关于客户端网络协议安全性测试工具

近日遇到一个内部工具，用于协议的安全性测试，具体的做法是改变客户端请求协议的某个值，来验证是否会引发不可预期的结果。在了解到其运行机制后，产生了一点看法，便有了此文的雏形，希望可以借此向老手进行探讨，向新手分享理解。

现有工具

现有的工具是以一个代理的形式，介于服务端与客户端之间，转发它们交互的数据包，并将其记录下来，之后通过对某个协议进行数值更改，从代理的位置将修改过的非正常的协议包发送至服务器等待服务器响应。

这种方案的工作方式，先是保证了由客户端发出的包一定能通过代理到达服务器，反过来服务器的包也一定能到客户端。由于工作机制的关系，服务器不能够区分来自客户端方向的包是否是由客户端发出且未经修改过的，所以即使是代理本身发出的协议，服务器也会将其当成客户端正常对其响应，之后代理会将该响应返回给客户端，这时我们就能够在客户端对该异常协议的响应进行结果校验。

缺点

该种方案的有效性，是建立在当客户端收到特定（测试目标）服务器回包之时，能够单独地“理解”该响应，不需要自身先前存在一个发包请求或是一个能够正确理解该响应的上下文环境。

比如，代理以释放技能协议的格式发送一条协议至服务器，并将服务器对此做出的响应返回给客户端，从客户端的角度看，仅是单独接收到一条自己释放技能协议的响应，如果客户端能够配合着这条独立的协议响应释放技能，那这样便能够证明这种方案对该技能测试有效。

再比如，玩家从商店购买道具是先走到商店处、点开商店、提交购买订单、进行订单结算，如果通过代理单独对订单结算这条协议进行测试，我们需要考虑当前角色位置作为前提条件，然后通过观察背包是否有出现购买商品并进行相应数值金币的扣除。因为该协议涉及到的上下文环境较为复杂，服务器并不一定能够正确的识别响应该协议，客户端也是如此。

改良思路

为了避免所提到的短处，可以将协议修改做成注入的形式，以配置表的方式，独立于程序之外存在，并实时监听该配置表的变化情况（watchdog）实现热更新，在需要进行测试时，通过修改该配置表实时同步到代理程序中，做到精确修改指定协议中的某个数值，同时不影响到修改协议以外的其他协议。

这样一来，即使是不熟悉该代理实现细节的同学，也能通过简单地了解配置表格式，从而自行修改参数进行测试。

具体实现参见 demo 代码中的代理模块。

该代理的核心在于，在转发客户端的协议至服务器的过程中，解析协议内容判断该协议是否有需要修改的属性存在于配置表中，如果有便将配置表中新的属性覆盖原有协议然后重新封装，而后将这个已更新的协议包转发给服务器，等待服务器响应的回包转发给客户端。

后续

测试思路看法

在进行安全性测试的时候，在服务端有对输入内容进行注入过滤的情况下，进行测试的属性值选择可以从两个角度去拓展：极值与非本类型值。测试所要达到的目的可以暂时制定成这样，当服务器协议受到破解，他人发送一个异常协议，而服务器不会对此做出异常的不可预期的响应。比如，当玩家请求购买数量为 “-1”，价格为 1 元宝的商品时，玩家身上的元宝数加 1 便是异常不可预期响应，而响应购买失败之类的相关错误码、没有响应或是直接断开连接则是正常响应。

局限：自动化

当制定好测试的角度可以从极值与非该类型值进行测试时，我们便可以准备进行制定一种通用模式来对需要测试的点进行自动化测试，但这种做法会存在一些问题。

一是由于我们是采用代理进行协议测试，是通过配置表建立在客户端发送的协议与其顺序的基础之上，因而自动化案例在该模式下只能建立在客户端上而不是代理。

二来安全性测试为一次性测试，一个测试点过一次即可，且需要根据测试协议所处不同环境之下制定不同的极值，适应性低（战斗中的负值可以解释为加血，而商品价格为负值则不合理），因此自动化收益较低。

第三，该测试的核心校验点在于，独立地校验服务端是如何对各个异常协议值进行响应，以错误码提醒或是直接丢弃该包不予响应，该响应与正常协议的响应差异在哪，而要对该响应行为进行判断也不便进行自动化。

局限：手动化

在不便进行自动化的情况下，我们回过头来看手动化，举个例子，当想对商店的所有商品进行测试时，成百上千的数量全部进行手动化明显也不合理。

因此，这方面无法以一种通用的模式来适应所以情况，计算机科学没有银弹，还需因地制宜，见机行事。