Battlefield: Play4Free Arguments Injection

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

Abstract *This paper describes a remote code execution vulnerability in Battlefield Play4Free[3]. The vulnerability was first presented by the authors at Black Hat Europe 2013[4], as part of a talk covering several interesting aspects related to games security.*

1 Software Description

From Wikipedia[5]: "Battlefield Play4Free (sometimes abbreviated BF: P4F) is a first-person shooter video game developed by EA Digital Illusions CE and Easy Studios and published by EA. Based on the Battlefield series, the game features a modern warfare battlefield setting. Play4Free is built on a modified version of Battlefield 2's game engine with improvements such as high resolution artwork and post-processing effects. The game is also less demanding on computer specifications, similar to Battlefield Heroes.

Figure 1: Battlefield Play4Free

As the game's title suggests, the game is available to players for free online, under Electronic Arts' "Play4Free" model. Play4Free uses a similar micro-transaction store system similar to that in Battlefield Heroes."

[1]http://twitter.com/luigi_auriemma
[2]http://twitter.com/dntbug
[3]http://battlefield.play4free.com
[4]http://www.blackhat.com/eu-13/briefings.html#Ferrante
[5]http://en.wikipedia.org/wiki/Battlefield_Play4Free

2 Vulnerability Description

Battlefield Play4Free is based on the Frostbite[6] game engine and it shares the same architecture with Battlefield Heroes[7]. The game architecture will be described first, in order to get a better understanding of the issues affecting the game.

2.1 Game Architecture

An overview of the game architecture is shown in the following image:

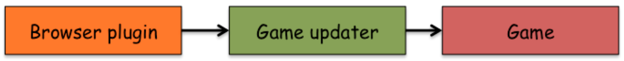

Figure 2: Battlefield Play4Free architecture

The interaction among these three components is the following:

- The *Browser Plugin* exports the following method to the *browsers*, so any website can call these functions:
  - *Start(bstrCmdLine, bstrDotnetfxUrl);*
- When *Start* is called the *Game Plugin* executes the following code:
  - *CreateProcessW("B*Updater.exe %bstrCmdLine% -host %website%");*
- *%website%*, the website calling the *Start* function, is checked against a whitelist[8] by the *Game Updater* component.
  - Only websites in the whitelist can use the *Game* component
- The *Game Updater* checks the game version and executes the *Game* by using a command line, which may contain the following arguments:
  - *dc, lang, sessionId, soldierName*

Please note that the arguments are passed from the *Game Plugin* as is to the *Game* component. Most interestingly, the Battlefield Play4Free *soldierName* argument can be abused.

[6]`http://dice.se/frostbite/`
[7]`http://www.battlefieldheroes.com`
[8]See *Appendix A* for details

2.2 CreateProcessW and Host Whitelist Bypass

The *Game Updater* component performs a whitelist check for security reasons, to prevent games running from malicious (non withelisted) hosts. The following image gives an idea of how the *command line* is handled by the game:

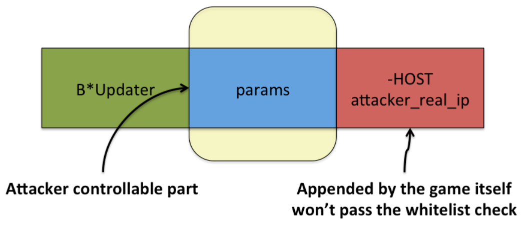
B*Updater
params
-HOST
attacker_real_ip
Attacker controllable part
Appended by the game itself
won't pass the whitelist check

Figure 3: The whitelist check in action

The vulnerability is exploitable only on *Windows XP an 2003*, due to the way the *CreateProcessW* API works for these versions. The *CreateProcess* function is defined as follows:

```
BOOL WINAPI CreateProcess(
  _In_opt_      LPCTSTR lpApplicationName,
  _Inout_opt_   LPTSTR lpCommandLine,
  _In_opt_      LPSECURITY_ATTRIBUTES lpProcessAttributes,
  _In_opt_      LPSECURITY_ATTRIBUTES lpThreadAttributes,
  _In_          BOOL bInheritHandles,
  _In_          DWORD dwCreationFlags,
  _In_opt_      LPVOID lpEnvironment,
  _In_opt_      LPCTSTR lpCurrentDirectory,
  _In_          LPSTARTUPINFO lpStartupInfo,
  _Out_         LPPROCESS_INFORMATION lpProcessInformation
);
```

Consider function parameter *LPTSTR lpCommandLine*. The *CreateProcessW* (not *CreateProcessA*) function handles this parameter differently depending on the version of Windows in use on the system. Specifically, prior to Windows Vista, if *lpCommandLine* is longer than 32kb, *CreateProcessW* truncates *lpCommandLine* to 32kb, and executes the command. For Windows Vista and later versions, it terminates. The following image shows a tricky way to bypass the whiltelist check:

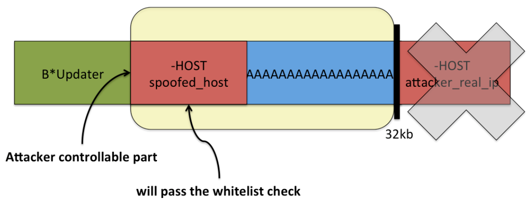

Figure 4: Bypass the whitelist check by padding out the real host

2.3 Targeting Windows XP

Battlefield Play4Free has been available since 2011, and it requires *DirectX 9* in order to run. The game uses a low amount of system resources and it is used on Windows XP systems too. Issues affecting Windows XP are still valuable. As of March 2013, Windows XP has about 40% of the market share[9].

2.4 The Root Cause

The game allows us to exploit the *soldierName* argument as the *Game Updater* supports the char " *(double quote)*, while the *Game* component doesn't. Because of that it is possible for an attacker to inject arbitrary sequences of "arguments". The following image shows the problem by detailing how each components handles a given command line string (*bstrCmdLine*):

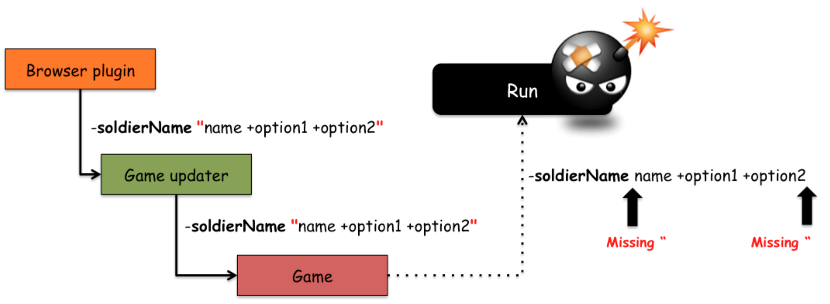
Browser plugin
-soldierName "name +option1 +option2"
Game updater
-soldierName "name +option1 +option2"
Game
Run
-soldierName name +option1 +option2
Missing "
Missing "

Figure 5: Components handling bad arguments

It's interesting to note that *Battlefield Heroes* is not affected by this issue, since its *Game Updater* component doesn't support the double quotes for *soldierName* option. Please keep in mind that this vulnerability is limited to Windows XP, due to the way the *Browser Plugin* spawns the process using *CreateProcessW*.

2.5 Possible Attack Plan

To exploit the vulnerability, the *+modPath* option was used to specify a directory containing game mod data (sounds, map, etc..). Mod data is usually composed of *ZIP files* and *CON files* to configure the Frostbite game engine. Interestingly, the *+modPath* option can refer to arbitrary paths, which may include SMB/WebDAV locations. In general the *+modPath* option is used to load files, such as *RankSettings.con*. Consider the idea to craft a *RankSettings.con* file, to invoke the following commands on the game engine:

- *sound.addSound*
- *ObjectTemplate.soundFilename*
- *sound.listSoundsToFile*

To make the attack more stealthy an attacker can add *Game.crash* to the sequence of commands above. This will cause the program to terminate just after executing all the commands.

[9]http://en.wikipedia.org/wiki/Microsoft_Windows#Usage_share

2.6 +modPath Exploitation: Step 1

This section describes the approach used to demonstrate the vulnerability. The following sequence of commands were used:

- *sound.addSound*
  - specifies the name of the sound resource
- *ObjectTemplate.soundFilename*
  - used for commands, note:
    - the double quote char (") cannot be used and no escape char is available
    - the comma char (,) is used to delimit file names, for example, "hello, world" is split in 2 lines with the comma removed
    - each line will be converted to lowercase
    - each line must be unique, and duplicates are discarded
    - each line generated is followed by the string: *;0.000000;0.000000;0;0.000000;0;0;UNKNOWN*
    - all backslashes will be converted to forward slashes
    - each line can be a maximum of 1024 bytes due to a *sprintf_s*
    - the usage of % is limited due to a format string bug
- *sound.listSoundsToFile*
  - stores the commands in a file

2.7 +modPath Exploitation: Step 2

It's important to keep in mind the following limitations:

- *UNC* paths are excluded because backslashes are converted to forward slashes
- *ftp.exe*
  - doesn't support passive mode, and the non-passive mode will trigger the firewall
- *vbs*
  - *soundFilename* limitations make its usage very difficult, although a good option for a *download&execute* approach
- *telnet -f*
  - limited to text data
- *rundll*
  - comma not allowed
- *webdav*

    - available from XP to Windows 8 except for 2003 (disabled), as a manual or automatic service[10].

- *tftp.exe*
    - available by default on Windows XP, it works on udp, and it's not limited by the firewall

2.8 +modPath Exploitation: Step 3

To demonstrate the vulnerability, *tftp.exe* was used. This is probably the best choice since only Windows XP and 2003 are affected by the vulnerability. The following describes how to bypass the limitations:

- use && in the batch file to avoid handling the string "*;0.000000;0.000000;0;0.000000;0;0;UNKNOWN*":
- use %% instead of % to avoid the format string vulnerability:
- the *%TEMP%* folder was chosen because Windows XP uses *8.3* expansion, alternatively fixed paths could be used, such as: *c:\windows\temp*, etc.

The proof of concept video provides several additional details on how an attacker may exploit this issue.

3 Proof Of Concept

A proof of concept video[11] showing how to spawn a reverse shell is available on our Vimeo channel[12]. The video shows all steps described in this paper.

4 Appendix A: Whitelist

The following are the whitelisted domains, for the *host check* performed by the game:

- 159.153.184.131
- dice-mdraper1.dice.ad.ea.com
- dice-jtarnstro1.dice.ad.ea.com
- easy-bfh-dev-trunk
- easy-bfh-test-trunk
- easy-bfh-test-release
- preprod.battlefield.play4free.com
- battlefield.play4free.com
- pte.battlefield.play4free.com
- preprod.battlefield.play4free.com

[10]`http://batcmd.com/windows/xp/services/webclient/`
[11]`http://vimeo.com/61364094`
[12]`http://vimeo.com/revuln`

5 FAQ

The following FAQ provides additional information about the issue described in this paper.

- Is this issue exploitable only on Windows XP systems?
  - Yes, XP and 2003.
- What are the affected versions of the components?
  - *Plugin* component version *1.0.80.2*, and *Game* component *1.52.245751.0*
- I can't reproduce the issue, why?
  - Because EA, after our paper, fixed on *25 March 2013* the issue with a new release of the *Plugin* component, version *1.0.95.0*

6 Revision History

- 25 March 2013: Version 1.1 released, FAQ section added.
- 22 March 2013: Version 1.0 released.