THE CONCEPT OF GAME HACKING BYPASSING GAME PROTECTION (HACKSHIELD)

大可(Dark)

關於我

- 熟悉的語言:C/C++ , PASCAL , ASM
- 專長:windows 程式設計&逆向工程
- 講課經驗:在ZCamp2008講過課
- 興趣:程式設計&資訊安全&美食&聊天&動漫&睡覺&看電影&聽音樂&彈鋼琴&打電玩

Hackshield 簡介

- **Hackshield**是一款防止外掛程式的入侵的軟件，執行遊戲時，**Hackshield**會偵防外掛使用者的電腦， 並封鎖不正常的程式碼， 有效防止按鍵精靈、加速器等的外掛的運行。
- 轉自
  - http://eco.gamecyber.com.tw/tw/hanckshield01.html

稍微介紹一下CE與UCE

- CE (Cheat Engine)
  - 讀寫記憶體
  - 按照使用者指定的方式去比對記憶體的資料
  - 有開放原始碼
- UCE(Undetected Cheat Engine)
  - 防外掛軟體會封鎖CE
  - 有人改CE原始碼,改成防外掛檢測不到

如果你看懂今天的主題

- 不需要再去找Bypass HS(hackshield)版的UCE
- 也不用自己修改CE原始碼
- 可以寫一個程式，使”任何工具”繞過HS

如何實現這個工具呢?

- 其實這部分很容易, 前提是要知道如何繞過HS
- 可以把繞過HS的方法,寫成DLL
- 然後把DLL inject到指定的外掛工具中

Hackshield行為分析

可以歸納為"攻" 與 "守" 兩個動作

攻(主動式的防禦)

- 目的:偵測對遊戲不利的程式
- inline hook SSDT- NtDeviceIoControlFile:
  - 攔截分析: 由於CE(Cheat Engine)從driver呼叫 CE內部的OpenProcess方式，所以Usermode 必須呼叫 DeviceIoControl 跟 driver交換訊息。
- 不定期取得process的資訊。
- enum window尋找可疑的視窗。

守(被動式的防禦)

- 目的:讓遊戲記憶體不被外部程式Access,並防止HS遭修改
- Inline hook SSDT- NtOpenProcess
  - 防止被外部程式獲得Process Handle，
  - 然而工作管理員卻是白名單(可以給我們利用)
- Hook shadow SSDT-NtUserSendInput
  - 防止模擬鍵盤滑鼠輸入
- 此外會不定期對遊戲的code segment做CRC check
  - 防止遊戲程式碼被修改!
- Inline hook SSDT– NtReadVirtualMemory
  - 防止遊戲記憶體被讀取
- Inline hook SSDT– NtWriteVirtualMemory
  - 防止遊戲記憶體被寫入

外掛任務其實很簡單

- 外掛也可以分為攻與守兩個方式
  - 攻:繞過,防護對遊戲記憶體做讀寫
  - 守:防止工具被HS檢驗到
- 上面兩點都達到大概就圓滿落幕了

攻擊手法 1: 就是不讓你載入驅動

- Hook NtLoadDriver
  - 這樣就能使HS的驅動不正常運作

但是

- 防外掛不只包含那個驅動, 有其他模組會檢查HS驅動執行是否正常
  - 不正常就關閉遊戲

Demo時間

攻擊手法 2:把她整個拿掉

- 追蹤 HS載入的地方
- 修改程式，完全拔掉 HS
- 可以完全拔掉...

但是

- 遊戲會發"確認HS運作正常"的封包

Demo時間

關於攻擊手法 1 及 2

- 如果破壞 hackshield 的完整性
  - 要進行繁瑣的修復動作
    - 分析伺服器送出的確認封包
    - 模仿加密過程送出
    - 才能讓遊戲正常執行
- So , 選擇與hackshield共存

攻擊手法 3: 跟遊戲打"鉤鉤"

- 透過hook , 來載入我們的dll
- 防止 HS對Executable Module list做驗證
  - dll injection成功之後,把dll module給隱藏
  - 就能直接存取遊戲的記憶體空間
- 別用ReadProcessMemory 或 WriteProcessMemory讀寫記憶體
  - 遊戲有對這兩個API做Inline SSDT Hook

如何隱藏DLL Module?

- 找到PEB(Process Environment Block)
  - 在fs[0x30]可以找到peb ,像這樣" mov eax , fs: [0x30]"
- 到PEB->Ldr, 去enum下面三個module的List_Entry結構
  - InLoadOrderModuleList
  - InMemoryOrderModuleList
  - InInitializationOrderModuleList
- 比對module,把要隱藏的module連結弄斷
  - Current->Blink->Flink=Current->Flink
  - Current->Flink->Blink=Current->Blink
- 指定的dll module就會被孤立

Demo時間

攻擊手法 4: 工作管理員大大

- 開工作管理員可以結束掉遊戲的process
  - 可見工作管理員能對遊戲做OpenProcess的動作
  - 工作管理員(taskmgr.exe)在HS的白名單裡面

- 從 PEB 到 _RTL_USER_PROCESS_PARAMETERS 結構，改掉ImagePathName來偽裝

Demo時間

上述方法原始碼如下

```
void Fake_Fake_FakeXD (WCHAR* wszImagePathName)
{
	_asm
	{
		mov eax,fs:[0x30]	//eax points to PEB
		mov eax,[eax+0x010]	//eax points to _RTL_USER_PROCESS_PARAMETERS
		add eax,0x38		//eax points to ImagePathName(UNICODE_STRING)
		add eax,0x4		//UNICODE_STRING.Buffer
		mov ebx,wszImagePathName
		mov [eax],ebx
	}
}
```

攻擊手法 5: 別忘了你老爸是誰

- HS針對 NtOpenProcess 弄了一個inline hook，偵測這個hook是否有效
  - 無效的話就BSOD。
- 先不要動NtOpenProcess,他們忘記了NtOpenProcess上面的老大
  - ObOpenObjectByPointer
- 利用這來獲得Process Handle

Demo時間

攻擊手法 6:別拿Process Handle了

利用 KeAttachProcess 這個Native API

弄個Read&WriteProcessMemoryByPid

如此就不用煩惱取得Process Handle的種種問題!

Hook Shadow SSDT

- hook shadow SSDT，把獲得視窗Handle的函數hook掉
- 例如
  - NtUserQueryWindow
  - NtUserGetForegroundWindow
  - NtUserWindowFromPoint
  - NtUserFindWindowEx
  - NtUserBuildHwndList

Demo時間

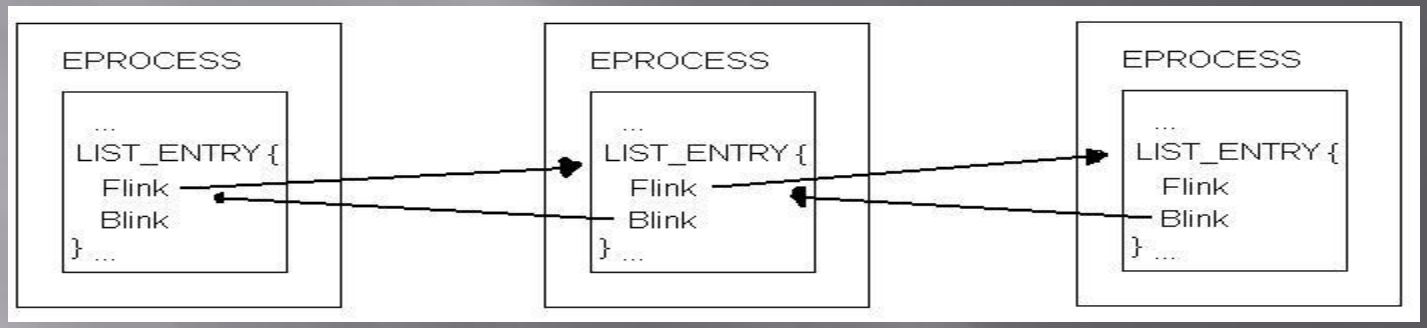
Hook ObOpenObjectByPointer

防止外掛的Process Handle被HS拿去用

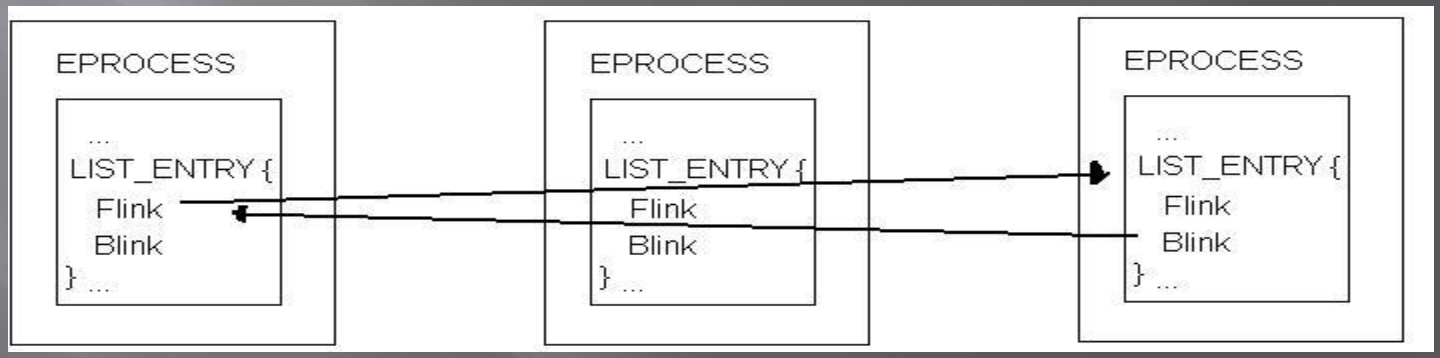
Demo時間

Hidden Process With DKOM

防止HS從Process List獲得我們外掛的一些資訊



Afert



Demo時間

結論

要保護遊戲是件不容易的事情

幕後花絮

參考

- DKOM(Direct Kernel Object Manipulation)
- http://www.blackhat.com/presentations/win-usa-04/bh-win-04-butler.pdf
- HOWTO: Implement your own NtOpenProcess in kernel-mode
- http://wj32.wordpress.com/2009/02/19/howto-implement-your-own-ntopenprocess-in-kernel-mode/
- Undocument-PEB Structure
- http://undocumented.ntinternals.net/UserMode/Undocumented%20Functions/NT%20Objects/Process/PEB.html
- 简单说说SSDT
- http://icylife.net/yunshu/show.php?id=435
- NEXON采用的新反作弊软件--Hack Shield Pro 介绍
- http://qbar.games.qq.com/popkart/165052.htm?owner=66191052
- Cheat Engine
- http://www.cheatengine.org/

聯絡方式

Email: cl4rk.z3r0@gmail.com

Blog: http://cl4rk.pixnet.net/blog/

ANY Question?