NIEYLANA, SSLEVIN

Unwrapping WildTangent Games

Version 1.1
January 2009

Table of Contents

1. **Introduction**
2. **Target 1: Aces of the Galaxy**
3. **Target 2: Diego's Safari Adventure (FWS Overlay)**
4. **Target 3: Dora's Carnival 2 (10JP Overlay)**
5. **Target 4: Mahjong Quest 3 (Expired)**

Editor: Nieylana

Foreword

The WildTangent Wrapper is a Software Protection system developed by WildTangent Inc, which is based out of Redmond WA. The wrapper is used as a marketing technique for developers to release their software as a trial, and provide you the option of either buying the game outright or paying as you play.

Each game comes with 2 free trials, however these trials are only deducted if you have played the game for longer than 3 minutes. After you have used up both trials you must either buy the game or buy wildcoins to continue playing.

The WildTangent Wrapper also offers developers the ability to give a pre-determined about of trial play time instead of trial sessions, but the majority if not all use the trial sessions option.

All WildTangent games come with 2 executable files. The executable ending in '-WT.exe' is called the launcher, and the other executable is called the shell.

By wrapping an executable with WildTangent, the code from the application is stripped away and replaced with 00s. This stripped code is then added to the launcher executable as an encrypted overlay. If the launcher determines the user is allowed to play the game it will then rebuild the executable in memory and run it.

The methods used to store wildcoins and trial sessions use encryption and randomization very heavily so we will not be pursuing this option in the paper.

The goal of this paper is to show a couple things the developers of the wrapper overlooked, and exploit these, ultimately resulting in obtaining the original executable prior to being wrapped.

The techniques described in this paper are the result of work between myself and SSlEvIN of ARTeam, we started this project more or less just for fun, but the unique challenges provided by this wrapper quickly made it more and more of a challenge. Initially I started to pursue the possibility of resetting the trial sessions but due to the heavy use of cryptography I abandoned that route, and went for other ways. After about a week, I managed a way to defeat it, this technique is described in Target 4.

Enjoy,
Nieylana, SSLEVIN

Disclaimers

All code included with this tutorial is free to use and modify; we only ask that you mention where you found it. This tutorial is also free to distribute in its current unaltered form, with all the included supplements.

All the commercial programs used within this tutorial have been used only for the purpose of demonstrating the theories and methods described. No distribution of patched applications has been done under any media or host. The applications used were most of the times already been patched by other fellows, and cracked versions were available since a lot of time. ARTeam or the authors of the papers shouldn't be considered responsible for damages to the companies holding rights on those programs. The scope of this document as well as any other ARTeam tutorial is of sharing knowledge and teaching how to patch applications, how to bypass protections and generally speaking how to improve the RCE art. We are not releasing any cracked application.

Verification

ARTeam.esfv can be opened in the ARTeamESFVChecker to verify all files have been released by ARTeam and are unaltered. The ARTeamESFVChecker can be obtained in the release section of the ARTeam site: http://releases.accessroot.com

Table of Contents

Foreword 1
Disclaimers 2
Verification 2
Table of Contents 2
1. Unwrapping WildTangent Games, Nieylana/SSIEvIN 3
1.1. Abstract 3
1.2. Targets 3
1.3. Aces of the Galaxy 3
1.3.1 Preparation 3
1.3.2 Checking out the target 3
1.4. Diego's Safari Adventure (FWS/CWS Overlay) 6
1.4.1 Preparation 6
1.4.2 Checking Out The Target 6
1.5. Dora's Carnival 2 (10JP Overlay) 8
1.5.1 Preparation 8
1.5.2 Checking Out The Target 8
1.6. Mahjong Quest 3 (Expired) 9
1.6.1 Preparation 9
1.6.2 Checking Out The Target 9
2. Greetings 12
3. Document History 13
3.1. Conclusions 13

1. Unwrapping WildTangent Games, Nieylana/SSIEvIN

1.1. Abstract

This tutorial will cover the basics on unwrapping 4 targets wrapped with the WildTangent Wrapper, each target looked at in this tutorial must be handled a different way. We will cover a standard game, 2 games with Flash Overlays (FWS/CWS, and 10JP), and an expired game.

Tools used in this tutorial include:

1. OllyDbg (Latest Version)
2. LordPE (Latest Version)
3. HexEditor (I Use 010 Editor)

1.2. Targets

The games are available for download at:

- Aces of the Galaxy: http://hp.wildgames.com
- Diego's Safari Adventure: http://hp.wildgames.com
- Dora's Carnival 2: http://dell.wildgames.com
- Mahjong Quest 3: http://hp.wildgames.com

1.3. Aces of the Galaxy

1.3.1 Preparation

If you scan the –WT.exe executable in the target's directory with PEiD you will see that the wrapper is written if Microsoft Visual C++.

1.3.2 Checking out the target

Open up the installation directory : C:\Program Files\HP Games\Aces of the Galaxy. You will notice there is AcesOfTheGalaxy.exe and AcesOfTheGalaxy-WT.exe

If you try to run AcesOfTheGalaxy.exe, it crashes. Let's look at it in Olly to figure out why.

This is our OEP:

```
007B072E  $  0000          ADD BYTE PTR DS:[EAX],AL
007B0730  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0732  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0734  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0736  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0738  .  0000          ADD BYTE PTR DS:[EAX],AL
007B073A  .  0000          ADD BYTE PTR DS:[EAX],AL
007B073C  .  0000          ADD BYTE PTR DS:[EAX],AL
007B073E  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0740  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0742  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0744  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0746  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0748  .  0000          ADD BYTE PTR DS:[EAX],AL
007B074A  .  0000          ADD BYTE PTR DS:[EAX],AL
007B074C  .  0000          ADD BYTE PTR DS:[EAX],AL
007B074E  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0750  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0752  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0754  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0756  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0758  .  0000          ADD BYTE PTR DS:[EAX],AL
007B075A  .  0000          ADD BYTE PTR DS:[EAX],AL
007B075C  .  0000          ADD BYTE PTR DS:[EAX],AL
007B075E  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0760  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0762  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0764  .  0000          ADD BYTE PTR DS:[EAX],AL
007B0766  .  0000          ADD BYTE PTR DS:[EAX],AL
```

It should be clear as to why it crashes.... There's NO CODE?!.

Now if you run the AcesOfTheGalaxy-WT.exe (now called the Launcher), the application runs and you have to use tokens to play the games.

The launcher uses CreateProcessA to start AcesOfTheGalaxy.exe (The Shell EXE). Remember Shell EXE contains no code, it's has place holders where the code used to be.

By starting the process with CREATE_SUSPEND the Launcher is able to write the code into the place holders before continuing execution.

So let's open the Launcher EXE in OllyDbg, Press Ctrl+G and type CreateProcessA. Set a SWBP on this API so we know when the launcher is trying to start the shell EXE.

```
7C802367  8BFF          MOV EDI,EDI
7C802369  55            PUSH EBP
7C80236A  8BEC          MOV EBP,ESP
7C80236C  6A 00         PUSH 0
7C80236E  FF75 2C       PUSH DWORD PTR SS:[EBP+2C]
7C802371  FF75 28       PUSH DWORD PTR SS:[EBP+28]
7C802374  FF75 24       PUSH DWORD PTR SS:[EBP+24]
7C802377  FF75 20       PUSH DWORD PTR SS:[EBP+20]
7C80237A  FF75 1C       PUSH DWORD PTR SS:[EBP+1C]
7C80237D  FF75 18       PUSH DWORD PTR SS:[EBP+18]
7C802380  FF75 14       PUSH DWORD PTR SS:[EBP+14]
7C802383  FF75 10       PUSH DWORD PTR SS:[EBP+10]
7C802386  FF75 0C       PUSH DWORD PTR SS:[EBP+C]
7C802389  FF75 08       PUSH DWORD PTR SS:[EBP+8]
7C80238C  6A 00         PUSH 0
7C80238E  E8 43BA0100   CALL CreateProcessInternalA
7C802393  5D            POP EBP
7C802394  C2 2800       RET 28
```

Press F9 to run the Application.

After the WildTangent Launcher window shows up click on Play. OllyDbg should break on the CreateProcessA API. Step with F8 until the RET 28, and then step F8 one more time to return to user code.

You may have to analyze the code (Ctrl+A). You should see this:

```
0048DC8C  .  8985 24FFFFFF     MOV DWORD PTR SS:[EBP-DC],EAX
0048DC92  .  C645 FC 03        MOV BYTE PTR SS:[EBP-4],3
0048DC96  .  8D8D 08FFFFFF     LEA ECX,DWORD PTR SS:[EBP-F8]
0048DC9C  .  E8 2F3BF7FF       CALL 004017D0
0048DCA1  .  83BD 24FFFFFF 00  CMP DWORD PTR SS:[EBP-DC],0
0048DCA8  .v 0F84 96000000     JE 0048DD44
0048DCAE  .  A1 181F4B00       MOV EAX,DWORD PTR DS:[4B1F18]
0048DCB3  .  83E0 01           AND EAX,1
0048DCB6  .  F7D8              NEG EAX
0048DCB8  .  1BC0              SBB EAX,EAX
0048DCBA  .  F7D8              NEG EAX
0048DCBC  .  48                DEC EAX
0048DCBD  .  8985 48FFFFFF     MOV DWORD PTR SS:[EBP-B8],EAX
0048DCC3  .  8B8D 48FFFFFF     MOV ECX,DWORD PTR SS:[EBP-B8]
0048DCC9  .  51                PUSH ECX                                 Arg2
0048DCCA  .  8B55 08           MOV EDX,DWORD PTR SS:[EBP+8]
0048DCCD  .  8B02              MOV EAX,DWORD PTR DS:[EDX]
0048DCCF  .  50                PUSH EAX                                 Arg1
0048DCD0  .  8B8D 00FFFFFF     MOV ECX,DWORD PTR SS:[EBP-100]
0048DCD6  .  E8 C5000000       CALL 0048DDA0                            AcesOfTh.0048DDA0
0048DCDB  .  0FB6C8            MOVZX ECX,AL
0048DCDE  .  85C9              TEST ECX,ECX
0048DCE0  .v 74 2B             JE SHORT 0048DD0D
0048DCE2  .  68 60264B00       PUSH 004B2660
0048DCE7  .  E8 E1D4FFFF       CALL 0048B1CD
0048DCEC  .  83C4 04           ADD ESP,4
0048DCEF  .  8985 44FFFFFF     MOV DWORD PTR SS:[EBP-BC],EAX
0048DCF5  .  8B55 08           MOV EDX,DWORD PTR SS:[EBP+8]
0048DCF8  .  8B42 04           MOV EAX,DWORD PTR DS:[EDX+4]
0048DCFB  .  50                PUSH EAX                                 hThread
0048DCFC  .  FF15 A8F24800     CALL DWORD PTR DS:[<&KERNEL32.ResumeThread>]  ResumeThread
```

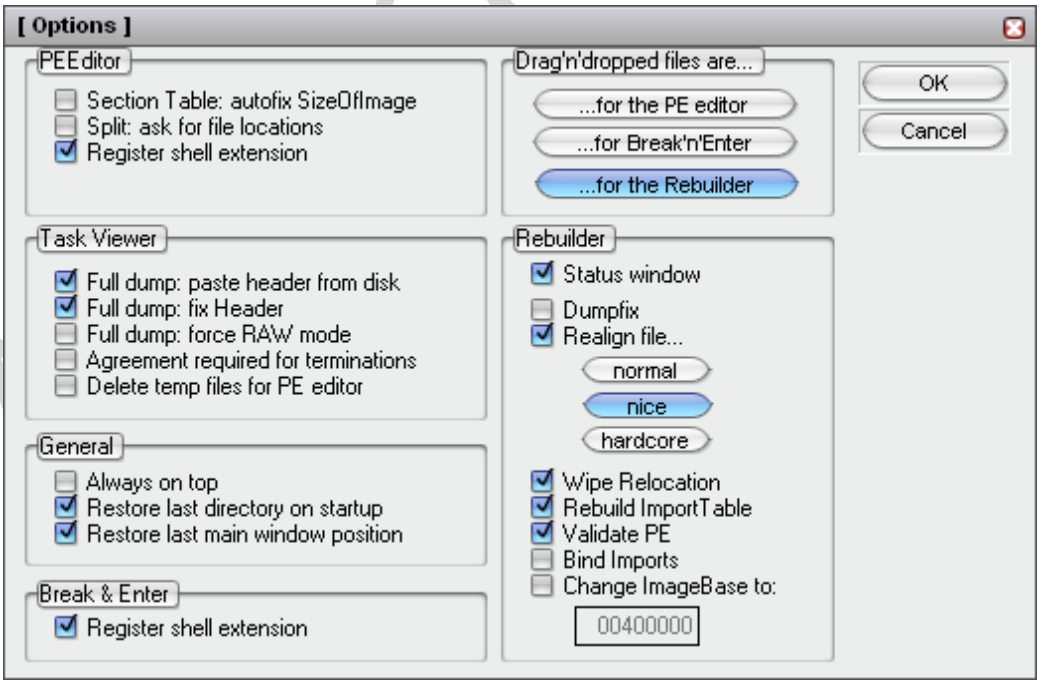
Notice the call to ResumeThread, between where we are right now, and the ResumeThread, the launcher must write all the code into the shell EXEs memory space. This occurs at line 0048DCD6.

Step all the way until the call to ResumeThread. DO NOT step over the call.

At this point we have the shell executable loaded into memory and all the code written to the process's memory space. So effectively we have the original executable in memory.

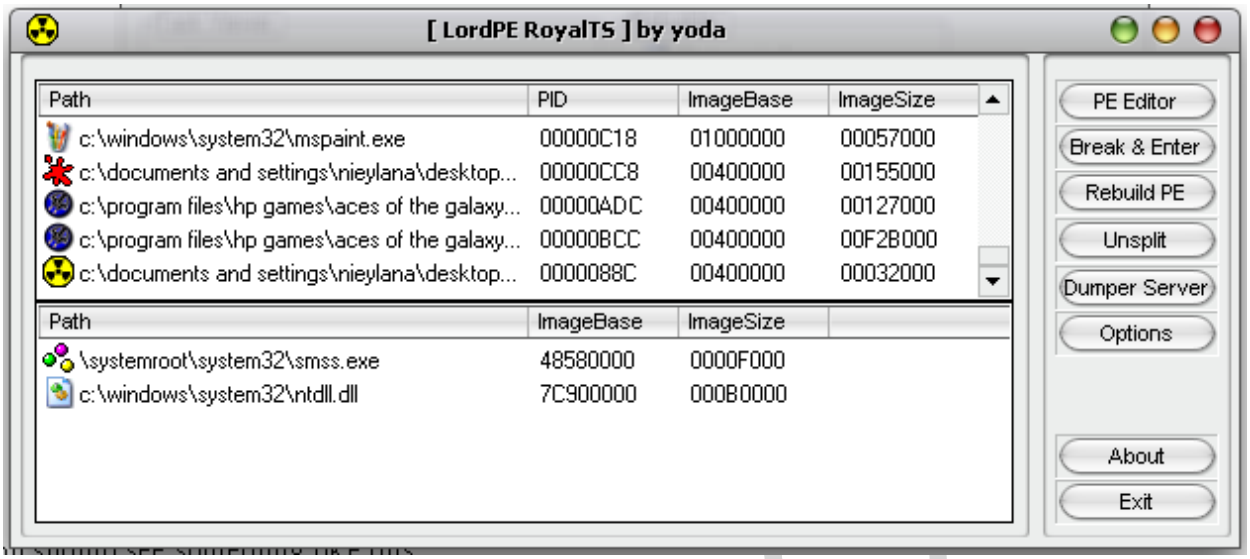
Minimize OllyDbg, we're done with it. DO NOT exit though

Open up LordPE and make sure your settings are as follows:



After your options match mine, click ok. Then scroll down to the bottom of the list of processes,

You should see something like this:



Notice the 2 Aces of the Galaxy processes, the top one is the Launcher, the bottom is the Shell.

Right click on the bottom Aces Of the Galaxy Process (AcesOfTheGalaxy.exe) and select Dump Full.

Save the dumped file to the installation directory of the program. You have successfully dumped the game. The dumped.exe will run as the full version. This is the generic method of unwrapping the games. Prolems only occure when the game is Flash based because you must re-append the Flash Overlay to the dumped executable. The next target will show how to re-append the flash overlay to the dumped executable.

1.4. Diego's Safari Adventure (FWS/CWS Overlay)

1.4.1 Preparation

If you scan this target's launcher with PEiD you will again see that it was written in Microsoft Visual C++, but you will also notice that it has an overlay. We will find out later that the overlay it has is in fact an FWS Flash Overlay.

1.4.2 Checking Out The Target

First you need to create a dumped file like explained above, name it dumped.exe.

Re-appending the flash overlay is going to rely heavily on a hex editor. I recommend using 010 Editor (cracks available for it) because it has a template functionality which will help you easily locate the Overlay and it's size and other things. This tutorial is going to assume you have 010 Editor.