الملف المرفق عبارة عن تطبيق أندرو يدapk ، بدون الحاجة لتشغيله يمكننا العمل على تحليله بتحليل ملفاته والكود المصدري.

باستخدام أداة apktool نقوم بعمل فك للبرنامج ونحصل على الناتج على شكل صور وملفات التطبيق.

بالمرور على الملفات يشد انتباهنا الآتي

- assets\ctf.png: ملف يفترض بأن يكون صورة وليس كذلك
- com\checkthisout\inset\C.smali : كود مصدري يتعامل مع هذا الملف

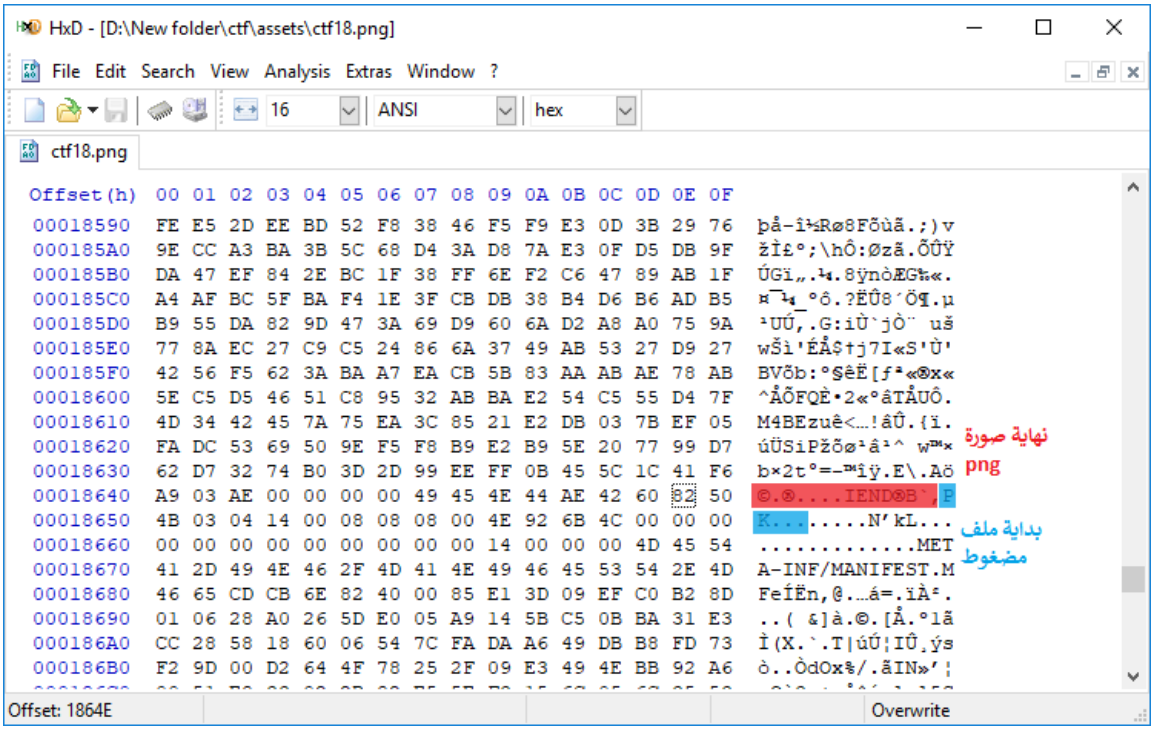
بتحليل الكود نجده يقوم بالآتي:

```
xor-int/lit8 v10, v10, 0x12
```

بالتالي فالملف الصورة يتم عمل xor بقيمة 18. -*تلميحة*-

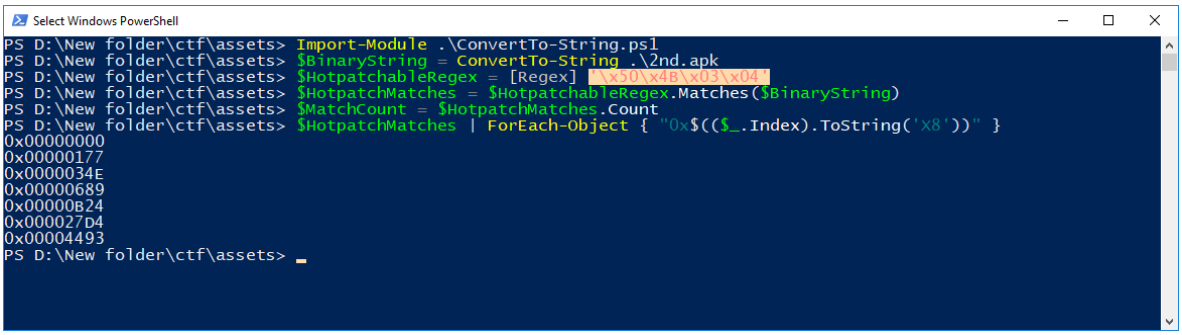
نقوم بعمل سكربت باورشل لعمل xor للملف وينتج لدينا ملف صورة، بمقارنته مع ملف app_logo.png الشبيه بالمحتوى نجده بحجم أكبر مما يدفع للشك.

بالنظر لمحتوى الملف من بيانات نجد آثارًا لملفات أخرى بداخله، بالتالي نركز عليه البحث أكثر.



بإمعان النظر، نجد أن بنهاية ملف png يوجد هناك قيمة PK التي تمثل جزء من ترويسة الملفات المضغوطة، أو تطبيقات أندرويد بحكم أنها ملفات مضغوطة بالنهاية.

المعروفة ونجد zip التي تمثل ترويسة ملفات '\x50\x4B\x03\x04' نستطيع عمل سكربت باورشل بسيط للبحث عن سلسلة بايتات قرابة ال7 ملفات داخل الملف المشبوه.



بمعرفة مواقع الترويسات نستطيع استخراجها من الملف وإصلاح المعطوب منها إن وجد، ونحصل على ملفات مضغوطة بداخلها تشابه ملفات تطبيقات أندرويد.

بجمع الملفات في ملف واحد والتعامل معه كapk ، وفكه باستخدام apktool مجددًا، نستطيع التعامل معه ك برنامج آخر، ونبدأ التحليل من جديد.

بالبحث بالملفات والمصادر، لا نجد الكثير من الأشياء الغريبة عدا بعض الفلاقز الخاطئة المفترض تجاهلها، ما يهم هنا:

- ملف strings.xml نجد قيمة SubmissionURL التي تمثل رابط مشاركة الفلاق.
- ملف com\android\setting\E.smali نجد نص كالتالي :

"GRDDKMBUGM2DINBVGQZDIMRVGE2EMNCEGQ3TKRRUHE2UMNJXGRDDIRI="

باستخدام أداة CyberChef نستطيع عمل decode باستخدام base32 ومن ثم تحويلها من هكس إلى نص عادي، ويظهر الفلاق.