Operation Blockbuster Goes Mobile

By Anthony Kasza, Juan Cortes, Micah Yates

Published: 2017-11-20 · Archived: 2026-04-05 18:16:08 UTC

Unit 42 has discovered a new cluster of malware samples, which targets Samsung devices and Korean language speakers, with relationships to the malware used in Operation Blockbuster. The specific points of connection between these new samples and Operation Blockbuster include:

- payloads delivered by the macros discussed in Operation Blockbuster Sequel
- malware used by the HiddenCobra threat group
- malware used in the 2016 attack on the Bangladesh SWIFT banking system
- APK samples mimicking legitimate APKs hosted on Google Play

Although Unit 42 cannot provide a full picture of the details surrounding the delivery of these samples, we are confident this activity targets Korean language speakers who use Samsung devices. Based on this evidence we believe this new malware is likely targeting South Koreans.
The newly discovered samples show new capabilities not previously documented. A strong relationship between previously identified malware samples attributed to these campaigns and the newly discovered samples examined in this report.

New Malware Cluster
At the center of the cluster of new malware samples is a PE (ed9e373a687e42a84252c2c01046824ed699b32add73dcf3569373ac929fd3b9) uploaded to VirusTotal with the filename "JAVAC.EXE". The sample requires two command line parameters to run, the first is a port number which the binary binds to, acting as a webserver, and the second is also a port number which is used for encrypted protocol communications.
The first port mimics an Apache server, using header values that Apache would use and will serve different resources to requests on the port, depending on the User-Agent header values used. Some of the responses given are embedded in the original PE, whilst others are expected to be found on the local disk. The following JavaScript files are embedded in the resource section of JAVAC.exe:

Filename	SHA256	Purpose
jquery50.js	2b15e4289a3eb8e4eb8c2343895002dde7f5b2791e3c799b4f869be0aa85d2e8	Gets and sets client HTTP Cookie header values e.g. "GoogleAppCookie". Redirects clients to "main.js"
jquery52.js	b183625c006f50f2b64ebe0aebda7b68ae285e53d1b4b00c8f49cde2dfc89348	Gets and sets client HTTP Cookie header values e.g. "GoogleAppCookie". Redirects clients to "update.js"
jquery99.js	941cd0662cae55bc06727f1d658aba67f33442e63b03bebe012dad495e9e37dc	Redirect all client requests to mboard_ok.css.
main.js	790662a047047b0470e2f243e2628d8f1b62794c1359b75ed9b856325e9c961a	Collect system information and invoke a system shell. These are used to accomplish the following: Install and invoke an APK Write an ELF file to disk on the client
umc.apk	4694895d6cc30a336d125d20065de25246cc273ba8f55b5e56746fddaadb4d8a	Three nested APKs which ultimately lead to a backdoor APK

		implant. This file is likely installed silently by visiting the next resource with an HTTP client. Further details on this APK follow below.
update.js	cf3e9baaac7efcaff8a9864da9f12b4115ba3f148ae5cfc21f3c158f6182b792	Redirect all client requests to a URL which triggers a vulnerability in Samsung devices to install an APK.

The system name this PE HTTP server is intended to execute on has a hostname of "RUMPUS-5ED8EE00". This is checked by JAVAC.exe during execution. Besides the resources listed in the table above, it is important to note that JAVAC.exe expects additional files located on the system due to some of the resources referencing local JavaScript files. These include the following filenames:

- mboard_ok.css
- node_n.js
- node_e.js
- node_g.js
- node_p.js
- node_ok.js
- node_nc.js
- node_ex.js

We have not been able to obtain copies of these resources.
Related ELF ARM Samples
The ELF ARM file embedded in main.js is written to HTTP clients' disks by the logic in main.js. Below is a table outlining indicators from this embedded ELF ARM.

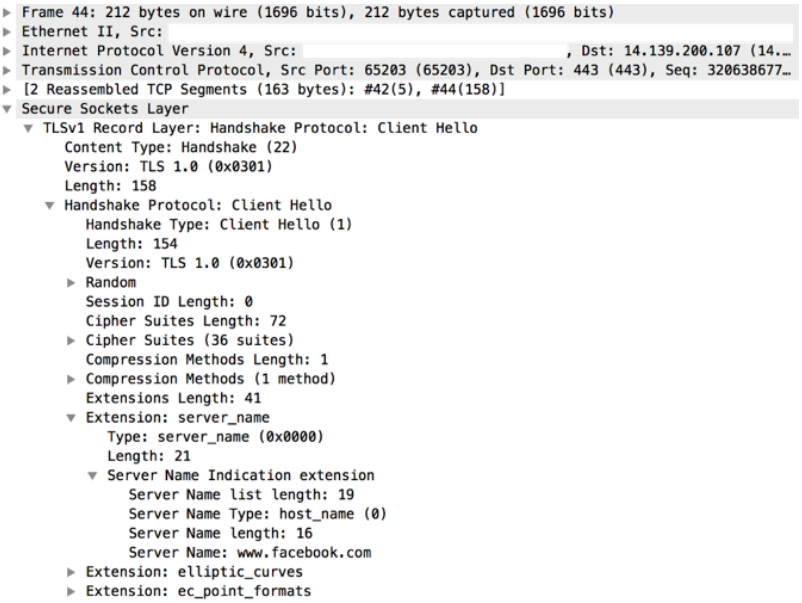
SHA256	**Description**	**Embedded IPv4 Addresses**
0ff83f3b509c0ec7070d33dceb43cef4c529338487cd7e4c6efccf2a8fd7142d	ELF ARM file embedded in main.js	97.211.212.31 14.139.200.107 175.100.189.174

This ELF ARM file is one of three we identified. These ELF files are similar to PE files named Cruprox by Symantec, Manuscrypt by Kaspersky, and Clevore by Trend Micro. The ELF ARM samples contain lists of domains (used for deception) and IPv4 addresses (used for command and control). These domains and IPv4 addresses are used to generate crafted TLS sessions similarly to the "fake TLS" communication mechanisms in section 4.3.3.1 of the Operation Blockbuster report by Novetta.
The ELF ARM samples choose one of the embedded domains to populate an SNI field of a TLS connection to one of the embedded IPv4 addresses. By doing command and control in this way an analyst observing the connection stream only sees what looks like (but is not) a TLS connection to a legitimate domain name. The domain names included in 0ff83f3b509c0ec7070d33dceb43cef4c529338487cd7e4c6efccf2a8fd7142d are as follows:

- myservice.xbox[.]com
- uk.yahoo[.]com
- web.whatsapp[.]com
- www.apple[.]com
- www.baidu[.]com
- www.bing[.]com
- www.bitcoin[.]org
- www.comodo[.]com
- www.debian[.]org
- www.dropbox[.]com

- www.facebook[.]com
- www.github[.]com
- www.google[.]com
- www.lenovo[.]com
- www.microsoft[.]com
- www.paypal[.]com
- www.tumblr[.]com
- www.twitter[.]com
- www.wetransfer[.]com
- www.wikipedia[.]org

An example TLS "Client Hello" record generated by 0ff83f3b509c0ec7070d33dceb43cef4c529338487cd7e4c6efccf2a8fd7142d is given below. It includes a legitimate domain name in its SNI field yet is sent to a command and control IPv4 address.



By examining strings, binary functions, and embedded IPv4 addresses of 0ff83f3b509c0ec7070d33dceb43cef4c529338487cd7e4c6efccf2a8fd7142d, we were able to hunt for and locate two additional ELF ARM samples. Below is a table of the related ELF ARM samples:

SHA256	Description
800f9ffd063dd2526a4a43b7370a8b04fbb9ffeff9c578aa644c44947d367266	ELF ARM file likely of the same malware family as 0ff83f3b509c0ec7070d33dceb43cef4c529338487cd7e4c6efccf2a8fd7 Embedded in 06cadaac0710ed1ef262e79c5cf12d8cd463b226d45d0014b2085432c (described below)
153db613853fb42357acb91b393d853e2e5fe98b7af5d44ab25131c04af3b0d6	ELF ARM file likely of the same malware family as 0ff83f3b509c0ec7070d33dceb43cef4c529338487cd7e4c6efccf2a8fd7

Related APK Samples

In addition to ELF ARM files the HTTP Server can also serve APK files. As previously stated, an APK with SHA256 4694895d6cc30a336d125d20065de25246cc273ba8f55b5e56746fddaadb4d8a is embedded as a resource in the HTTP PE server sample with a name of "umc.apk".

Umc.apk defines intent filters to receive events from the Android operating system when the APK is replaced (PACKAGE_REPLACED), when the device receives a text message (SMS_RECEIVED), and when the device is in use by

a user (USER_PRESENT). Umc.apk installs an embedded APK with the SHA256 value of a984a5ac41446db9592345e547afe7fb0a3d85fcbbbdc46e16be1336f7a54041.
A984a5ac41446db9592345e547afe7fb0a3d85fcbbbdc46e16be1336f7a54041 has a name of "install.apk".
The purpose of install.apk is to cleanup umc.apk and install a third APK with a SHA256 hash of 4607082448dd745af3261ebed97013060e58c1d3241d21ea050dcdf7794df416 and a name of "object.apk".
Object.apk is the final malicious payload. This APK ensures that it is running when the device is booted and provides backdoor capabilities to its controller.

- Record the microphone
- Capture from the camera
- Upload, execute, and manipulate local files
- Download remote files
- Record GPS information
- Read contact information
- Observe SMS or MMS messages
- Record web browsing history and bookmarks
- Scan and capture WiFi information

Below is an image of decompiled code from a main component of the backdoor. It shows the internal version number for this APK is “4.2.160713” it is unclear if this is an accurate representation of the number of iterations of development undertaken on this malware family, or if it is to give the APK an air of legitimacy.

```
public class WideApplication
{
  public static AudioRecorder m_ar;
  public static boolean m_bUSBConn = false;
  public static CameraSnapshot m_cam;
  public static GpsPosition m_gps;
  public static MSGManager m_msg;
  public static int m_nDone = 0;
  public static final int m_nMaxSleep = 2880;
  public static int m_nSleep;
  public static PowerManager m_pm;
  public static String m_strConfigPath;
  public static String m_strPropertyPath;
  public static String m_strUID;
  public static String m_version = "4.2.160713";
  public static WireManager m_wifi;
```

Configuration information for object.apk is included in the APK as a resource named "assest.png". The configuration information can be decoded using the following Python function:

```
def cnfdecr(s):
  b = ''
  for each in s:
    tmp = ord(each)
    tmp = tmp - 55
    tmp = tmp ^ 0x12
    b += chr(tmp)
  return b
```

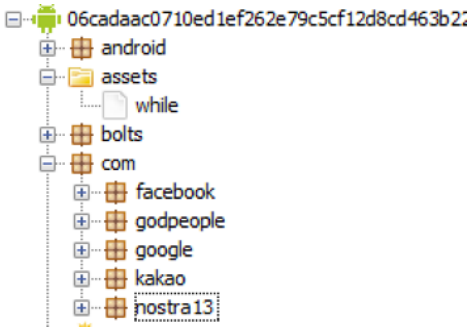
The decoded configuration values and their purposes follow:

Value	Purpose
4	Proxy Count
113.10.170.98	IPv4
443	Port Number
98.101.211.250	IPv4
443	Port Number

173.0.138.250	IPv4
443	Port Number
192.168.1.49	IPv4
443	Port Number
60	Sleep Time
5	Max Repetition Count

Following our analysis of the payload APK, we were able to locate an additional related APK. The APK with SHA256 hash value of 06cadaac0710ed1ef262e79c5cf12d8cd463b226d45d0014b2085432cdabb4f3 contains 800f9ffd063dd2526a4a43b7370a8b04fbb9ffeff9c578aa644c44947d367266, one of the ELF ARM files discussed in a table under the section titled, "Related ELF ARM Samples".

This APK, 06cadaac0710ed1ef262e79c5cf12d8cd463b226d45d0014b2085432cdabb4f3, contains resources which reference legitimate applications of varying popularity. We hypothesize the inclusion of these resources are to disguise the application's true intent and to make the application seem legitimate. The inclusion of KaKaoTalk resources leads us to believe this APK is targeting South Koreans. The image below shows some of the referenced mobile applications resources:



The purpose of 06cadaac0710ed1ef262e79c5cf12d8cd463b226d45d0014b2085432cdabb4f3 is to execute the ELF ARM file is contains. Below shows decompiled source code of the "com.godpeople.GPtong.ETC.SplashActivity" resource in the APK which contains the main functionality of the APK. It executes the ELF ARM file named "while" and logs activity to the debug log named "snowflake".

```
@SuppressLint(value={"NewApi"}) private void execute() {
    String v0 = this.getFilesDir().getPath();
    String v3 = String.valueOf(v0) + "/while";
    this.copyAssets("while", v0);
    new File(v3).setExecutable(true);
    try {
        Runtime.getRuntime().exec(v3);
        Log.d("snowflake", "success");
    }
    catch(IOException v1) {
        Log.d("snowflake", "fail");
        v1.printStackTrace();
    }
}

protected void onActivityResult(int arg2, int arg3, Intent arg4) {
    switch(arg2) {
        case 1: {
            if(arg3 == -1) {
                this.finish();
            }

            break;
        }
    }
}

public void onCreate(Bundle arg3) {
    super.onCreate(arg3);
    this.execute();
    this.setContentView(2130903067);
    if(ClassCommon.config_setting == null) {
        ClassCommon.config_setting = this.getSharedPreferences("config_setting", 0);
    }
```

Relationships to Known Samples

Originally, the PE server was identified by its binary overlaps with the following samples:

- 410959e9bfd9fb75e51153dd3b04e24a11d3734d8fb1c11608174946e3aab710
- 4cf164497c275ae0f86c28d7847b10f5bd302ba12b995646c32cb53d03b7e6b5

When executing, both samples create the mutex "FwtSqmSession106839323_S-1-5-20" which has ties to Operation Blockbuster and the attacks on the SWIFT banking system. Once this overlap in indicators was identified, and manual investigation began, additional overlaps began to emerge.
Additional functional code overlaps are found between the following samples and the PE server:

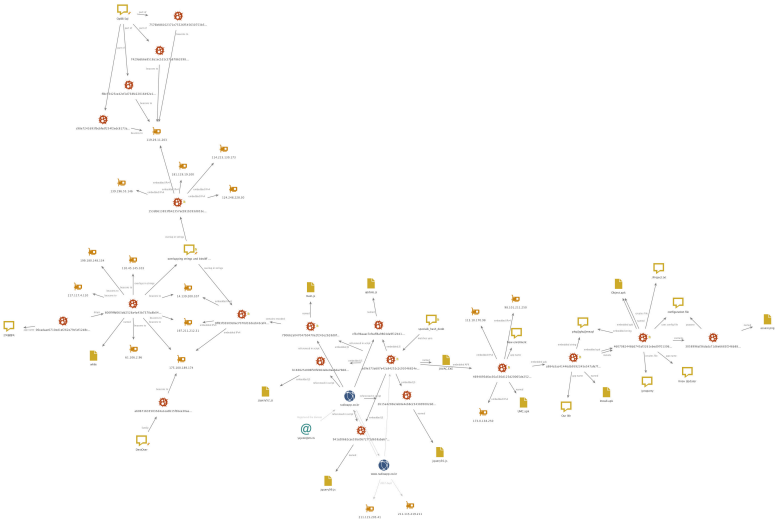
- 1d195c40169cbdb0f50eca40ebda62321aa05a54137635c7ebb2960690eb1d82
- af71ba26fd77830eea345c638d8c2328830882fd0bd7158e0abc4b32ca0b7b74

The PE server sample is not the only sample with ties to previously identified malware. Infrastructure reuse also exist between the IPv4 addresses embedded in ELF ARM files detailed in the previous section and previously identified malware. For example, 175.100.189.174 is embedded in 800f9ffd063dd2526a4a43b7370a8b04fbb9ffeff9c578aa644c44947d367266 and is also contacted by a606716355035d4a1ea0b15f3bee30aad41a2c32df28c2d468eafd18361d60d6, a documented Destover sample.
Another example of IPv4 address reuse is 119.29.11.203. This IPv4 address is embedded in the ELF file with SHA256 of 153db613853fb42357acb91b393d853e2e5fe98b7af5d44ab25131c04af3b0d6 and is also contacted by 7429a6b6e8518a1ec1d1c37a8786359885f2fd4abde560adaef331ca9deaeefd which is a PE payload delivered by the macros in the following malicious documents:

- 7576bfd8102371e75526f545630753b52303daf2b41425cd363d6f6f7ce2c0c0
- ffdc53425ce42cf1d738fe22016492e1cb8e1bc657833ad6e69721b3c28718b2
- c98e7241693fbcbfedf254f2edc8173af54fcacebb7047eb7646235736dd5b89

These macros share the same logic as macros discussed by Unit42 in previous reports.

Final Thoughts
It is clear that source code was reused between previously reported samples and the cluster of new samples outlined by Unit 42. Additionally, command and control IPv4 addresses were reused by the malware discussed in this analysis. Technical indicators as well as soft indicators, such as APK themes and names, provide soft and tenable ties to the actors behind Operation Blockbuster and the HiddenCobra group.
The image below summarizes all of the relationships presented in this report:



Attribution is difficult to confidently achieve even with an in-depth technical knowledge and large pool of telemetry to hunt through. Without targeting and delivery information this report offers a partial perspective on this new activity targeting Korean speaking Samsung users.
Palo Alto Networks customers can review this cluster of newly discovered malware by examining the GoingMobile AutoFocus tag.
Unit 42, before publication, notified both Samsung and the KrCERT of the activity detailed here. We would like to thank both organizations for working so quickly with us.
Indicators of Compromise
SHA256
06cadaac0710ed1ef262e79c5cf12d8cd463b226d45d0014b2085432cdabb4f3
0ff83f3b509c0ec7070d33dceb43cef4c529338487cd7e4c6efccf2a8fd7142d
153db613853fb42357acb91b393d853e2e5fe98b7af5d44ab25131c04af3b0d6
1d195c40169cbdb0f50eca40ebda62321aa05a54137635c7ebb2960690eb1d82
2b15e4289a3eb8e4eb8c2343895002dde7f5b2791e3c799b4f869be0aa85d2e8
410959e9bfd9fb75e51153dd3b04e24a11d3734d8fb1c11608174946e3aab710
4607082448dd745af3261ebed97013060e58c1d3241d21ea050dcdf7794df416

4694895d6cc30a336d125d20065de25246cc273ba8f55b5e56746fddaadb4d8a
4cf164497c275ae0f86c28d7847b10f5bd302ba12b995646c32cb53d03b7e6b5
7429a6b6e8518a1ec1d1c37a8786359885f2fd4abde560adaef331ca9deaeefd
7576bfd8102371e75526f545630753b52303daf2b41425cd363d6f6f7ce2c0c0
790662a047047b0470e2f243e2628d8f1b62794c1359b75ed9b856325e9c961a
800f9ffd063dd2526a4a43b7370a8b04fbb9ffeff9c578aa644c44947d367266
941cd0662cae55bc06727f1d658aba67f33442e63b03bebe012dad495e9e37dc
a606716355035d4a1ea0b15f3bee30aad41a2c32df28c2d468eafd18361d60d6
a984a5ac41446db9592345e547afe7fb0a3d85fcbbbdc46e16be1336f7a54041
b183625c006f50f2b64ebe0aebda7b68ae285e53d1b4b00c8f49cde2dfc89348
c98e7241693fbcbfedf254f2edc8173af54fcacebb7047eb7646235736dd5b89
cf3e9baaac7efcaff8a9864da9f12b4115ba3f148ae5cfc21f3c158f6182b792
ed9e373a687e42a84252c2c01046824ed699b32add73dcf3569373ac929fd3b9
ffdc53425ce42cf1d738fe22016492e1cb8e1bc657833ad6e69721b3c28718b2

Mutexes
FwtSqmSession106839323_S-1-5-20

IPv4s

110.45.145.103
113.10.170.98
114.215.130.173
119.29.11.203
124.248.228.30
139.196.55.146
14.139.200.107
173.0.138.250
175.100.189.174
181.119.19.100
197.211.212.31
199.180.148.134
211.115.205.41
217.117.4.110
61.106.2.96
98.101.211.250

Domains

www.radioapp[.]co[.]kr

Filenames
JAVAC.EXE
jquery50.js
jquery52.js
jquery99.js
main.js
umc.apk
update.js
mboard_ok.css
node_n.js
node_e.js
node_g.js
node_p.js
node_ok.js
node_nc.js
node_ex.js
object.apk
Install.apk
while
