

تقرير تحليل برمجيات

ضمن مسابقة OPCDE 2018 لاستخراج العلم

١٣ ابريل, ٢٠١٨

تم إعداده بواسطة محمد المدوّح

ملخص المراجعات

التاريخ	الإصدار	التحرير بواسطة	ملاحظات
١٣ ابريل, ٢٠١٨	١,٠	محمد المدوّح	النسخة النهائية من الإصدار الأول

نقدم في هذا التقرير شرح تفصيلي لتحليل تطبيق للهواتف الذكية يعمل على نظام الأندرويد "ctf.apk" لاستخراج العلم "flag" و رابط تسليم التقرير "URL" ضمن مسابقة "capture the flag" المقدمة من المهندس محمد الدّوب بالتعاون مع "OPCDE" حيث يمنح الفائز تذكرة مجانية لحضور مؤتمر "OPCDE 2018" المقام في دبي بتاريخ السادس من ابريل ٢٠١٨.

من خلال تحليلنا للتطبيق, استطعنا استخراج رابط التسليم المخفي في أحد عناصر التطبيق المشفرة "https://comae.typeform.com/to/XyWXds" و تمكنا من استخراج العلم "OPCDEBBQOMG_I_WON" المخفي بأكثر من ترميز و المتواجد أيضا ضمن العنصر المشفر من التطبيق.

استخدمنا في هذا التحليل أدوات إما مفتوحة المصدر أو متاحة مجاناً للجميع. و اتبعنا طريقة استخراج لا تتطلب أي علم سابق عن منصة الأندرويد أو تطوير البرمجيات عليها.

التحضير للتحليل و الأدوات المستخدمة

قمنا بتحميل ملف المسابقة و فك تشفيره و نسخه إلى جهازين حاسب آلي.
الأول يعمل على نظام "Windows 10" و الآخر يعمل على نظام "Linux: SIFT Workstation".
في نظام التشغيل "Windows 10", استخدمنا الأدوات أدناه:

١. Zip-7 لاستخراج محتويات ملفات بامتداد ".zip", ".apk"
٢. HxD لمعاينة محتويات الملفات بترميز Hex
٣. XorFiles لفك تشفير الملفات التي تستخدم xor كخوارزمية للتشفير

في نظام التشغيل "Linux: SIFT Workstation" استخدمنا:

١. foremost لاستخراج الملفات المخفية داخل ملف معيّن

كذلك استخدمنا عدة أدوات متوفرة على شبكة الويب للتحويل من ترميز لآخر:

١. للتحويل من Hex إلى ASCII:
<https://www.rapidtables.com/convert/number/hex-to-ascii.html>
٢. للتحويل من Base32 إلى ASCII:
https://emn178.github.io/online-tools/base32_decode.html
٣. للتحويل من Base64 إلى ASCII:
<https://www.base64decode.org>

تحليل التطبيق و استخراج العلم و رابط التسليم

قمنا بدايةً باستخراج محتويات ملف "ctf.apk" باستخدام أداة "7-zip" لينتج معنا الملفات أدناه:

```
assets .١
META-INF .٢
res .٣
AndroidManifest.xml .٤
classes.dex .٥
resources.arsc .٦
```

لفت انتباهنا صورة معطوبة في المسار "\assets\ctf.png". فقمنا بفتحها باستخدام HxD لتحليلها و ذلك بمقارنتها بصورة أخرى (عشوائية) سليمة بامتداد "ctf.png" كما هو موضح أدناه.

Offset(d)	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
00000000	9B	42	5C	55	1F	18	08	18	12	12	12	1F	5B	5A	56	40	>B\U.....[ZV@
00000016	12	12	13	43	12	12	12	8F	1A	10	12	12	12	90	5B	CE	...C.....[Î
00000032	7C	12	12	12	13	61	40	55	50	12	BC	DC	0E	FB	12	12a@UP.¼Û.û..
00000048	12	16	75	53	5F	53	12	12	A3	9D	19	EE	73	17	12	12	..uS_S...f...îs...
00000064	12	1B	62	5A	4B	61	12	12	00	66	12	12	00	66	13	CC	..bZKa...f...f.Î
00000080	74	0D	6A	12	12	ED	B7	5B	56	53	46	6A	4C	6E	EF	15	t.j...í·[VSFjLnî.
00000096	A6	36	D5	67	37	98	D4	48	87	87	F7	AF	E5	CC	69	FD	6Ög7~ÔH#÷÷~âîiý
00000112	AF	AD	CC	C9	AC	CF	A5	6F	11	7A	2A	00	9C	16	5B	02	-.îÉ-îÿo.z*.œ.[.
00000128	26	32	13	08	0A	93	B0	07	1B	C2	01	36	53	C3	90	86	&2..."°..Â.6SÃ.+

شكل ١: عينة من ملف ctf.png

Offset(d)	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG.....IHDR
00000016	00	00	00	69	00	00	00	25	08	06	00	00	00	41	18	C7	...i...%.....A.Ç
00000032	2D	00	00	0A	73	49	44	41	54	68	DE	ED	9B	5B	8C	25	-...sIDATHpí>[E%
00000048	45	19	C7	FF	DF	57	D5	7D	CE	CC	CE	EC	EC	CE	0E	3B	E.ÇýßWÖ}ÎÎÎÎÎÎ.;
00000064	28	10	C9	CA	25	8B	08	91	20	D1	44	C4	CB	83	9B	60	(.ÉÊ%<..' ÑDÄÊf>`
00000080	62	40	E3	83	4F	6A	A2	26	42	82	C1	07	1F	34	46	91	b@âfOjç&B,Á...4F`
00000096	68	78	30	41	8D	F1	91	07	13	2F	28	D7	44	13	2F	18	hx0A.ñ'.../(xD./.
00000112	43	B2	59	10	02	02	8B	30	BB	73	D9	99	D9	B9	EE	99	C²Y....<0»sÜ™Ü¹î™
00000128	FB	B9	75	77	55	7D	3E	54	75	9F	3E	67	66	90	7D	F5	û¹uwU}>TuŸ>gf.}ö

شكل ٢: عينة من صورة عشوائية بامتداد "ctf.png"

لاحظنا أوجه تشابه استنتجنا منها الآتي:

١. ملف "ctf.png" يبدو مشفر و غير معطوب.
٢. تم استخدام مفتاح تشفير تناظري "symmetric" بحجم 1 byte و قيمته 0x12

و من خلال ذلك اشتبهنا بأن تكون خوارزمية التشفير إما "shift up" أو "xor" و لكن تبين لنا أن الملف المشفر "ctf.png" يحتوي على انخفاض لبعض القيم الثابتة بدلاً من ارتفاعها. مثلاً، ثاني byte في شكل ٢، انخفض من 0x50 إلى 0x42 كما هو موضح في شكل ١. و من خلال ذلك قررنا محاولة فك تشفير الملف بخوارزمية xor.

باستخدام HxD, قمنا بفتح "ctf.png" و استبدال جميع القيم بقيمة 0x12 و من ثم حفظه في ملف منفصل باسم "ctf.key" كما هو موضح في شكل ٣.

```
Offset(d) 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
00000000 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 .....
00000016 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 .....
00000032 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 .....
00000048 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 .....
00000064 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 .....
00000080 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 .....
00000096 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 .....
00000112 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 .....
00000128 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 .....
شكل ٣: عينة من ملف ctf.key
```

ثم قمنا باستخدام أداة XorFiles لإجراء عملية xor بين "ctf.png" و "ctf.key" حتى ظهرت معنا الصورة أدناه. و قمنا بتسميتها "out.png"



شكل ٤: "out.png"

بعد ذلك قمنا باستعراض الملف باستخدام HxD و تبين لنا أمران:

١. تبين لنا وجود رابط موقع و عند استخراجها و محاولة فتحه, حصلنا على صفحة الويب الموضحة في الشكل رقم ٦.

```
Offset(d) 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
00117648 00 00 24 00 68 00 74 00 74 00 70 00 73 00 3A 00 ..$.h.t.t.p.s...
00117664 2F 00 2F 00 63 00 6F 00 6D 00 61 00 65 00 2E 00 /. /.c.o.m.a.e...
00117680 74 00 79 00 70 00 65 00 66 00 6F 00 72 00 6D 00 t.y.p.e.f.o.r.m.
00117696 2E 00 63 00 6F 00 6D 00 2F 00 74 00 6F 00 2F 00 ..c.o.m./t.o./
00117712 58 00 79 00 57 00 58 00 64 00 53 00 00 00 00 02 X.y.W.X.d.S.....
شكل ٥: "https://comae.typeform.com/to/XyWXdS"
```



OPCODE 2018 - Win Your Complementary Ticket!

Enter the flag and win a free ticket!

Continue

press ENTER

شكل ٦: موقع تسليم التقرير

٢. اشتبهنا بوجود محتوى مخفي لملف مضغوط و ذلك بملاحظتنا لقيمة 0x504B (PK) بشكل متكرر في آخر ملف "out.png" كما هو موضح أدناه. تتواجد قيمة "0x504B" كمفتاح تعريف "signature" للملفات المضغوطة.

```
Offset(d) 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
00118672 4E 46 2F 41 50 4B 4B 45 59 2E 44 53 41 50 4B 01 NF/APKKEY.DSAPK.
00118688 02 14 00 14 00 08 08 08 00 00 98 9F 01 70 EF 90 .....~ÿ.pi.
00118704 50 5A 04 00 00 D4 0D 00 00 13 00 00 00 00 00 00 PZ...Ô.....
00118720 00 00 00 00 00 00 00 89 06 00 00 41 6E 64 72 6F .....%...Andro
00118736 69 64 4D 61 6E 69 66 65 73 74 2E 78 6D 6C 50 4B idManifest.xmlPK
00118752 01 02 14 00 14 00 08 08 08 00 00 98 9F 01 7C 05 .....~ÿ.|.
00118768 76 4E 77 1C 00 00 8C 3A 00 00 0B 00 00 00 00 00 vNw...E:.....
00118784 00 00 00 00 00 00 00 00 24 0B 00 00 63 6C 61 73 .....$...clas
00118800 73 65 73 2E 64 65 78 50 4B 01 02 0A 00 0A 00 00 ses.dexPK.....
00118816 08 00 00 00 98 9F 01 01 6E D4 78 85 1C 00 00 85 ....~ÿ..nÔx.....
00118832 1C 00 00 1C 00 00 00 00 00 00 00 00 00 00 00 00 .....
00118848 00 D4 27 00 00 72 65 73 2F 64 72 61 77 61 62 6C .Ô'..res/drawabl
00118864 65 2F 69 63 5F 6C 61 75 6E 63 68 65 72 2E 70 6E e/ic_launcher.pn
00118880 67 50 4B 01 02 0A 00 0A 00 00 08 00 00 98 9F gPK.....~ÿ
```

شكل ٧: اشتباه وجود ملف مضغوط مخفي

قمنا بنقل ملف "out.png" إلى جهازنا الآخر "SIFT Workstation" و استخدام أداة foremost لاستخراج الملف المضغوط من الصورة و ذلك بتنفيذ الأمر الآتي:

```
foremost -t zip -i out.png -o ~/Pictures/CTF
```

ثم قمنا بنقل الملف المضغوط و لينطلق عليه اسم "out.zip" إلى جهاز الـ "Windows 10" ثم قمنا بفك ضغط الملف باستخدام "7-zip" ليتبين لنا أنه عبارة عن تطبيق أندرويد آخر يحتوي على الملفات أدناه:

١. META-INF

```

res.٢
AndroidManifest.xml.٣
classes.dex.٤
resources.arsc.٥

```

قمنا باستعراض ملف "classes.dex" باستخدام HxD و استطعنا استخراج عدة أعلام من خلال خبرتنا المسبقة بأشكال الترميز و أنواعها. فقط علم واحد يحمل قيمة صحيحة مرشحة للفوز, أما الآخرين يحملون قيم غير مقبولة.

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00001A70 63 68 65 2F 68 74 74 70 2F 75 74 69 6C 2F 45 6E che/http/util/En
00001A80 74 69 74 79 55 74 69 6C 73 3B 00 06 4D 45 54 48 tityUtils;..METH
00001A90 4F 44 00 1D 4D 46 58 47 36 35 44 49 4D 56 5A 43 OD..MFXG65DIMVZC
00001AA0 41 5A 54 42 4E 52 5A 57 4B 49 44 47 4E 52 51 57 AZTBNRZWKIDGNRQW
00001AB0 4F 00 05 4D 4F 44 45 4C 00 09 50 41 52 41 4D 45 O..MODEL..PARAME
00001AC0 54 45 52 00 04 50 41 53 53 00 06 52 2E 6A 61 76 TER..PASS..R.jav

```

شكل ٨: "MFXG65DIMVZCAZTBNRZWKIDGNRQWO"

بعد تحويل العلم من Base32 إلى ASCII, حصلنا على النص الآتي:
"another false flag"

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00001B10 00 02 56 4A 00 02 56 4C 00 03 56 4C 49 00 04 56 ..VJ..VL..VLI..V
00001B20 4C 49 49 00 03 56 4C 4C 00 01 5A 00 02 5A 4C 00 LII..VLL..Z..ZL.
00001B30 10 5A 6D 46 73 63 32 56 6D 62 47 46 6E 49 44 74 .ZmFsc2VmbGFnIDt
00001B40 51 00 02 5B 42 00 13 5B 4C 6A 61 76 61 2F 6C 61 Q..[B..[Ljava/la
00001B50 6E 67 2F 53 74 72 69 6E 67 3B 00 01 61 00 08 61 ng/String;...a...

```

شكل ٩: "ZmFsc2VmbGFnIDtQ"

بعد تحويل العلم من Base64 إلى ASCII, حصلنا على النص الآتي:
" falseflag ;P"

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00001120 4E 75 6C 6C 50 6F 69 6E 74 65 72 3A 20 00 38 47 NullPointer:.8G
00001130 52 44 44 4B 4D 42 55 47 4D 32 44 49 4E 42 56 47 RDDKMBUGM2DINBVG
00001140 51 5A 44 49 4D 52 56 47 45 32 45 4D 4E 43 45 47 QZDIMRVGE2EMNCEG
00001150 51 33 54 4B 52 52 55 48 45 32 55 4D 4E 4A 58 47 Q3TKRRUHE2UMNJXG
00001160 52 44 44 49 52 49 3D 00 01 49 00 02 49 4C 00 03 RDDIRI=...I..IL..

```

شكل ١٠: "8GRDDKMBUGM2DINBVGQZDIMRVGE2EMNCEGQ3TKRRUHE2UMNJXGRDDIRI="

حاولنا تحويل العلم من Base32 إلى ASCII و لكن لم تتم عملية التحويل بنجاح. ثم أعدنا النظر إلى العلم و قررنا تجربة إزالة رقم ٨ الموجود في بداية المفتاح و المحاولة مرة أخرى. و حصلنا النتيجة التالية: "4F504344454242514F4D475F495F574F4E". بذت لنا كقيمة بترميز Hex لنص معين و ذلك لاحتوائها على قيم Hex متعددة بحجم 1 byte تبدأ بقيمة ٤ مثل 0x4F و 0x43 و 0x42 و هي عادة ترميز ASCII للأحرف الإنجليزية الكبيرة "capital letters". لذلك قمنا بتحويلها من Hex إلى ASCII ثم حصلنا النص الآتي: "OPCDEBBQOMG_I_WON"