WG_principle

原创

sdt0263 于 2010-05-13 13:11:00 发布 1172 收藏 6
文章标签： 游戏 算法 网游 破解 脚本 加密

首先，我想说明几点：

第一，这篇文章并不是具体教你如何写外挂，只是带你大致浏览一下网游外挂的制作流程，并就其中的一些关键技术点加以简单说明。大家可以用看故事书的心情来阅读此文，了解一下网游外挂制作过程中的一些原理。

第二，网游数据的破解爱麻烦，通常一个网游外挂制作团队内都有一名破解高手坐镇。所以碰到破解方面的问题我就只能一笔带过了。

一、游戏封包的加密与解密算法的破解

破解封包的加密与解密算法是制作外挂的第一步，是外挂制作中最具技术含量的步骤，同样也是一个十分令人头痛的环节。如果加密与解密算法被成功地破解，那么外挂制作也就完成了一半。破解封包的加密与解密算法的行为同样属于黑客们其中之一的行为，因此我们可以在黑客网站里找到相应的资料，另外网络上也有专门的破解网站为大家提供信息。

在破解封包的加密与解密算法之前我们首先需要知道一些情况。我们知道随着机器性能的提高与网络带宽的提升，新的游戏运行商对游戏封包的加密与解密算法的设计变得越来复杂。那种原先通过分析封包数据就可以得出加密与解密算法的时代已经变成了过去。现在如果再要破解一个游戏的封包的加密与解密算法，那么其必须通过分析程序源代码才能清楚。

1.1 封包的概念 本文所讲的封包是指由sockets协议进行发送与接收的数据包。广义的封包是指计算机之间互相进行通信的数据包，其可以因通信协议的不同而在内容上有所不同。

1.2 破解原理 目前破解封包加密与解密算法的方法主要是通过动态调试技术来实现的。其原理是首先通过动态调试跟踪并取出加密与解密算法的代码段，然后再通过分析这些代码最终得出结论。

那为什么我们可以跟踪并取得这些代码呢？首先我们知道无论游戏程序如何设计，其加密与解密算法的代码永远存在于程序中；其次我们知道在程序流的执行过程中，加密与解密算法的代码段一定会被执行。

1.3 破解需要具备的知识 要能顺利进行破解则必须具备一定的知识，一是熟练掌握汇编原理与汇编语言，二是要熟悉加壳与脱壳原理（虽然很多游戏不需要），三是要熟悉代码结构的知识，四是要熟悉动态调试技术与调试工具的使用，五是具有高级语言知识与较高的编程修为。

1.4 破解的技术与方法 动态调试工具我们可以采用OllyDbg工具或其他工具，不熟悉OllyDbg工具的可以查看它的中文帮助。动态调试主要是跟踪代码的执行，而我们查找加密与解密代码段就是一个跟踪的过程。一般跟踪的起点可以是windows消息、socket中的send与recv等函数。有时程序有可能将发送与接收过程由一个专门的线程进行处理，那么这种情况下我们需要找出该处理线程。至于具体如何进行跟踪本文不再进行详述，具体内容可以到相关网站上查看，比如看雪学院、笨冬瓜等网站。这里主要讲一下加密与解密算法代码的特征。

1.5 加密与解密算法代码特征 虽然在未进行分析之前我们很难判断一段代码是否是加密或解密算法，但我们还是可以根据一些特征进行较大概率的猜测。一情况下加密与解密都要进行一系列的异或、移位、加减、乘除和重复运算过程，因些一段代码中若具有上述特征，我们可以进行较为肯定的断定。

1.6 代码反推导 代码反推导是进行破解算法的主要方法，代码反推导的水平主要与一个人自身的编程修为相关，但在这里仍有一些基本的方法。代码反推导可以有一定的程式，首先可以将汇编码写成三元表达码，其次将代码中的转移指令转换为条件语句或循环语句，再次将代码中的变量进行迭代，最后进行变量形式转换与语句形式转换。通过以上的步骤，一般我们可以将汇编语言转换为高级语言，而当我们推导出高级语言后，就能进行较为实义的分析。

①.选择一款目标游戏

制作网游外挂的第一步就是选定一款游戏。目标游戏不是乱选的，里面也有很多讲究。

第一点，选择自己熟悉的游戏类型。如果你之前已经做过网游外挂，那选择一款类似的游戏会给你节省很大的时间，如果是第一次制作的话，那也选一款自己熟悉的游戏类型。

第二点，尽量不要选择热门的游戏，因为热门的游戏往往意味着竞争对手的增多，而且反外挂系统也比较厉害。像梦幻诛仙驱动+反外挂...头疼.

第三点，不要小看玩家人数少的游戏，游戏规模小，一般技术也比较容易突破。如果想销售的话。一款游戏，只要你能形成吃独食的场面，再加上营销搞得好的话，其中的利润将超过你的想象。但要注意，最好不要碰上因为游戏规模过小导致游戏厂商把游戏关闭的衰事。

第四点，尽量选择尚在测试期内的游戏，这使得你有充足的时间制作外挂。

②.目标游戏初分析

1.确定制作网游外挂的类型.

目标网游选定好之后，你首先要做的第一件事就是确定你要制作的网游外挂类型。

网游外挂虽然统称为外挂，但细分的话可以分为以下二类：内挂和脱机外挂。

内挂就是在游戏内呼出的网游外挂，它依赖于网游客户端，所使用到的技术主要包括鼠标和键盘的模拟，内存特殊变量区域的搜索，或者是挂钩游戏的收包函数和模拟游戏的发包函数。

脱机外挂就是指不依赖于客户端，能独立模拟客户端和游戏服务器进行通讯的网游外挂。脱机外挂的实现方式只有一种，就是模拟网游客户端的收包和发包过程。

总体而言，内挂的整体制作难度比脱机外挂要简单一些，但脱机外挂制作要比内挂更有趣，而且用起来也更方便，不必启动庞大的客户端程序。

某些时候脱机比较容易上手，分游戏也不是都是这样。所以以下都是脱机教学为例：

二、游戏指令与数据结构的筛查

游戏指令和指令中所携带的数据结构的筛查是外挂制作的第二步，这一步并不是很难，但十分烦锁。游戏指令和其数据结构的筛查并无技巧可言，主要是通过多次重复比较数据而最终确定结果。

在筛查游戏指令和其数据结构之前我们需要对封包截获技术有所了解，同时能对当前流行的几款封包截获工具如FPE、WPE等有所熟练使用。然而由于某些游戏运行商会针对一些问题而制定相应对策，因此有些时候需要我们自己编写封包截获工具。自己编写封包截获工具的好处还在于可以具体针对某一款游戏而编制特定的工具，这之中最重要的是可以事先将加密数据解密成明码，为分析封包提供方便。

2.1 封包截获技术 根据具体的截获原理不同，封包截获技术可分为：一是Hook技术、一是socket重写技术。无论使用何种截获技术我们最终要跟踪的都是socket中的发送函数与接收函数，如send、recv等。

2.1.1 Hook原理 Hook原理是通过向应用程序中注入dll文件，并改写应用程序函数导入表中的DLL调用函数，Hook技术要求我们对可执行程序的文件即PE文件结构有所了解。

2.1.2 socket重写原理 socket重写原理是通过重写整个socket文件，用新写的socket文件代替原socket文件，并由新socket文件调用原socket文件中的函数，从而截获函数的调用过程。socket重写技术要求改写后的socket文件具有跟原socket文件相同的接口，否则程序调用将发生错误。

2.2 封包的分析 封包分析时最重要的就是尽量减少分析时的干扰，干扰越少越有利于我们能针对性地得到结果。因此封包分析时一般是将游戏角色尽量带到一个玩家或怪物比较少的地方，同时在分析出一个后尽量过滤一个。

2.3 分析结果的处理 封包分析完毕后，我们可以为每一个指令定义一个含义比较明确的代码，并为每条指令所携带的结构信息定义相应的数据结构，为指令中的状态码也定义相应的代码。为所有指令与数据结构进行相应的定义，可以使我们在后续的外挂代码书写过程中隐藏掉实现的细节。

1.网络截包工具（Microsoft Network Monitor）的使用简介

目标网游的初步分析最主要的工作是分析游戏初始阶段网游客户端和服务器之间的数据通讯。这一阶段主要是指从输入用户名和密码开始登录游戏到玩家人物出现在游戏场景中这个阶段。这是开始阶段最关键的一个步骤，如果你能够成功破解网游数据通讯部分的加密，并用Debug程序成功模拟整个登录过程，那你几乎就已经成功了一半了。如果无法破解加密的话，那就需要赶快重新选定一款游戏了。或者请高手或者朋友帮你分析=。=

关于初步分析，首先要确定网游客户端和服务器之间的大致通讯过程，最起码你要知道客户端连接的是哪一个服务器，连接的端口是多少，在登录的过程中发送和接受了几个包？而要了解这些东西，你就要使用到网络截包工具了。

初学者可以使用.Microsoft Network Monitor V3.1，方便简单。大家可以到下面的网址去下载该软件。

http://support.microsoft.com/kb/933741/zh-cn

下面，我简单介绍一下该软件的使用方法。

安装好程序之后，运行程序，点击【Start Page】页的【Create a new capture tab】按钮，创建一个新的数据捕获会话，点击工具栏上绿色的开始按钮，就可以开始捕获网络数据了。整个程序的界面如下图所示：

各个窗口的作用如下：

Network Conversations下面有二项：

My Traffic代表本机作为发送方或者接收方参与的网络数据包。选中该项后，Frame Summary中将仅仅列出与本机相关的网络数据包。

Other Traffic 则是网络上其他机器之间的网络数据包。因为正好在拦截期内经过本机，所以被顺道拦截了下来。

Capture Filter 是设定拦截数据时的过滤器。

Display Filter 是对拦截结果的过滤设定。

Select Networks 是设定需要拦截本机上的那一个网络。

Aliases用于设定友好名。

Frame Summary 中列出的是符合条件的所有网络数据包

Frame Details则是当前选中的网络数据包的详细结构

Hex Details 则是当前选中网络数据包的二进制格式

2.分析初始阶段C/S网络数据通讯

简单介绍了网络截包工具的使用之后，下面我们就开始初步分析了。

在这篇文章里，我以某款网络游戏作为假定目标。（具体是哪一款，大家就不要深究了。）

首先在【aliases】窗口中将本地客户端和游戏服务器分别命名为：MyComputer和GameServer。注意不要忘了点击【apply】按钮。

然后在【Display Filter】中输入如下语句，仅显示游戏客户端和服务器之间的数据包。数据包类型为TCP是因为网游通讯的协议是TCP协议。

下图中的数据包列表就是目标网游从输入用户名和密码登录游戏到人物出现在游戏中（然后立即退出。）这一阶段客户端和服务器之间的所有往来的数据包。

图中红线标识的三个包代表了一个连接的过程。注意它的TCP Flags的变化。

MyComputer è GameServer .S..... 客户端请求建立连接

MyComputer ç GameServer .S..A... 服务器同意建立连接

MyComputer è GameServer ....A... 连接建立

以上三个包称为建立TCP连接的三段式握手。当你调用Socket类的Connect方法时就会产生上面的三个TCP包。

图中蓝线标识的是连接断开的过程。

MyComputer è GameServer F...A... 客户端请求断开连接

MyComputer ç GameServer ....A... 服务器同意断开请求

MyComputer ç GameServer F...A... 服务器请求断开连接

MyComputer è GameServer ....A... 客户端同意断开请求

调用Socket类的Disconnect方法时就会产生上面的四个TCP包。

从上图中我们不难看出在验证用户名和密码的过程中，客户端和服务器之间总共连接了二次，所以在之后的外挂程序编写过程中，我们同样也要连接二次。

TCP Flag为...PA...表示该TCP包内带有数据，而....A...则是回应包，用于回应上一个包的发送方：我已经收到你上一个包了，它本身不带数据。所以一般一个...PA...包都有一个对应的....A...包（例如编号为266和269），但如果回应的时候，发现正好有数据要发送，则可以将回应包掺杂在发送包中发送过去（例如编号为273的回应包就掺杂在275这个包内）。

下面观察客户端和服务器之间的实际数据往来。

1. 客户端连接到服务器
2. MyComputer ç GameServer 服务器给客户端发送7字节的数据
3. MyComputer è GameServer 客户端给服务器发送90字节的数据
4. MyComputer ç GameServer 服务器给客户端发送65字节的数据
5. MyComputer ç GameServer 服务器给客户端发送48字节的数据
6. MyComputer è GameServer 客户端给服务器发送48字节的数据
7. MyComputer ç GameServer 服务器给客户端发送208字节的数据
8. 服务器断开连接
9. ……

以上就是第一次连接的大致过程。观察每个包内的具体传输数据是没有意义的，因为网游之间的通讯肯定是加密的，你每次拦截下来的数据都会不一样。通常游戏服务器给客户端发送的第一个包都是KEY包（例如上面的7字节的包），客户端在接收到KEY包之后执行相应的数据加密初始化。所以接下来的任务就是根据已掌握的数据通讯规律，对游戏客户端的加密算法进行破解了。

3.游戏加密算法破解

网络游戏所使用的网络通讯函数肯定也是微软操作系统所提供的标准API函数，所以通常在接受网络数据的API函数中下一个断点，当接收到第一个7字节包时，断点激活，然后逐渐跟进去，查看游戏客户端是如何处理该段数据的，然后我们在外挂中依样画葫芦，进行同样的处理。整个破解过程相当的枯燥无聊，因为面对的都是汇编代码。大致的说一下。

4.Debug版本制作

破解完成之后，就要制作一个能够登录游戏的Debug版本了，用于确认游戏加密算法的破解是否成功。

至于选择何种编程语言和工具制作外挂则没有限定，常用的如VC，Delphi，等都可以易语言和VB不太推荐局限性太多~但并不是说语言不好，具体的编程在此就不具体说明了，可以根据个人的喜好所选择，

下面谈谈网游中数据通讯的基本单位：指令包。

所谓指令包就是代表了一个最基本含义的数据包。比如游戏人物向左移动时，游戏客户端就会向服务器发送一个指令包（人物走路包），通知服务器更新游戏人物的坐标。当游戏人物周围出现一个新的怪物时，服务器会向客户端发送一个指令包（怪物出现包），通知客户端在画面上绘制出该怪物。所以，可以说指令包就是客户端和服务器之间所使用的通讯语言，而外挂的工作就是解析该种语言，然后模拟客户端和服务器端进行通讯。

各个游戏定义的指令包的格式都不一样，但一般一个指令包通常含有以下几个元素：

XX XX XX XX XX XX XX ...

XX XX 红色部分通常与该指令包的长度相关。他可能是指整个指令包的长度，也可能是指他余下部分指令的长度，这需要根据游戏的具体情况来确定。

之所以专门要用一定空间来说明指令包的长度，这是由SOCKET通讯的机制所决定的。

SOCKET连接建立好之后，通过SOCKET连接读取到的数据并不是以指令包为分割的。有可能一个TCP包中正好包含一条指令包，也有可能仅仅包含指令包的一部分（如下图所示）。所以这时候就要根据指令包长度将收到的网络数据截取成单个的指令包。

有一点需要指出的是：刚开始的几个数据包不一定遵循一定的规律，这时候就需要进行特殊处理（因为在开头，所以也比较好处理），而之后的数据包肯定是遵循指令包格式的，不然就乱套了。

XX XX 蓝色部分通常称为指令包标识，用于说明该指令包是属于哪一种类型。比如怪物攻击包，玩家的移动包……,游戏客户端根据收到的相应指令包采取不同的动作。事实上，在客户端程序的内部就是一个很大的Switch语句，用来处理不同的指令包,如下所示：

```
Switch(指令包类型)
{
 Case 移动包： 绘制相关人物的移动； break；
 Case 攻击包： 绘制A攻击B的画面； break；
 …
}
```

XX XX XX 部分是指令包详细的信息，该部分随着不同的指令包而有不同的格式。比如，如果是玩家移动包的话，他就会在此部分详细说明是哪个ID在移动，移动点是从哪儿到哪儿。

Debug成功制作完成的话，那我们就已经成功了一小半了。接下去的工作主要分为动态和静态二个方面。动态方面是根据已掌握的指令包格式，逐个分析游戏中的各个基本指令。静态方面则是从游戏客户端中解析出相应的资源。

发表于: 2010-2-9 1:43:00

- 档案
- |
- 主页
- |
- 短信
- |
- 树状
- |
- 收藏
- |
- 编辑
- |
- 删除
- |
- 引用
-

Re:外挂源代码&教学&工具下载[包含商业源代码]

③.网游基本指令分析

1.监控网游客户端的发送包和接受包

要分析网游指令，首先就要得到网游指令的数据样本。那么如何得到网游指令的数据样本呢？

首先想到的是使用网络截包工具，这种方式在网游发展早期的话还有一定的可行性，因为那时候网游大部分采用明文通讯，而现在的网游数据通讯肯定是加密的，所以使用网络截包工具截获到的数据毫无意义。既然无法使用通用的工具，那我们只能把目光移向游戏客户端了，因为在游戏客户端内处理的肯定是未加密的数据。

像之前的加密破解一样，跟踪游戏对网络数据的处理过程，分别找到游戏中对解密后的指令包的处理函数入口和游戏对要发送指令包进行加密的函数入口这二个地方，然后修改这二处地方入口点的汇编指令，使之先调用我们编写的函数，然后再调用原始的过程。在我们自己编写的函数内部，分别记录下接收到的指令包和要发送的指令包（如下图）。这样，网游客户端的发送包和接受包的监控就完成了。

2.分析网游指令的通用方法

能够获得网游指令样本数据之后，接下去就是实际的分析过程了。

一般而言，根据指令包所起的不同作用，可以将游戏指令包分为不同类别:

连接相关指令包 用于与游戏建立连接以及退出游戏时的指令包

玩家属性相关指令包 与玩家本身状态相关的指令包，在刚进入游戏系统时，服务器会发送过来大量的关于玩家角色基本信息的指令包。

环境相关指令包 由于告知玩家周围怪物/NPC/其他玩家状况的指令包

移动相关指令包 与走路相关的指令包

战斗相关指令包 与战斗相关的指令包

交易相关指令包 与交易相关的指令包

……

制作一款外挂，如果只要求基本功能的话，分析20，30个指令包应该就差不多了，如果要求功能比较完善的话，至多50个指令包也就差不多了。

具体分析的方法也很简单，无非是排除干扰因素，逐个击破，以及重复试验，确保分析结果正确。

下面的贴图是我对某一款游戏的28个指令包的具体分析。

三、游戏地图文件的破解与转换

地图文件的转换也是外挂制作中的一项任务，外挂中使用的地图一般仅需要知道道路通行信息就可以，对于其他的视觉信息是没有必要的，因些我们需要对游戏中的地图文件进行一定的转换工作。未被转换的游戏地图文件同样也是比较庞大的。由于不同的游戏运行商为了表达不同的游戏效果，因此其在地图文件上的格式也不尽相同，我们必须分析某款游戏的具体的地图文件格式。

3.1 地图文件格式的分析 一般情况下地图文件并未被加密，但有些也可能进行了加密。地图文件格式的分析即可以直接分析地图文件数据，也可以通过跟踪游戏客户端代码而破解出其实现过程。跟踪代码的方法同上述加密与解密算法的破解方法相似，所不同的是我们需要跟踪读写文件的函数。

3.2 外挂地图文件格式的组织 外挂地图文件格式是由外挂所需要的信息的要求决定的，在外挂中我们仅需要知道通路信息即可，因此外挂中的地图文件格式一般被组织为R＊C的格式。在R＊C的每一格中记录地图在该点的通行状态，一般地我们可以定义通行状态为0阻塞状态为1。然而事实上有些外挂公司为防止其他外挂制作者的破解，往往把地图文件进行一定的加密处理。

④.网游资源解析

游戏资源主要是指包含在网游客户端内的与游戏相关的数据资源：其中最重要的是地图资源，其他的资源包括NPC的位置坐标啊，地图传送点的坐标啊，等等。

1.地图资源解析

地图资源的解析是外挂制作中很重要的一个方面，寻路算法以及地图间的自动移动等都要依赖于地图资源。

外挂中所使用的地图资源不完全等同于游戏中使用的地图资源。游戏中使用的地图资源包含有更多的信息，而外挂所使用的地图称之为BOOL地图，它仅仅包含一项信息：指定的某一点是可移动点还是障碍点。

例如上图就是某个游戏地图所对应的BOOL地图，其中蓝色部分代表不可移动区域，白色部分代表可移动区域。（另外绿色的点是通过解析其他的资源文件获得的NPC的坐标点，红色点是传送点。）

那么如何获得游戏地图所对应的地图资源呢？通常在网游客户端的程序安装目录内会存在一个类似map或者res之类的目录。地图资源处在其中的可能性很大，而且地图资源本身的文件格式也比较明显，通常它都是以如下的格式存在的：

前面N个字节的地图头信息中肯定包含了地图的宽度信息和高度信息，设宽度和高度分别为W和H。后面的部分通常都是地图点信息。地图上的一个点通常由n个字节所构成，整个地图文件的大小由此计算而得： N + (W*H*n)。

所以判断是否是地图文件很简单，只要看一下文件的头，然后查看一下文件的大小。如果有一组文件满足上面的规律的话，那基本上可以肯定是地图资源文件了。

接下来就是要把游戏地图转化为BOOL地图。前面提到地图文件中地图上的一个点通常有n个字节所构成，里面含有很多丰富的信息，而BOOL地图所关心的是否是可移动点的信息通常仅需要用1位即可表示。（注意是位，1个字节有8位，所以n个字节总共有8n位）将一张游戏地图中所有点的该位信息收集起来组成的新的地图就是BOOL地图。它标示了地图中那些区域玩家是可移动的，那些区域是不可移动的，为后面的寻路算法提供了基础数据。

原理明白了，接下去就是写一个程序解析地图资源。大致步骤如下：

1. 打开一个地图文件

2. 读入地图文件的头信息，解析出地图宽度和高度

3. 读入地图点信息，对每一个点所代表的n字节数据执行位操作，提炼出其中某一位的信息，保存到自己的结构中。（此处我建议大家采用BMP格式的数据来保存提炼出来的位信息，好处有三点：一是保存完之后，直接可以用图片浏览工具查看结果，不必自己再写一个绘制的程序，二是使用BMP格式保存的话，保存的数据容量也小，三是在外挂中显示地图时可以将BMP图片直接作为背景图片贴在窗口上。）

由于之前你尚无法确定n字节中的哪一位代表了点的是否可移动属性，所以每一位你都要取一遍组成一幅地图，然后查看哪一幅和游戏地图最接近。多读几个地图文件做实验，很容易就可以确认下来的。

2.其他资源的解析

地图资源的解析是通过了解物理磁盘上地图文件的保存格式，然后自己写程序解析出来的，使用这种方法还可以解析其他很多资源信息。大家可以仔细观察游戏的安装目录，根据子目录和文件的名字可以分析出很多有用的信息。

但目前游戏厂商也越来越狡猾了，保存在磁盘上的资源文件通常进行了变形（压缩或者加密），使你无法通过简单的分析获得你所需要的信息。

一种解决办法就是观察游戏是如何处理变型的资源文件的。因为在游戏中资源肯定是以原始形式存在的，通常都是在游戏初始化的时候，从磁盘上读入变形的资源文件，然后将其恢复为原始状态的资源形式，我们就跟踪该段的处理过程，然后自己模仿写一段程序将变形资源恢复为原始资源形式。

另外一种方法就是直接从游戏的内存中读取有用的资源信息。该方法的理论依据就是：资源信息在游戏中肯定是明文形态，而且是被有序组织的，也有部分游戏是加密过的，不过认真分析一般抠取这些列表或者数据都不困难。比如，如果你想要获得所有游戏物品的信息列表，你可以随意选择一个物品名称，然后在游戏的内存中查找他的位置。所有的物品在游戏内存中肯定是以某种链表的形式存在的，你只要找到了一个，就可以顺藤摸瓜，找到该链表的头，然后自己写一个程序，读写游戏的内存空间，将整个游戏的物品列表全部读取出来。

四、外挂中智能AI部件的实现

游戏外挂智能AI部件的实现主要由算法与数据两部分组成，在一个具有推广意义的外挂中AI部分的设计是必不可少的。外挂AI部分的设计主要包括以下几方面的内容：游戏脚本指令体系、自动行走算法、自动杀怪算法和综合控制算法，而在上述算法中相关配置文件的设计是相应算法设计不可缺少的组成部分。

由于智能AI部件的实现是一个比较复杂的事情，同时具体的实现过程可以因每个人理解不同而具有完全不相同的实现代码与实现的数据结构，但一般地它们还是具有一定的共性。

AI部分也是对一个人的分析能力的考验，下面先粗略得介绍一下几个算法的作用与意义，深入的探讨在具体的各个章节中进行。

4.1 游戏脚本指令体系 脚本是使外挂具有通用性的一个实现方法，也是对游戏角色进行控制的手段。脚本指令体系设计的好坏直接关系到外挂的功能与性能。

4.2 自动行走算法 完成自动行走是一个必备的功能，自动行走性能的好坏主要由算法决定。要实现自动行走首先必须找到行走的道路，然后再沿指定道路前进。寻路算法目前主要由A*算法来处理。

4.3 自动杀怪算法 自动杀怪也是一个必备的功能，自动杀怪性能好坏直接关系到升级速度的快慢。

4.4 综合控制算法 一部分由脚本完成一部分由程序处理，综合控制与性能最为相关，综合控制处理的好坏直接影响升级的速度。

眼镜
Exp:741
中士

⑤.网游中的算法

1.寻路算法

外挂中最有名的，也最重要的一个算法就是寻路算法了。所谓寻路算法就是指给定一张地图数据，以及起始点和目标点，然后利用算法计算出一条路径来。它所依赖的数据基础是BOOL地图，这个我们在之前的讲述中已经成功获得了。下面讲一下具体的算法。

常用的寻路算法实现有二种，一种是A*算法，还有一种等高线算法。还记得在大学里面学过遍历图的二种算法吗，一种是深度优先，一种是广度优先。A*算法就对应了深度优先算法，而等高线算法则对应了广度优先算法。

A*算法是最常用的寻路算法了，不过它也有个很大的缺陷，那就是计算出来的路径通常是贴边的，所以如果你在游戏中观察用外挂控制的人物的走动的话，你会发现他通常是沿着障碍物的边走动的，走动起来显得很不自然。

A*算法和等高线算法在CSDN还有Gameres都有例子~这里就不重复了.

另外一点需要指出的是，寻路算法以及之前提到的BOOL地图的解析针对的都是2D的网游，那些纯3D的网游中的人物采用的是碰撞模型，一般3D游戏都是有高度的也就是Z坐标，它可以是正数也可以是负数。这个涉及的东西就多了去了这里就不在详说。

顺便提一下游戏中的走路。目前的2D网游中对于人物走动的处理方式主要有二种：一种是直接向服务器发送玩家要到达的目标地址，还有一种是以当前的坐标点为基点，给服务器发送相应的偏移量。

直接发送目标地址的方式，如果网游服务器端做的不够严谨的话（没有对玩家要移动的地址和玩家当前地址之间的距离进行校验），可能会存在瞬移的BUG可供外挂利用。（以前我就曾经碰到过一款有瞬移BUG的网游，利用外挂飞来飞去，飞得太猛了，后来就被游戏开发商给修正了。）

发送偏移量的移动方式见下图，●是当前玩家所在的位置，如果玩家需要向上（即向北）移动一步的话，则向服务器发送偏移量7，如果要向斜向角（即西南方向）移动一步的话，则发送偏移量4。很明显，发送偏移量的移动方式不存在瞬移的可能性。

五、外挂配置文件的构造与设计

外挂配置文件的设计是属于外挂智能AI设计中的一部分。实现AI功能的基础一方面是算法，另一方面数据。算法的介绍集中在《游戏智能AI部件的实现》章节中讲述，这儿主要解述外挂中的配置文件的设计。

配置文件的支持直接与实现某种功能相关，支持的内容越全面那么与脚本配合后所能实现的功能越强大，并且对智能AI过程的设计起到帮助的作用。在一般游戏外挂的配置文件中主要包括以下几部分内容，其分别为地图配置文件、过门配置文件、物品配置文件、怪物配置文件、NPC配置文件、装备配置文件、技能配置文件和战斗配置文件。

5.1 地图配置文件　这部分内容由《游戏地图文件的破解与转换》的章节进行详细的讲解。地图配置文件的意义除了起显示作用外，其最重要的作为是提供寻路算法的数据支持。

5.2 过门配置文件　现在的游戏大部分是采取小幅地图切换的模式，图与图进行切换的点我们称之为过门。由于过门之间在数据上是不连续的，因些我们需要为些建立过门之间的联系。过门配置文件的意义主要是提供图与图之间寻路算法的数据支持。

5.3 物品配置文件　物品配置主要决定物品是否被捡取、丢弃、购买、出售、修理和物品所属种类等，物品的各类决定物品被何种NPC所处理。

5.4 怪物配置文件 怪物配置主要决定怪物是否被攻击、躲避及用什么形式攻击、多少等级间被攻击等。

5.5 NPC配置文件　NPC配置主要决定NPC在何图、何坐标、处理何类工作等。

5.6 装备配置文件　装备配置主要决定何种职业用何种装备、何等级别用何等装备等。

5.7 技能配置文件　技能配置主要决定何技能在何级别被何职业所学习与练习等。

5.8 战斗配置文件　战斗配置主要决定何级别该在何地图级别等。

外挂配置文件的具体构造形式主要由其内容所决定，同时可以考虑是否对文件进行加密处理。

2.其他算法

外挂中除了寻路算法之外，还有其他的一些算法应用，比如地图间移动算法：在已知各个地图间传送点的情况下，计算出从地图A移动到地图B所要经过的所有地图，这同样是一个经典的图论算法问题。

此外还有打怪时如何搜索最近怪物的算法，以及最有效的自动战斗的算法，这些算法要根据每款游戏的实际情况而进行相应的变动。

六、脚本解析器设计与游戏脚本指令

我们为什么要进行脚本解析器设计呢？其目的主要是解决外挂通用性的问题，我们知道有些游戏运行商时常会调整或扩增其游戏中的一些数据。如果我们把这些信息硬编码在程序中，那么游戏运行商每修改一次数据，我们就要重新更改与编译外挂一次，这为外挂的推广使用带来了极大的不便。另外，脚本解析器也是作为智能AI处理的一个部分，游戏行为的实现很大一部分依赖于脚本代码的书写。因此设计脚本解析器与游戏脚本指令体系是十分重要的，其中脚本指令体系更为重要。

6.1 脚本解析器的设计 脚本解析器设计是属于编译原理的范畴，大家如对编译原理有所掌握，那么设计一个脚本解析器是相当容易的。脚本解析器的设计是与脚本指令体系相关的，离开了脚本指令体系的设计脚本解析器的设计也就免谈，因此在脚本解析器设计前我们首先需要设计好我们的脚本指令体系。

6.2 编译原理 编译原理是讲叙代码翻译的一门课程，编译原理主要涉及词法分析、语法分析、语法树构建、代码转译等方面的知识。现代的高级程序语言是属于形式语言的，它是按一定的格式与规则进行书写，从而表达一定的行为与逻辑。而对于脚本解析器来讲所涉及的内容则较为狭窄，一般情况下脚本解析器被设计为解设执行的程序体系，因此主要涉及词法分析与语法分析的内容。

6.2.1 词法分析 词法分析是指将我们编写的文本代码流解析为一个一个的记号，分析得到的记号以供后续语法分析使用。在词法分析中同样涉及到一错误的判断与处理。

6.2.2 语法分析 语法分析是将上述得到的记号按一定的规则进行检测，若符合某个规律则处理相应规律所对应的事情。语法分析最终可以将脚本代码的行为给解析出来，并最终完成脚本规定的行为。

6.3 脚本指令体系与组成 脚本指令体系是脚本设计中的核心，脚本指令体系设计的合理与优异与否直接与外挂能力与智能水平直接相关。一般情况下脚本支持的指令越多则脚本所能实现的能力越大。任何外挂中使用的脚本主要具有以下几个要素：自定义变量支持语句、类型识别能力、赋值语句、脚本流控制语句、系统变量支持语句、数值和字符串运算语句、比较语句及游戏操控语句等。

6.3.1 自定义变量支持语句 主要考虑变量的作用域与变量类型的问题。变量作用域在外挂中主要分为全局与局部两类。变量类型在外挂中主要分为数值型、字符串型和时间型三类。另外变量所能接收数据的长度也必须给予考虑。

6.3.2 类型识别能力 这类处理可以用显性或用非显性的方法进行处理，显性处理较为符合目前大多数编程语言的习惯。

6.3.3 赋值语句 分为数值型、字符串型与时间型三类数据的赋值问题。一般采用主流程序使用的方法，这种方式的用户较为广泛。

6.3.4 脚本流控制语句 分为条件语句、多分支语句、真性循环语句、假性循环语句、计数循环语句和强制跳转语句。

6.3.5 系统变量支持语句 与实际的游戏相关。比如游戏中角色的职业、级别等，另外也可以设定游戏中没有的但我们使用频繁的变量。系统变量的多少直接与脚本所能支持的功能和执行的性能有关。

6.3.6 数值和字符串运算语句 数值间有加减乘除等运算，字符串有相连定位查找等运算，时间有加减等运算。

6.3.7 比较语句 游戏中的比较语句主要包括数据间的比较、字符串的比较和时间的比较。比较方式主要有大小与相等。

6.3.8 游戏操控语句 与实际的游戏相关，比如买、卖、修、存、取、捡、丢、用、走、砍、挖、杀等，具体格式由游戏决定。

⑥.题外话

网游外挂产业真的是一个很有趣的领域，他主要是包含二个方面：技术和营销。

技术方面，网游外挂所涉及的技术之广并不逊色于一款网游所涉及的技术。其中技术难度最大的还是破解方面，据闻中国几大顶尖的破解组织高手大部分都与外挂破解有或多或少的联系。近几年随着大部分的游戏都运用了NProtect，在与NProtect的逐渐斗法过程中，战场从RING3层逐渐转移到系统内核层，也着实培养出了一批对于Window内核有深刻理解的高手，国人之幸啊。

讨论网游的技术是一个很有趣的话题，但个人认为更有趣的方面在于网游外挂的营销。由于近几年网游外挂制作团队的大量涌现。同一款游戏常常有着好几家外挂在互相竞争。这时候的外挂营销就像一个硝烟弥漫的战场，你不仅要与外挂同行竞争，还要和游戏发行商互相斗法，同时随着近几年相关法律的健全，你还要小心自己不要被相关政府部分盯上。

在与同行竞争的过程中，不仅要比拼技术（游戏采用了更严密的保护手段，你还要能破解它），比拼反应时间（游戏客户端有了更新，你要迅速反应），更要比拼营销手段，推出各种各样的优惠套餐，尽可能的拉拢住更多的客户，比如，如果外挂单月的收费是20，则半年只要100，全年只要180,这样表面上你好像少收了钱，但实际上你把客户和你牢牢的绑定住了，即使竞争对手推出更好更便宜的外挂，你也不用害怕，因为你有足够的时间来反应。类似的手段还有很多，这主要看你的营销水平了。

同时还要处理好与竞争对手之间的关系，能很快消灭对手的，则绝不留情。明显不如对手，且对手满怀敌意的，则干脆开放免费版，让大家都不好受。势均力敌的，能结成同盟的最好达成价格联盟，避免互相砍价给彼此造成损失。互相敌视的，则要尽量给竞争对手下绊子，比如攻击对手的站点啊，到他的论坛上面捣乱啊……,总之就是想尽办法击败对手，取得一家独大的场面，一旦你能在一款游戏里面吃独食，形成垄断，那外挂的价格就随你定了。够你爽的。

外挂的营销就像一个群雄并起的战国时代，大家各施手段，各凭本事，看谁能笑到最后。这里面发生过很多有趣的故事。不过近几年国内相关法律越来越严格了，所以并不建议大家踏入这个泥水潭。