XEngine代理服务器文档

目录

[XEngine代理服务器文档 1](#_Toc193791321)

[前言 3](#_Toc193791322)

[阅读者 3](#_Toc193791323)

[概述 3](#_Toc193791324)

[相关模块 3](#_Toc193791325)

[一 技术结构 3](#_Toc193791326)

[1.1 目录结构 3](#_Toc193791327)

[1.2 源码目录 3](#_Toc193791328)

[1.3 协议解释 4](#_Toc193791329)

[二 配置环境 4](#_Toc193791330)

[2.1 WINDOWS 4](#_Toc193791331)

[2.2 LINUX 4](#_Toc193791332)

[2.3 MacOS 4](#_Toc193791333)

[三 配置说明 4](#_Toc193791334)

[3.1 基础配置 4](#_Toc193791335)

[3.2 最大配置 5](#_Toc193791336)

[3.3 时间配置 5](#_Toc193791337)

[3.4 日志配置 5](#_Toc193791338)

[3.5 报告配置 5](#_Toc193791339)

[3.6 代理配置 5](#_Toc193791340)

[四 通信协议 6](#_Toc193791341)

[4.1 Tunnel 6](#_Toc193791342)

[4.2 Socks 6](#_Toc193791343)

[4.3 Forwad 6](#_Toc193791344)

[4.3.1 登录 6](#_Toc193791345)

[4.3.2 枚举 7](#_Toc193791346)

[4.3.3 绑定 8](#_Toc193791347)

[4.3.4 转发 9](#_Toc193791348)

[4.4 Proxy 9](#_Toc193791349)

[附录 9](#_Toc193791350)

[附录1 更新历史 9](#_Toc193791351)

|  |  |  |  |
| --- | --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布 | 文件名称： | XEngine代理服务器文档 | |
| 适用版本： | V1.12.0.1001 | |
| 发布日期： | 2025-03-25 | |
| 拟 制： qyt | | |

# 前言

## 阅读者

开发人员,测试人员

## 概述

此文档包含了代理服务器的相关技术说明和接口定义!

## 相关模块

此服务使用了XEngine作为开发包.要使用此服务代码必须配置安装好XEngine开发环境.

请参考Readme配置环境

# 一 技术结构

采用的是TCP协议来作为基础通信协议

服务器启动会绑定四个端口,用来处理协议.

## 目录结构

* XEngine\_APPClient:应用例子代码目录
* XEngine\_Docment:文档目录
* XEngine\_Release:编译发布目录
* XEngine\_Source:源代码目录

## 源码目录

* XEngine\_ModuleConfigure:配置文件模块
* XEngine\_ModuleProtocol:协议解析与组包模块
* XEngine\_ModuleSession:会话处理模块
* XEngine\_ServiceApp:服务程序

## 协议解释

TCP私有协议采用的是XEngine定义的协议,你可以参考XEngine的协议文档详细了解协议的定义与解释.

# 二 配置环境

## 2.1 WINDOWS

在完成环境配置后.你可以进入代码目录.使用VS打开XEngine.sln,然后编译.

如果环境没有错误,编译会直接成功.包含5个模块和一个EXE程序

成功后,你需要拷贝XEngine\_Release下的文件到你编译的目录下.然后拷贝依赖的XEngine模块到你的编译目录下,运行XEngine\_ServiceApp.exe即可

注意：XEngine环境可以通过VSCopy脚本拷贝到你的编译目录下,前提是你配置好你的XEngine环境.

## 2.2 LINUX

如果没有错误,你可以在XEngine\_Release目录下看到编译的XEngine\_ProxyServiceApp

## 2.3 MacOS

MacOS需要13以及以上版本,编译和运行方式可以参考Linux.

# 三 配置说明

配置文件:XEngine\_Config.json

## 3.1 基础配置

* tszIPAddr:服务器IP地址
* bDeamon:是否守护进程运行
* nSocksPort:SOCKS代理服务端口
* nTunnelPort:HTTP隧道代理服务端口
* nForwardPort:转发代理端口
* nProxyPort:全代理转发端口

## 3.2 最大配置

XMax 配置

* MaxClient 最大允许客户端
* MaxQueue 最大允许队列
* IOThread:网络IO处理线程个数
* nForwardThread:转发代理线程个数

## 3.3 时间配置

XTime配置

* nTimeCheck:检测次数
* nSocksTimeout:SOCKS心跳检测超时时间
* nTunnelTimeout:Tunnel心跳检测超时时间
* nForwardTimeout:转发心跳检测超时时间
* nProxyTimeout:代理转发超时时间

## 3.4 日志配置

配置信息:XLog

* nMaxSize:最大日志文件大小
* nMaxCount:文件备份个数
* nLogLeave:日志级别
* nLogType:日志类型
* tszLogFile:日志保存路径

## 3.5 报告配置

XReport 配置

* bEnable:是否启用
* tszServiceName:服务名称
* tszAPIUrl:API地址

## 3.6 代理配置

XProxy配置

* bEnable:是否启用
* tszDefaultAddr:后台转发默认地址.没有规则匹配的将匹配此地址
* tszRuleIPAddr:规则IP转发数组.通过 – 符号分割,前面是原始IP或者端口,后面是转发的目标地址

# 四 通信协议

## 4.1 Tunnel

基于HTTP的TUNNEL标准协议,请参考相关RFC文档

## 4.2 Socks

基于SOCKSV5的标准协议,请参考相关RFC文档

## 4.3 Forwad

转发协议是私有的定制协议,可以通过这个协议获取可以转发的用户和列表,并且转发数据.

转发协议支持匿名和命名转发.他们的区别是:

* 命名转发需要双方登录验证才可以做转发
* 匿名转发是由转发方发起的一个连接,需要对方提供了一个服务.然后请求方连接成功后在做数据转发

### 4.3.1 登录

不登录无法获取得到列表信息

#### 4.3.1.1 请求

协议头:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = *ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_AUTH*

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_FORWARD\_LOGREQ

unPacketSize = sizeof(*XENGINE\_PROTOCOL\_USERAUTH*)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

负载:

用户信息

#### 4.3.1.2 回复

协议头:wReserve表示成功还是失败,0等于成功

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = *ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_AUTH*

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_FORWARD\_LOGREP

unPacketSize = 0

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 4.3.2 枚举

#### 4.3.2.1 请求

协议头:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_USER\_FORWARD

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_FORWARD\_LISTREQ

unPacketSize = 0

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

#### 4.3.2.2 回复

协议头:wReserve表示成功还是失败,0等于成功

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_USER\_FORWARD

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_FORWARD\_LISTREP

unPacketSize = JSONSIZE

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:

{

"Array":[

{

"bForward":false,

"st\_UserAuth":{

"enClientType":0,

"enDeviceType":0,

"tszDCode":"",

"tszUserName":"test",

"tszUserPass":""

},

"tszDstAddr":"",

"tszSrcAddr":"127.0.0.1:64900"

}

],

"Count":1

}

### 4.3.3 绑定

绑定一个转发客户端在请求后,另一端也会收到同样的请求.另一端收到的请求只有协议头.

匿名转发不会转发请求给服务端

#### 4.3.3.1 请求

协议头:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_USER\_FORWARD

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_FORWARD\_NAMEDREQ

unPacketSize = JSONSIZE

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

{  
    **"tszDstAddr"**:**"192.168.1.6:1000"**  
}

#### 4.3.3.2 回复

协议头:wReserve表示成功还是失败,0等于成功

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = *ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_*FORWARD

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_FORWARD\_NAMEDREP

unPacketSize = 0

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 4.3.4 转发

如果绑定成功,以后的数据都会转发到相应的客户端

## 4.4 Proxy

全转发代理支持,就是负载均衡的一种,可以把入口流量转发到后台指定机器上面.可以做流量解析和控制

# 附录

## 附录1 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 更新日期 | 更新说明 | 修改者 | 适用版本 |