RCE in F*EX

Posted on July 8, 2020

Background

F*EX is a Perl-based HTTP file exchange service. Quoting from the vendor’s homepage:

> F*EX (Frams’ Fast File EXchange) is a service to send big (large, huge, giant, …) files from a user A to a user B. The sender uploads the file to the F*EX server using a WWW upload form and the recipient automatically gets a notification e-mail with a download-URL.

Issue Description

While reviewing the F*EX implementation, the function `copy` from `lib/fex.pp` was analyzed:

```
# copy file (and modify) or symlink
# returns chomped file contents or link name
# preserves permissions and time stamps
sub copy {
  my ($from,$to,$mod) = @_;
  my $link;
  local $/;
  local $_;

  $to .= '/'.basename($from) if -d $to;

  if (defined($link = readlink $from)) {
    mksymlink($to,$link);
    return $link;
  } else {
    open $from,'<',$from or return;
    open $to,'>',$to or return;
    $_ = <$from>;
    close $from;
    eval $mod if $mod;
    print {$to} $_;
    close $to or http_die("internal error: $to - $!");
    if (my @s = stat($from)) {
      chmod $s[2],$to;
      utime @s[8,9],$to unless $mod;
    }
    chomp;
    return $_;
  }
}
```

The `eval $mod if $mod` call indicates a potential `eval` injection issue. Identifying the callers reveals that the `copy` function is invoked by `bintar` from `bin/fexsrv`, which is shown below.

```
sub bintar {
  my $tmpdir = "$FEXHOME/tmp";
  my $fs = "$ENV{PROTO}://$ENV{HTTP_HOST}";

  if (chdir "$FEXHOME/bin") {
    fexlog($connect,@log);
    chdir $fstb if $fstb;
    mkdir $tmpdir;
    foreach my $f (@_) {
      copy($f,"$tmpdir/$f","s#fexserver = ''#fexserver = '$fs'#");
      chmod 0755,"$tmpdir/$f";
    }
    chdir $tmpdir or http_die("internal error: $tmpdir - $!");
    my $tar = `tar cf - @_ 2>/dev/null`;
    unlink @_;
    nvt_print(
      'HTTP/1.1 200 OK',
      'Server: fexsrv',
      "Content-Length: ".length($tar),
      "Content-Type: application/x-tar",
      '',
    );
    print $tar;
    exit;
  }
}
```

It can be observed that in this call, the `$mod` argument is indeed passed to `copy`. Parts of the `$mod` argument are based on the `HTTP_HOST` variable, which is user-controlled. Further tracing down callers of `bintar` yields the following code from `bin/fexsrv`, which is part of the HTTP request parsing logic:

```
      # special request for F*EX UNIX clients
      if ($ENV{SCRIPT_NAME} eq 'xx.tar') {
        bintar(qw'fexget fexsend xx zz ezz');
      }
```

It should be noted that no authentication is required in order to trigger this code path. The vulnerability hence is a pre-auth RCE issue.

Fix

The issue has been fixed in `fex-20160919_2`.

Credit

Gregor Kopf of Secfault Security GmbH

Disclaimer

The information provided is released “as is” without warranty of any kind. The publisher disclaims all warranties, either express or implied, including all warranties of merchantability. No responsibility is taken for the correctness of this information. In no event shall the publisher be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages, even if the publisher has been advised of the possibility of such damages.

The contents of this advisory are copyright (c) 2020 Secfault Security GmbH and may be distributed freely