

[TOOL] Pixparser – Cisco PIX Firewall Configuration File Parser

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-10/0024.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 10/08/03

To: list@securiteam.com

Date: 8 Oct 2003 10:23:46 +0200

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

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

Pixparser – Cisco PIX Firewall Configuration File Parser

DETAILS

Pixparser takes a Cisco PIX firewall configuration, parses the rules and output this in an easy to read format.

Tool source:

```
#!/C:/Perl/Bin/perl -w
```

```
# Author: Wiseman (wiseman@spray.se)
```

```
# Filename: pixparser.pl
```

```
# Current Version: 0.7 beta
```

```
# Created: 28th of March 2003
```

```
# Last Changed: 7th of October 2003
```

```
#
```

```
# Description:
```

```
# -----
```

```
# This is a rather crude and quick hacked Perl-script to extract the
```

```
# access-lists from a Cisco PIX and make them more read-able
```

```
# This is beta and probably will stay that way for some time!
```

```
#
```

Securiteam: [TOOL] Pixparser – Cisco PIX Firewall Configuration File Parser

```
# Known issues:
# -----
# 1. A line like "access-list inbound_inet permit icmp any any echo-reply"
# will pass but the echo-reply won't be parsed as echo-reply but "All
ports"
# Will fix this later bro!
#
# -----
# Change History
# -----
# 0.7 beta: (7th of October 2003)
# Found some bugs regarding RegEX-expressions. Forgot to anchor the
# pattern to the beginning with the ^sign.
#
# 0.6 beta (March 2003):
# First working version, still buggy

use strict;

#####
### Declare local variables start
#####

my $temp;
my $temp2;
my $temp3;
my $filler = " ";
my $numberofargs = @ARGV;
my $filename = $ARGV[1];

my @interfaces;
my @access_groups;
my @access_lists;
my @sorted_list;
my $subset;
my $details;
my $junk;

my $if_type;
my $if_name;
my $if_securitylevel;

my $ag_name;
my $ag_direction;
my $ag_bound_to_if_name;

my $al_name;
my $al_ip_udp_tcp;
my $al_deny_or_permit;
my $al_source;
my $al_dest;
```

```
my $al_source_dest;
my $al_temp;
my $al_port;
my $al_port_part1;
my $al_port_part2;

#####
### Declare local variables end
#####

#####
### Input parser and Syntax start
#####

print "\n==| pixparser.pl v. 0.7 beta (7th of October 2003) by Wiseman
\wiseman@spray.se\ |==\n";

if ($numberofargs < 2) {

    print "\nSyntax:\n";
    print "-----\n";
    print "pixparser.pl <Input filename> <Output filename>\n";
    print "\n";
    print "Mandatory arguments:\n";
    print "-----\n";
    print " <Input filename> : Name of inputfile. This file should contain
the Cisco Pix configuration\n";
    print " <Output filename> : Name of outputfile\n";
    die ("\n");

} # End if

#####
### Input parser and Syntax end
#####

### Open input-file and put the contents in one HUGE array

    open (PARSEFILE,$ARGV[0]) || die ("==| Error! Could not open file
$ARGV[0]");

    print "\nLoading PIX Configuration-file from $ARGV[0]...";

    my @Parse_array = <PARSEFILE>;
    my $Parsefile_size = @Parse_array;
    print "Done\n";

    close (PARSEFILE);

#####
### Setup start
```

Securiteam: [TOOL] Pixparser – Cisco PIX Firewall Configuration File Parser

```
#####  
#  
# When Setup is finished the Interfaces, Access-groups and Access-lists  
# will end up in a array of its own. Bug-fix here! Forgot the ^ at first!  
  
# Parse Interfaces (if)  
  
foreach $temp (@Parse_array) {  
  if ($temp =~ (m/^nameif/i)) {  
    push (@interfaces, $temp);  
  } # End if  
} # End foreach  
  
# Parse Access groups  
  
foreach $temp (@Parse_array) {  
  if ($temp =~ (m/^access-group/i)) {  
    push (@access_groups, $temp);  
  } # End if  
} # End foreach  
  
# Parse Access lists  
  
foreach $temp (@Parse_array) {  
  if ($temp =~ (m/^access-list/i)) {  
    push (@access_lists, $temp);  
  } # End if  
} # End foreach  
  
#####  
### Setup end  
#####  
  
#####  
### Parsing start  
#####  
  
# This is foreach 1  
foreach $temp (@interfaces) {  
  
# Pattern memory:  
# $1 = interface-type, for instance "ethernet0"  
# $2 = interface-name, for instance "outside"  
# $3 = interface-securitylevel, for instance security100  
  
$temp =~ (m/\bnameif\b\s+(\b\S+\b)\s+(\b\S+\b)\s+(\b\S+\b)/i);  
  
$if_type = $1;  
$if_name = $2;  
$if_securitylevel=$3;
```

Securiteam: [TOOL] Pixparser – Cisco PIX Firewall Configuration File Parser

```
($junk, $if_securitylevel) = split(/security/, $if_securitylevel);

# Populate $details with a more easy to read info

    $details = "\n\n====| Interface-name:$if_name,
Interface-type:$if_type, Security-Level:$if_securitylevel |====\n";

# Push the hit, ie the interface and its details onto the list
    push (@sorted_list, $details);

#This is foreach 2
#
    foreach $temp2 (@access_groups) {

# Find the correct access-group to bind to the interface

# Pattern memory:
# $1 = access-group name, for instance "inbound"
# $2 = access-group rule apply direction, for instance "in" or "out"
# $3 = access-group interface name, for instance "inside"

    $temp2 =~
(m/\baccess-group\b\s+(\bS+\b)\s+(\bS+\b)\s+\binterface\b\s+(\bS+\b)/i);

    $ag_name = $1;
    $ag_direction = $2;
    $ag_bound_to_if_name = $3;

# Check if this access-group is indeed linked to the Interface

# This is if AG
    if ($if_name eq $ag_bound_to_if_name ) {

# Populate $details with a more easy to read info

# $details = "====| Access group name:$ag_name, Bound to
Interface:$ag_bound_to_if_name, Direction of rules is $ag_direction-bound
|====\n\n";
    $details = "====| Access Group name:$ag_name, Direction of rules is
$ag_direction-bound |====\n\n";

# Push the hit, ie the access list and its details onto the list

    push (@sorted_list, $details);

# Push headings onto the array
#
    $details = "|Source| |Destination| |Protocol| |Port| |Action|\n";
    push (@sorted_list, $details);
```

Securiteam: [TOOL] Pixparser – Cisco PIX Firewall Configuration File Parser

```
#This is foreach 3
foreach $temp3 (@access_lists) {

    $temp3 =~
(m/^\baccess-list\b\s+(\b\S+\b)\s+(\b\S+\b)\s+(\b\S+\b)\s+(.+)/i);

    $al_name = $1;
    $al_deny_or_permit = $2;
    $al_ip_udp_tcp = $3;
    $al_source_dest = $4;

# Check if this access-list is indeed linked to the access_group

    if ($al_name eq $ag_name) {

        $al_source_dest =~
(m/(\bany\b|\bhost\b\s+\d+\.\d+\.\d+\.\d+|\d+\.\d+\.\d+\.\d+|\d+\.\d+\.\d+\.\d+)\s+(.*)/i);

        $al_source = $1 ;
        $al_dest = $2;

# The string in $al_dest now contains the destination but *also* the port
(if any)
# Let's extract the port too!

# First we must check whether there indeed is a eq, gt or range in the
input!

        if ($al_dest =~ (m/^\beq\b/i)) {
            ($al_dest, $al_port) = split(/eq/, $al_dest);

        } elsif ($al_dest =~ (m/^\bgt\b/i)) {
            ($al_dest, $al_port) = split(/gt/, $al_dest);
            $al_port = ">$al_port";

        } elsif ($al_dest =~ (m/^\brange\b/i)) {
            ($al_dest, $al_port) = split(/range\s+/, $al_dest);
            ($al_port_part1, $al_port_part2) = split(/\s+/, $al_port);
            $al_port = " range $al_port_part1-$al_port_part2";
        } else {
            $al_port = " All ports";
        }

    } # End if

# Pad with $filler so the output is reaaallyyy nice!

    $al_source .= $filler x (32-length($al_source));
    $al_dest .= $filler x (32-length($al_dest));
    $al_port .= $filler x (20-length($al_port));
# $al_deny_or_permit .= $filler x (27-length($al_deny_or_permit));
    $al_ip_udp_tcp .= $filler x (12-length($al_ip_udp_tcp));
```

Securiteam: [TOOL] Pixparser – Cisco PIX Firewall Configuration File Parser

```
# Make "deny" uppercase

if ($al_deny_or_permit =~ (m/deny/i)) {
    $al_deny_or_permit = uc($al_deny_or_permit);
}

# Populate $details with a more easy to read info

# $details = "Source:$al_source Destination:$al_dest
Protocol:$al_ip_udp_tcp Port:$al_port Action:$al_deny_or_permit\n";
$details = " $al_source $al_dest $al_ip_udp_tcp $al_port
$al_deny_or_permit\n";

# Push the hit, ie the access list and its details onto the list

    push (@sorted_list, $details);

} # End if

} # End foreach 3

} # End if AG

} # End foreach 2

} # End foreach 1

# foreach $temp (@sorted_list) {
# print $temp;
# } # end foreach

#####
### Parsing end
#####

#####
### Save to file start
#####

#my $filename = $ARGV[1];

open(OUTFILE, ">$filename");
print "Saving output to $filename...";
print OUTFILE "====| pixparser.pl v. 0.6 beta by Wiseman
\"wiseman\"@spray.se\" |====";

foreach $temp (@sorted_list) {
    print OUTFILE $temp;
} # end foreach
print "done\n";
close (OUTFILE);
```

```
#####  
### Save to file end  
#####
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:wiseman@spray.se> Wiseman.

=====

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.