Preventing upload of EXE files

Some SFTP servers feature a simple “extension exclusion list” so that administrators can specify certain file extensions that the server should not let users upload. But that’s a pretty weak defense, as a clever attacker could always upload an EXE with a fake extension and then rename it or otherwise find alternative ways to run it on the server, thus compromising its security.

Syncplify Server!’s scriptable nature, though, allows you to do a lot more than just disallow certain file extensions. Here’s a sample script that can be attached to the “**AfterFileUpload**” event handler, to identify EXE files that have been uploaded with fake extensions and delete them right away.

```
{
  var FirstBytes = FileReadAsHex(Session.GetAbsPath(), 0, 2);
  var PEBytes := FileReadAsHex(Session.GetAbsPath(), 256, 4);
  if ((FirstBytes == '4D5A') && (PEBytes == '50450000')) {
    // It's an EXE, delete it!
    Log('Identified '+Session.GetAbsPath()+' as an EXE file, deleting it.');
    if DelFile(Session.GetAbsPath()) {
      Log('Deleted: '+Session.GetAbsPath());
    } else {
      Log('Failed to delete: '+Session.GetAbsPath());
    }
  }
}
```

The above script is provided as a mere example to identify Windows EXE files. But it could be easily modified in order to identify other file types.

All Windows EXEs, in fact, have stable distinguishing features in their binary code, and more precisely: the first 2 bytes (in hex) will always be **4D5A**, and the 4 bytes at offset 256 (0x100) will always be **50450000**. So if a file has those byte sequences in those exact locations, it’s safe to say it’s a Windows EXE.

Do you need to identify ZIP files instead? The first 4 bytes are always **04034B50**.

And so on… many file types can be identified by specific *“signatures”* in their binary code, that one can easily read using Syncplify Server!’s powerful scripting capabilities.

Alternative method

Starting with version 6, Syncplify Server! has also added a handy FileType function to its scripting engine. This function automatically identifies the MIME-Type of hundreds of file types by reading only the first 261 (at most) bytes from the file itself.

The above script could then be rewritten like this:

```
{
  if (FileType(Session.GetAbsPath()) == "application/x-msdownload") {
    // It's a Windows EXE, delete it!
    Log('Identified '+Session.GetAbsPath()+' as an EXE file, deleting it.');
    if DelFile(Session.GetAbsPath()) {
      Log('Deleted: '+Session.GetAbsPath());
    } else {
      Log('Failed to delete: '+Session.GetAbsPath());
    }
  }
}
```

Revision #1
Created 22 July 2023 16:39:32 by DevTeam
Updated 22 July 2023 16:51:40 by DevTeam