loghost, $loguser, $showconsole, @dest) =  
@_;  
  
my $newline="";  
  
if ($logip) {$newline=$newline.$id[0]." "};  
if ($loghost) {$newline=$newline.$id[1]." "};  
if ($loguser) {$newline=$newline.$id[2]." "};  
  
$newline=$newline.$line;  
  
if ($showconsole) {print "$newline\n";}  
  
foreach $destdir (@dest)  
{  
  $destination=$destdir.$filename;  
  open (DEST, ">>".$destination) || die "Can't open master log file  
$destination";  
  flock (DEST, 2) || die "Can't lock file for writing";  
  print DEST "$newline\n" || die "Can't write to file";  
  close (DEST) || die "Can't close master log file";  
}  
}  
  
#EOF
```

6. Sample config.txt

```
LOGIP=Y  
LOGHOST=N  
LOGUSER=Y  
SHOWCONSOLE=N  
D:\log  
\\logserver1\shared_floder\  
\\logserver2\hidden_share$\
```

7. sample mondir.txt
D:\Winnt\Internet Logs\
D:\Program Files\Antivirus Software\Log\
C:\Download Manager\Log\

ADDITIONAL INFORMATION

The tool can be also downloaded from:
<http://www.geocities.com/floydian_99/logagnt20beta.txt>
http://www.geocities.com/floydian_99/logagnt20beta.txt

The information has been provided by <mailto:floydian_99@yahoo.com>
Floydman.

=====

This bulletin is sent to members of the SecuriTeam mailing list.
To unsubscribe from the list, send mail with an empty subject line and body to:
list-unsubscribe@securiteam.com
In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.
In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

-
- *Previous message:* support@securiteam.com: "[NEWS] Cable Modem Termination System Authentication Bypass"
 - *Messages sorted by:* [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)