Beyond-Security's SecuriTeam.com

SecuriTeam Home
About SecuriTeam
Ask the Team
Advertising info
Security News
Security Reviews
Exploits
Tools
UNIX focus
Windows NT focus

1. SADoor, Non-Listening Remote Administration Tool
2. ShellForge, Python Based Shellcode Generator
3. IMAP-ftp, IMAP Based FTP Like Access
4. HOON, an AT&T to Shellcode
5. THC-Vmap, Version Mapper Tool

E-Mail this article to a friend
Send us comments

Tool Name **13/8/2003**

IMAP-ftp, IMAP Based FTP Like Access

Details

Tool:

```
#!/usr/bin/perl
#
# IMAP-ftp v0.9b by jfs@gibnet.gi
#
# IMAP has more to offer than just overflows.
#
# Most people consider POP3 and IMAP as the same thing.
However, IMAP is
# much more powerful, so you get to subscribe to folders
over the filesystem
# and collaborate with other users in the system.
#
# In advance you can't tell an imap folder from a normal file,
nor does IMAP
# (or tries to for that matter), so you can subscribe and fetch
messages
# from "normal" files, as this is the way IMAP works.
#
# This tool makes accessing files through IMAP easier, just
enter HELP
# at the prompt for a list of commands (or scroll down a few
lines).
#
# Nothing new here, just a heads up for sysadmins who care
about security
# and didn't know :)
#
# And remember, it's not a bug, it's a feature.
#
# For more details, check RFC 2060: "INTERNET MESSAGE
ACCESS PROTOCOL - VERSION 4rev1"
#
# Send any comments/addons to jfs@gibnet.gi
#
#
#IMAP tiene más para ofrecer que algunos overflows.
#
#La mayoría de la gente considera POP3 y IMAP como la
```

```
misma cosa. Sin embargo,
#IMAP es mucho más poderoso, usted puede suscribirse a
carpetas sobre el sistema
#de archivos y colaborar con otros usuarios en el sistema.
#
#Usted no puede llamar por adelantado una carpeta IMAP
desde un archivo normal,
#ni el IMAP (o intentos para a esa tema), así que usted
puede suscribir y traer
#mensajes de archivos "normales", que es como el IMAP
trabaja.
#
#Esta herramienta hace más fácil acceder a archivos atreves
del IMAP, incorpora
#una AYUDA con una lista de comandos ya veréis su
sencillez.
#
#Nada nuevo aquí, simplemente que lo tengan en cuenta los
administradores de
#sistemas, que cuiden su seguridad si no lo sabían :)
#
#Y recuerde, esto no es un bug, es una función.
#
#Para más detalles consulten, el RFC 2060: "INTERNET
MESSAGE ACCESS PROTOCOL - VERSION 4rev1"
#
#Envíe cualquier comentario a jfs@gibnet.gi

use Socket;
use IO::File;

my ($remote,$port, $iaddr, $paddr, $proto, $line);

$remote=@ARGV[0] || die("usage: iftp host[:port] [username]
[password]\n");

if($remote=~/(.*)\:(\d+)/)
  {
  $remote=$1;
  $port=$2;
  }
else
  {
  $port=143;
  }
$user=@ARGV[1];
$pass=@ARGV[2];

# Initialization

$|=1;
```

```
$cmd_id=0;

$proto = getprotobyname('tcp');

# Get arguments

$iaddr = inet_aton($remote) || die "no host: $remote";
$paddr = sockaddr_in($port, $iaddr);

socket(SOCK, PF_INET, SOCK_STREAM, $proto) || die
"socket: $!";

autoflush SOCK 1;

connect(SOCK, $paddr) || die "connect: $!";


$line=<SOCK>; # <* OK localhost IMAP4rev1 v12.250 server
ready>

if($line=~/^\* OK\s+([^\s]*)/)
  {
  print STDOUT "Connected to $1.\n";
  print STDOUT "$line";
  }
else
  {
  print STDOUT "Protocol error:\n$line";
  exit(1);
  }

# Login

if(! $user)
  {
  print "Username: ";
  $user=<STDIN>;
  chop($user);
  }

if(! $pass)
  {
  print STDOUT "Password required for $user.\n";
  print STDOUT "Password: ";
  $pass=<STDIN>;
  chop($pass);
  }

imap_send ("LOGIN $user $pass");

$line=<SOCK>; # 1 OK LOGIN completed
```

```
if($line=~/^\d+ OK/)
  {
  print STDOUT "Login successful as $user.\n";
  }
else
  {
  print STDOUT "Login incorrect/LOGIN not
supported:\n$line";
  exit(1);
  }


die "can't fork: $!" unless defined($kidpid = fork());

if ($kidpid) {
   # copy the socket to standard output
   while (defined ($line = <SOCK>)) {

     if($line!~/^\*/)
       {
      print STDOUT $line;
       }
     else
       {
      if($line=~/\* LIST \(.*\)\s+\".*\"\s+(.*)$/)
        {
  print "$1\n";
        }
       }
     }

kill("TERM", $kidpid); # send SIGTERM to child
   }
   # the else{} block runs only in the child process
else {
   # copy standard input to the socket
   print STDOUT "iftp> ";
   while (defined ($line = <STDIN>)) {

  # Parse commands
  if($line=~/\s*CAT\s+(.*)\s*$/i)
    {
        $file=$1;
    imap_send("SELECT $1");
    imap_send("FETCH 1 BODY[TEXT]");
    imap_send("CLOSE");
    }
  elsif($line=~/\s*PUT\s+([^\s]+)\s+([^\s]+)\s*$/i)
    {
        $from=$1;
    $to=$2;
  open(FILE,"< $from") || next;
  @file_content=<FILE>;
  close(FILE);

  $file_content=join("",@file_content);
  $file_len=length($file_content);

  imap_send("CREATE $to");
  imap_send("APPEND $to \{$file_len\}");
  print SOCK $file_content;
  print SOCK "\n";
  }
elsif($line=~/\s*TOUCH\s+([^\s]+)\s*$/i)
  {
        $from=$1;
  imap_send("CREATE $from");
  }
elsif($line=~/\s*APPEND\s+([^\s]+)\s+([^\s]+)\s*$/i)
  {
        $from=$1;
  $to=$2;

  open(FILE,"< $from") || next;
  @file_content=<FILE>;
  close(FILE);

  $file_content=join("",@file_content);
  $file_len=length($file_content);

  imap_send("APPEND $to \{$file_len\}");
  print SOCK $file_content;
  print SOCK "\n";
  }
elsif($line=~/\s*CP\s+([^\s]+)\s+([^\s]+)\s*$/i)
  {
        $from=$1;
  $to=$2;

  imap_send("CREATE $to");
  imap_send("SELECT $from");
  imap_send("COPY 1 $to");
  }
elsif($line=~/\s*LS\s+([^\s]+)$/i)
  {
  $dir=$1;
  imap_send("LIST \"$dir\" \"*\"");
  }
elsif($line=~/\s*HELP/i)
  {
  print STDOUT "IMAP-ftp v0.9b / jfs\@gibnet.gi\n";
```

```
    print STDOUT "Commands are:\n\n";
    print STDOUT "cat remote_file - shows remote file contents\n";
    print STDOUT "put local_file remote_file - uploads a file\n";
    print STDOUT "append local_file remote_file - appends a local file to an existing remote file\n";
    print STDOUT "cp remote_file remote_file - copies files remotely\n";
    print STDOUT "ls remote_file - lists a filename/directory\n";
    print STDOUT "touch remote_file - creates a remote file\n";
    }
  else
    {
    chop($line);
    if($line)
      {
   print STDOUT "?Invalid command\n";
      }
    }
   print STDOUT "iftp> ";

     }
   }

close(SOCK);

sub imap_send {
local($my_line)=@_;
$cmd_id++;
print SOCK "$cmd_id $my_line\n";
}
```

Links

The information has been provided by Mentes Inquietas - Restless Minds.