تقرير التحليل الجنائي الرقمي — تحديات picoCTF 2022

اسم الطالب: محمد حيدر مرشد ناصر المعمري

الملخص التنفيذي

تُظهر الصور مسارًا متدرجًا لتطبيق التحليل الجنائي الرقمي في بيئة picoCTF 2022: يبدأ بقراءة جدول تقسيم القرص وتحديد حجم قسم Linux، ثم ينتقل إلى استعراض نظام الملفات واستخراج ملف العلم، وينتهي باستعادة مفتاح SSH من صورة قرص واستخدامه للدخول إلى آلة التحدي المصرّح بها وقراءة العلم.

تعتمد الحلول على أدوات **The Sleuth Kit**، وبالتحديد `mmls` و`fls` و`icat`، إلى جانب أوامر Linux مثل `gunzip` و`chmod` و`ssh`. وتكمن الفكرة المشتركة في أن صورة القرص ليست مجرد ملف يُفتح، بل حاوية تضم أقسامًا وأنظمة ملفات؛ لذلك يجب تحديد نقطة بداية القسم الصحيح قبل تحليل محتوياته.

نطاق التقرير: تحليل ما يظهر في لقطات الشاشة وشرح دلالته التقنية. لم تُرفق صور الأقراص نفسها، ولم يُعَد تنفيذ الأوامر أو الاتصال بخوادم التحديات. تُنقل النتائج من الصور، وتُفصل الملاحظات التفسيرية والممارسات الموصى بها عن الخطوات المثبتة بصريًا.

---

١. الخلفية التقنية ومنهجية التحليل

١.١ ما المقصود بصورة القرص؟

صورة القرص **Disk Image** تمثيل لمحتويات وسيط تخزين، قد يضم جدول تقسيم وأقسامًا متعددة وأنظمة ملفات. الامتداد `img.` لا يكفي وحده لتحديد بنية الصورة، بينما يشير الامتداد `gz.` إلى أن الملف مضغوط بصيغة gzip ويحتاج إلى فك الضغط قبل استخدام الأدوات في هذه الأمثلة.

تحتوي الصور على جدول تقسيم من نوع **DOS Partition Table**، وتعرض أداة `mmls` الإزاحات والأطوال بوحدات قطاعات حجم كل منها **512 بايت**. لا تعني تسمية DOS أن نظام التشغيل الموجود داخل الصورة هو DOS؛ إنها تسمية بنية جدول التقسيم.

١.٢ وظيفة الأدوات المستخدمة

المعلومة الأساسية	الوظيفة في التقرير	الأداة أو الأمر
يحوّل الملف المضغوط إلى صورة قابلة للتحليل	فك ضغط صورة القرص	`gunzip`
يحدّد البداية والنهاية والطول ونوع القسم	عرض تخطيط الأقسام	`mmls`
يربط الاسم بمعرّف بياناته الوصفية	سرد أسماء الملفات والمجلدات	`fls`
يستخدم معرّف الملف داخل القسم المحدّد	استخراج محتوى ملف من الصورة	`icat`
يبحث دون حساسية لحالة الأحرف	تصفية المخرجات نصيًا	`grep -i`
يرسل الإجابة في تحدي Intro	الاتصال بخدمة التحقق	`nc`
يسمح للمالك فقط بالقراءة والكتابة	تقييد صلاحيات المفتاح	`chmod 600`
يُستخدم داخل بيئة التحدي المصرّح بها	الدخول باستخدام المفتاح المستخرج	`ssh`

١.٣ كيفية قراءة مخرجات The Sleuth Kit

في `mmls`، يشير **Start** إلى أول قطاع للقسم، و**End** إلى آخر قطاع، و**Length** إلى عدد القطاعات. أما **Description** فيساعد على تمييز أقسام Linux والمساحة غير المخصّصة وقسم التبديل. صف **Meta / Primary Table** يصف بيانات جدول التقسيم؛ ليس قسم بيانات مستقلًا ينبغي إضافته إلى أحجام الأقسام.

في `fls`، تمثّل الرموز مثل `d/d` مجلدًا، و`r/r` ملفًا عاديًا، و`l/l` رابطًا رمزيًا. الرقم قبل النقطتين هو معرّف البيانات الوصفية الذي يمكن استخدامه مع `icat`؛ وفي أنظمة الملفات الظاهرة يُشار إليه عادةً باسم **inode**. علامات `+` توضّح عمق التداخل في السرد المتكرر، ولا تعني وحدها أن الملف محذوف. النجمة `*` تشير إلى إدخال محذوف، وقد تظهر عبارة `(realloc)` عند إعادة تخصيص البيانات الوصفية. أما `OrphanFiles$` فهو عرض افتراضي تقدّمه الأداة للعناصر اليتيمة، وليس بالضرورة مجلدًا أنشأه المستخدم.

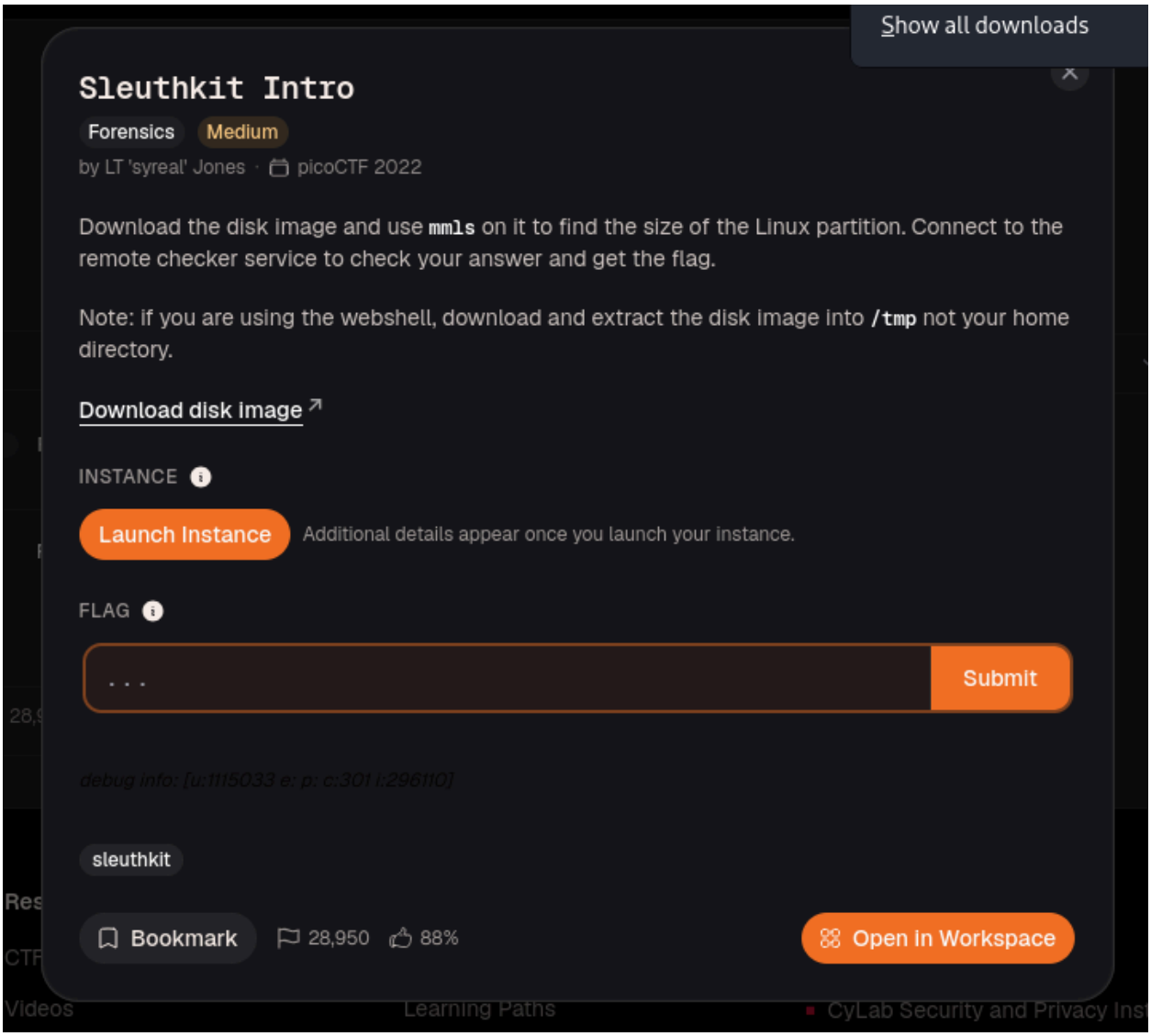
---

٢. التحدي الأول: Sleuthkit Intro

٢.١ وصف التحدي والهدف

تُظهر بطاقة التحدي أنه من فئة **Forensics** وبمستوى **Medium**، ضمن picoCTF 2022. المطلوب تنزيل صورة القرص، وتشغيل `mmls` لمعرفة حجم قسم Linux، ثم الاتصال بخدمة تحقق بعيدة للحصول على العلم.

تتضمن البطاقة رابط تنزيل الصورة وزر **Launch Instance** لتشغيل خدمة التحدي، ثم حقل **FLAG** لإدخال النتيجة. وتنبّه إلى استخدام `tmp/` عند العمل من webshell بدل المجلد المنزلي. هذه ملاحظة خاصة ببيئة webshell؛ أما اللقطات العملية فتُظهر العمل محليًا على سطح المكتب في Kali.

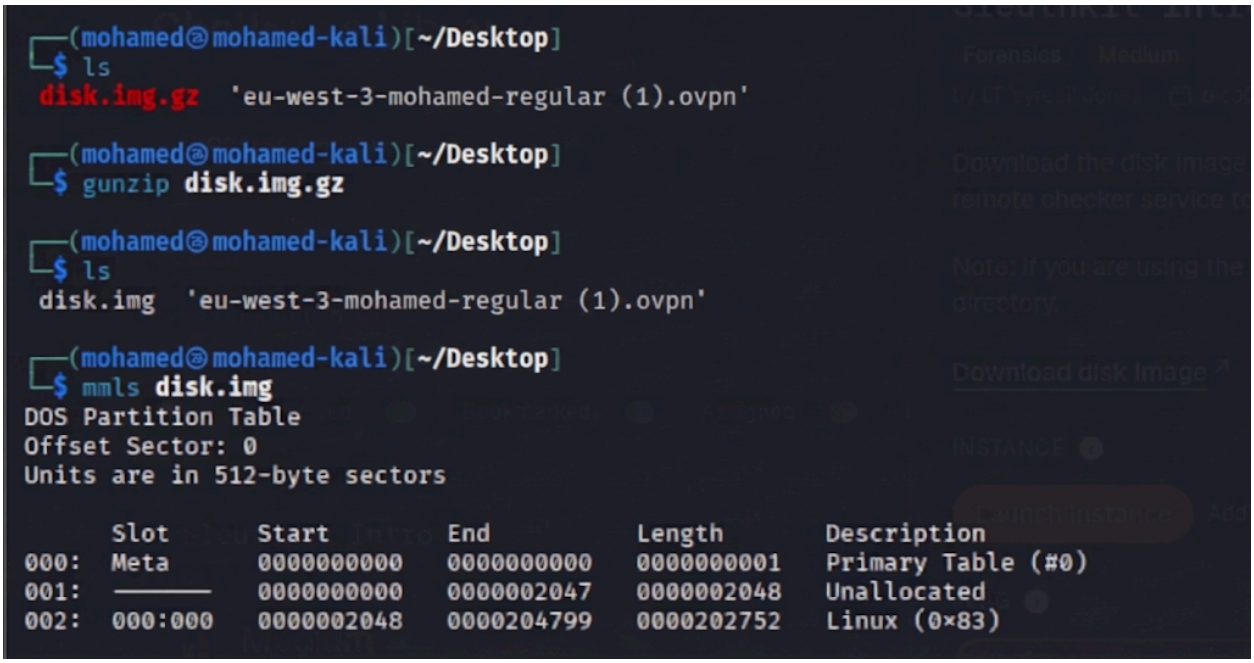


الشكل ١ — بطاقة Sleuthkit Intro ومتطلبات تحديد حجم قسم Linux.

٢.٢ تجهيز الصورة وقراءة جدول الأقسام

تبدأ الخطوات بعرض الملفات الموجودة، ثم فك ضغط `disk.img.gz`، ثم التأكد من ظهور `disk.img` وتشغيل أداة تحليل الأقسام:

```
ls
gunzip disk.img.gz
ls
mmls disk.img
```



الشكل ٢ — فك ضغط الصورة وعرض جدول الأقسام بواسطة mmls.

تحليل اللقطة:

- الملف `disk.img.gz` هو المدخل المضغوط، و`disk.img` هو الملف الناتج المستخدم للتحليل.
- ملف `ovpn.` الظاهر في قائمة الملفات لا يدخل في الأوامر المعروضة، ولا تثبت اللقطة الحاجة إلى VPN لحل هذا التحدي.
- جدول التقسيم يبدأ عند القطاع `0`، ووحدة القياس هي قطاعات بحجم `512` بايت.
- المساحة غير المخصّصة تمتد من القطاع `0` إلى `2047`.
- قسم **Linux (0x83)** يبدأ عند `2048` وينتهي عند `204799`، وطوله **202752 قطاعًا**.

يمكن التحقق من الاتساق الحسابي باستخدام العلاقة:

```
Length = End - Start + 1
202752 = 204799 - 2048 + 1
```

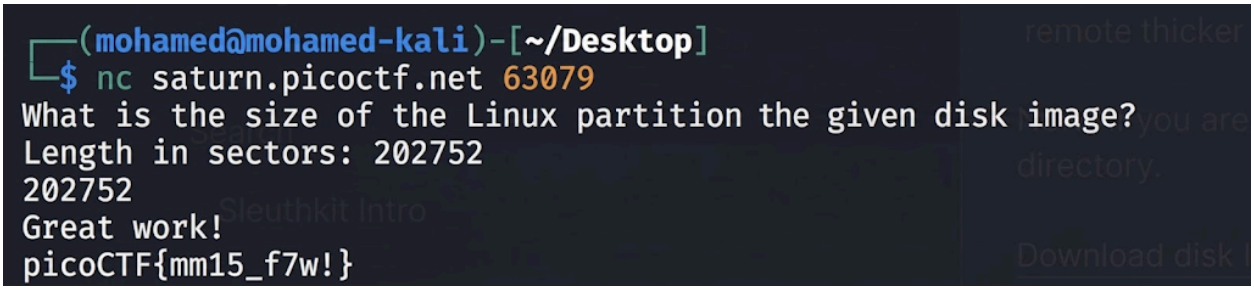
القيمة المطلوبة لاحقًا هي عدد القطاعات، لا عنوان البداية ولا النهاية. للتحويل إلى بايتات تُضرب قيمة الطول في حجم القطاع؛ لكن هذا التحويل ليس المطلوب في واجهة خدمة التحقق الظاهرة.

٢.٣ الاتصال بالخدمة وإرسال الإجابة

تُظهر اللقطة استخدام Netcat للاتصال بخدمة التحدي:

```
nc saturn.picoctf.net 63079
```

تعرض الخدمة السؤال **Length in sectors**، فتُرسل القيمة `202752`. ثم تظهر رسالة **Great work!** ويُعاد العلم، وهو دليل مباشر على قبول إجابة حجم القسم في هذه الجلسة.



الشكل ٣ — قبول الإجابة 202752 قطاعًا وإظهار علم Intro.

العلم الظاهر:

```
picoCTF{mm15_f7w!}
```

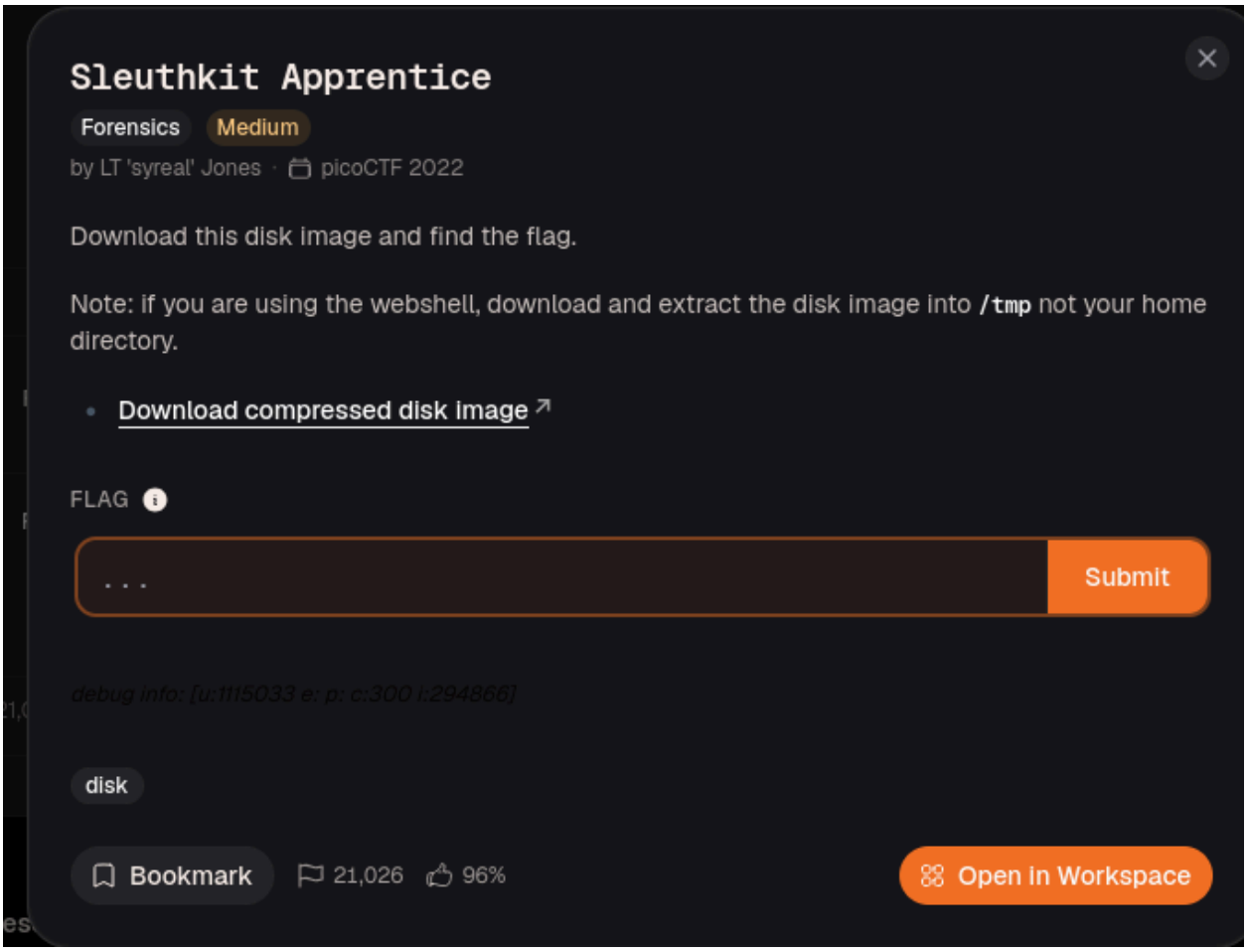
الاستنتاج: تم تحديد طول القسم الصحيح من جدول التقسيم، ثم تأكيده بواسطة خدمة التحدي. المنفذ `63079` خاص بالجلسة المصوّرة، وقد يتغير عند تشغيل نسخة جديدة.

---

٣. التحدي الثاني: Sleuthkit Apprentice

٣.١ وصف التحدي

يطلب هذا التحدي تنزيل صورة قرص والعثور على العلم داخلها. بخلاف Intro، لا يكفي تحديد حجم القسم؛ المطلوب الانتقال من بنية القرص إلى نظام الملفات، ثم استخراج محتوى ملف محدّد.



الشكل ٤ — بطاقة Sleuthkit Apprentice ورابط تنزيل صورة القرص.

تتكرر ملاحظة استخدام `tmp/` في webshell. تعرض البطاقة حقل العلم وزر **Submit**، لكنها لا تحدد اسم ملف العلم أو القسم الذي يحتويه؛ لذلك يُستدل عليهما من التحليل اللاحق.

٣.٢ فك الضغط وتحديد أقسام الصورة

```
gunzip disk.flag.img.gz
mmls disk.flag.img
```

```
┌──(mohamed@mohamed-kali)-[~/Desktop]
└─$ gunzip disk.flag.img.gz

┌──(mohamed@mohamed-kali)-[~/Desktop]
└─$ mmls disk.flag.img
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  ------    0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0000206847   0000204800   Linux (0x83)
003:  000:001   0000206848   0000360447   0000153600   Linux Swap / Solaris x86 (0x82)
004:  000:002   0000360448   0000614399   0000253952   Linux (0x83)
```

الشكل ٥ — تخطيط disk.flag.img: قسمان من نوع Linux وقسم تبديل.

تُظهر المخرجات الأقسام التالية:

القسم	البداية بالقطاعات	النهاية بالقطاعات	الطول بالقطاعات
Linux الأول	2048	206847	204800
Linux Swap / Solaris x86	206848	360447	153600
Linux الثاني	360448	614399	253952

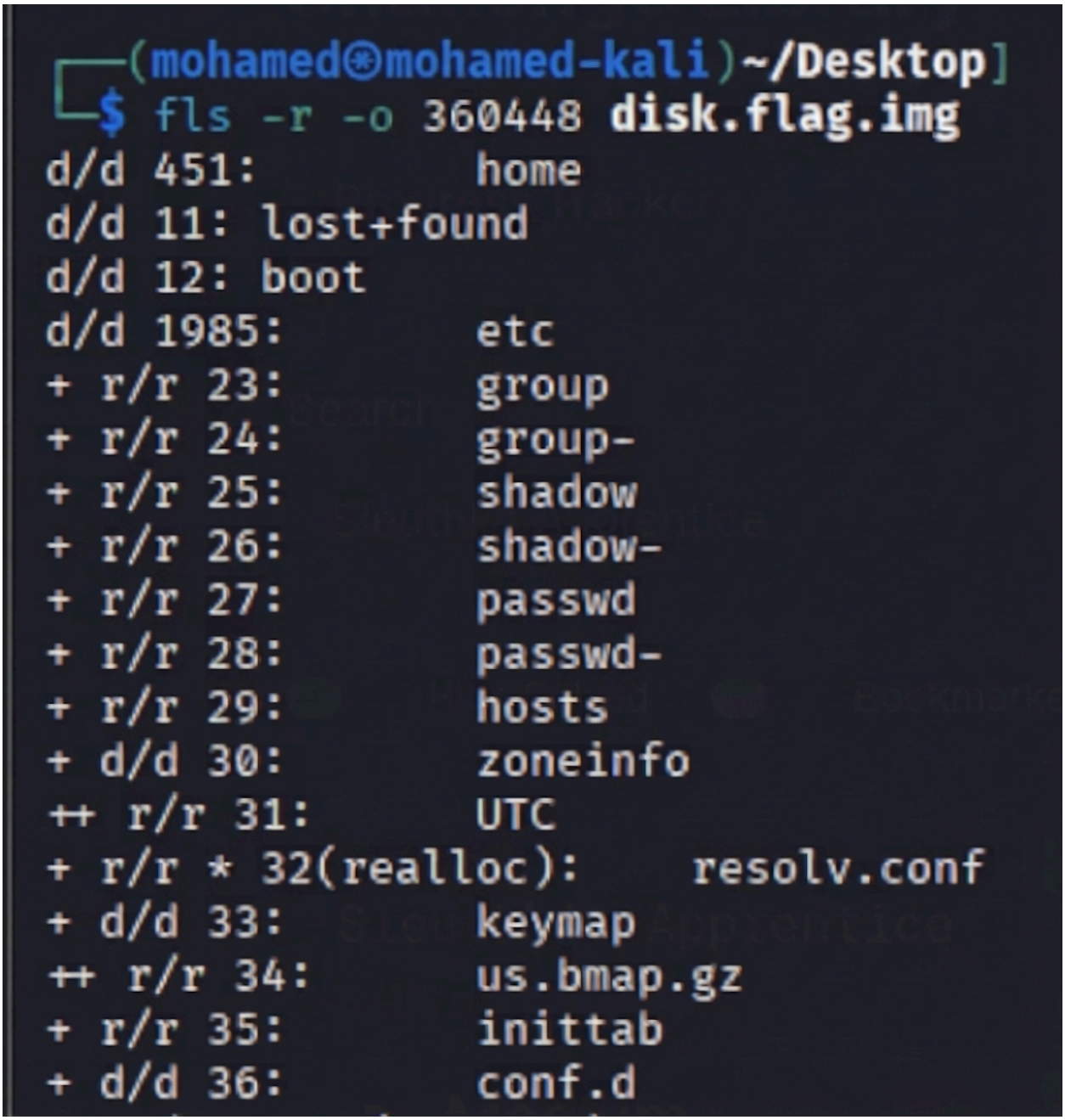
صف **Linux Swap / Solaris x86 (0x82)** يمثّل منطقة تبديل، وليس نظام ملفات عاديًا يُسرد بالطريقة نفسها في هذه الخطوات. أما القسم الثاني من نوع Linux فيبدأ عند **360448**، وهي الإزاحة المستخدمة في أوامر السرد والاستخراج التالية.

حدود الدليل: تعرض الصور اختيار هذا القسم وفحصه بنجاح، لكنها لا تُظهر جميع محاولات الاستكشاف السابقة أو سبب اختياره قبل تنفيذ `fls`. لا ينبغي وصف عملية استبعاد كل الأقسام على أنها خطوة موثّقة إذا لم تظهر في اللقطات.

٣.٣ استعراض نظام الملفات تكراريًا

```
fls -r -o 360448 disk.flag.img
```

- `-r`: سرد محتويات المجلدات تكراريًا.
- `-o 360448`: بدء تفسير نظام الملفات من القطاع الصحيح داخل صورة القرص.
- `disk.flag.img`: الصورة التي يُقرأ منها نظام الملفات.



الشكل ٦ — سرد نظام الملفات عند إزاحة القطاع 360448.

تُظهر اللقطة مجلدات مثل `home` و`boot` و`etc` و`lost+found`، إضافةً إلى ملفات إعدادات النظام مثل `group` و`passwd` و`shadow` و`hosts`. وجودها يدل على أن الأداة تقرأ بنية ملفات Linux مفهومة، وليس مجرد بيانات خام عشوائية.

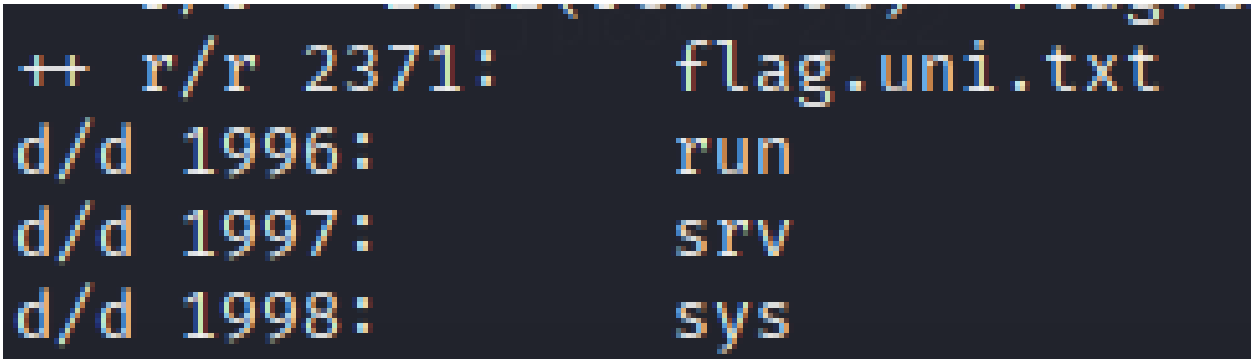
ملف `passwd` يضم عادةً معلومات الحسابات، بينما يُستخدم `shadow` عادةً لتخزين تجزئات كلمات المرور وبياناتها. وجود اسميهما في المخرجات لا يعني أن محتوياتهما استُخرجت في الخطوات

المصوّرة. كما تظهر علامة الحذف وإعادة التخصيص بجانب `resolv.conf`؛ وهي ملاحظة عن حالة ذلك الإدخال، وليست دليلًا على أنه يحتوي العلم.

٣.٤ تحديد ملف العلم ومعرّفه

تُظهر لقطة مكملة للمخرجات ملفًا باسم `flag.uni.txt` ومعرّفًا هو **2371**:

```
++ r/r 2371: flag.uni.txt
```



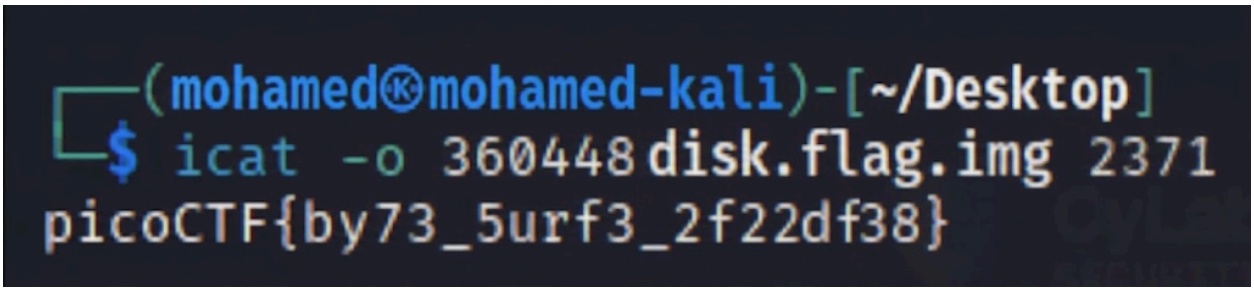
الشكل ٧ — تحديد flag.uni.txt بالمعرّف 2371.

الرمز `r/r` يبيّن أنه ملف عادي، بينما توضح علامتا `+` موقعه في عمق السرد. هذه القصاصة لا تعرض سلسلة المجلدات السابقة كاملة، لذلك لا تكفي وحدها لتوثيق مسار مطلق للملف. المعلومة المؤكدة والضرورية للاستخراج هي المعرّف **2371** داخل القسم المحدّد.

٣.٥ استخراج المحتوى باستخدام icat

```
icat -o 360448 disk.flag.img 2371
```

تستخدم `icat` الإزاحة نفسها التي نجح معها `fls`، ثم تقرأ بيانات الملف المشار إليه بالمعرّف `2371`. بذلك يتحول العثور على اسم ملف إلى الحصول على محتواه الفعلي دون الحاجة إلى تشغيل نظام التشغيل الموجود داخل الصورة.



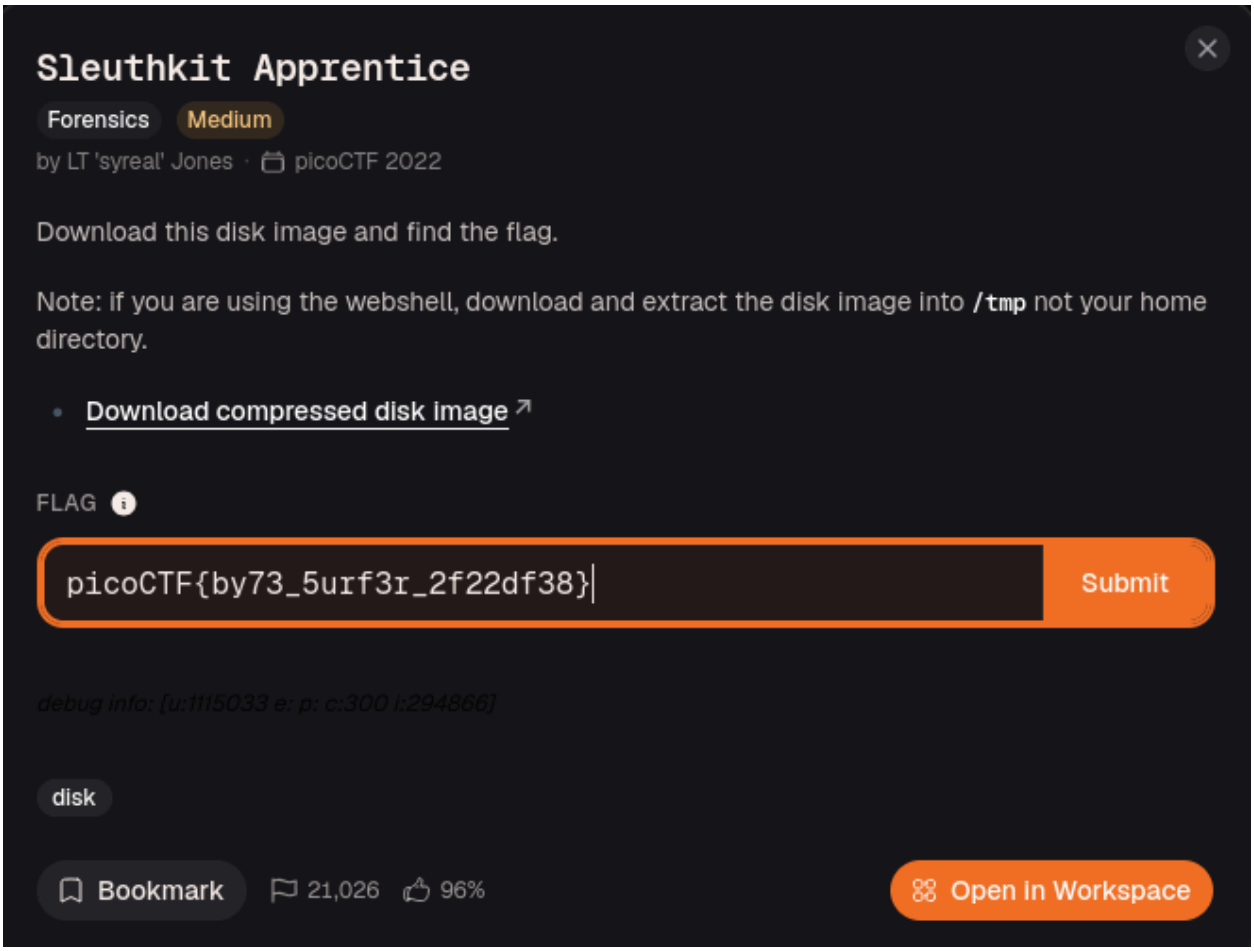
الشكل ٨ — استخراج محتوى ملف العلم باستخدام icat.

ملاحظة تحريرية على الأمر: تبدو المسافة بين `360448` و`disk.flag.img` غير واضحة في اللقطة؛ الصيغة الصحيحة القابلة للتنفيذ مكتوبة أعلاه بمسافة فاصلة بين الإزاحة واسم الصورة.

العلم، كما تؤكده أيضًا لقطة الإدخال التالية:

```
picoCTF{by73_5urf3r_2f22df38}
```

٣.٦ إدخال العلم في المنصة



الشكل ٩ — إدخال علم Sleuthkit Apprentice في المنصة.

تُظهر الصورة مطابقة قيمة الحقل للعلم المستخرج. لكنها لا تعرض رسالة قبول بعد الضغط على **Submit**؛ لذا فالاستنتاج الموثّق هو **استخراج العلم وإدخاله في الحقل**، لا إثبات قبول المنصة من هذه اللقطة وحدها.

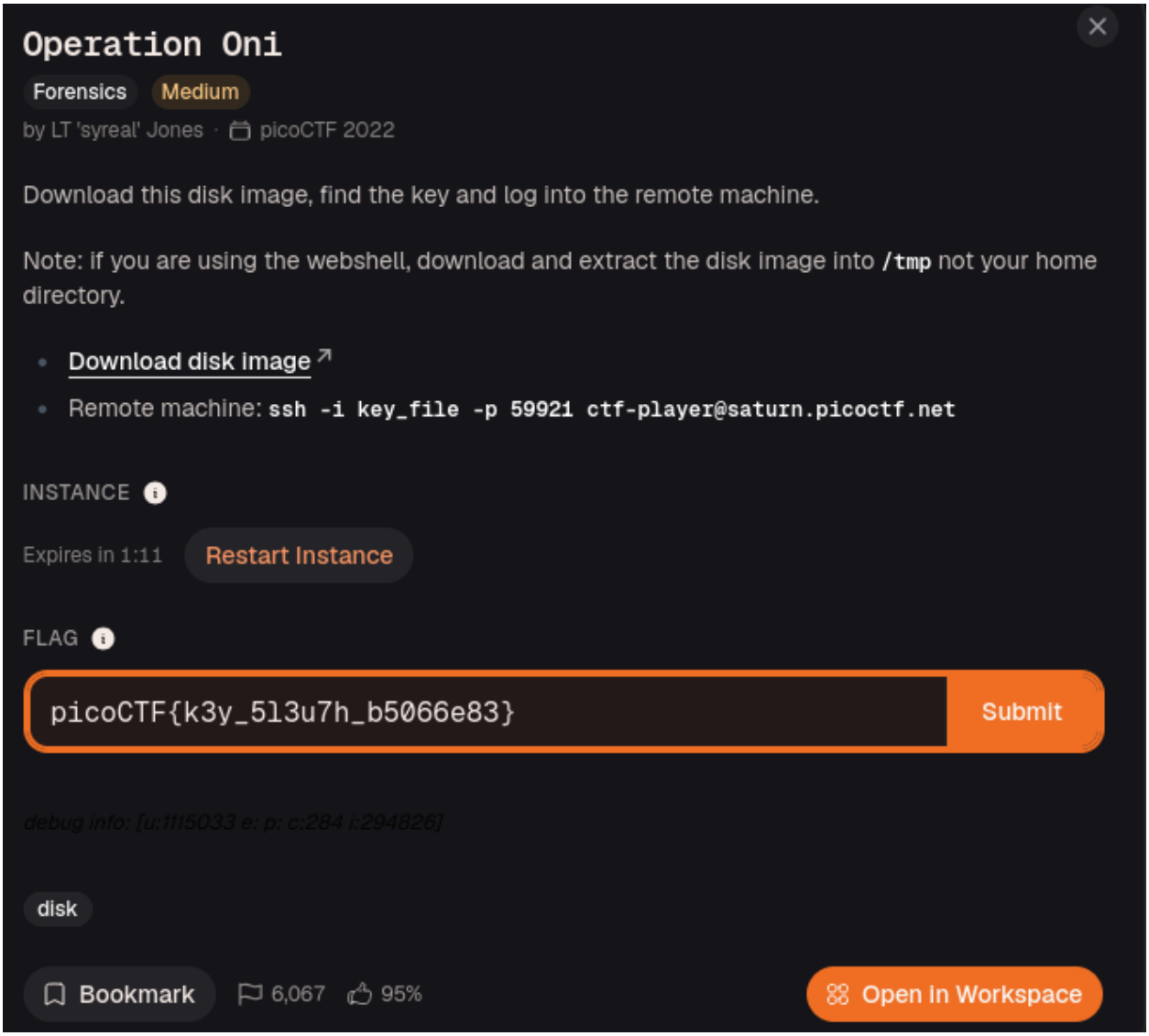
الخلاصة الفنية: يتكون الحل من سلسلة أدلة واضحة: تحديد بداية القسم `360448` ← سرد الملفات ← تحديد `flag.uni.txt` بالمعرّف `2371` ← استخراج محتواه باستخدام `icat`.

---

٤. التحدي الثالث: Operation Oni

٤.١ الهدف وسياق المهمة

يطلب التحدي تنزيل صورة قرص، والعثور على مفتاح، ثم استخدامه للدخول إلى الآلة البعيدة. هنا لا تكون النتيجة النهائية داخل صورة القرص بالضرورة؛ فالصورة تحتوي وسيلة المصادقة التي تقود إلى العلم على خادم التحدي.



الشكل ١٠ — بطاقة Operation Oni: وصف المهمة وأمر الاتصال والعلم المدخل.

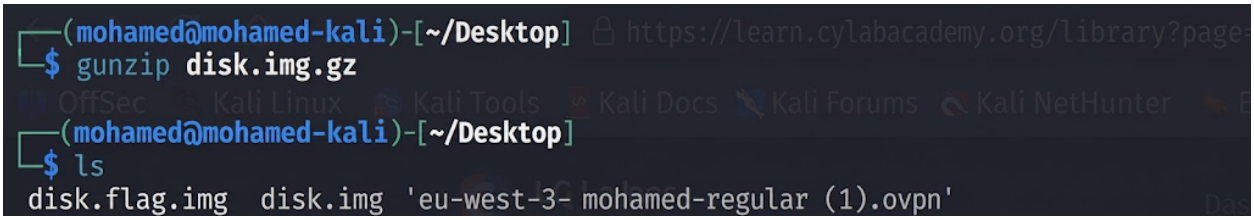
ملاحظة على ترتيب الأدلة: هذه الصورة كانت آخر لقطة في الصفحة الأصلية، وتعرض حالة متقدمة تتضمن العلم المدخل بالفعل. وُضعت هنا لتقديم اسم التحدي ومتطلباته وأمر الاتصال؛ أما إثبات استخراج المفتاح وقراءة العلم فيأتي في اللقطات التالية.

تُظهر البطاقة الأمر `ssh -i key_file -p 59921 ctf-player@saturn.picoctf.net`، ومؤقتًا للجلسة وزر **Restart Instance**. وهذا يفسر ارتباط بيانات الاتصال بنسخة مؤقتة قد تنتهي أو تتغير.

٤.٢ تجهيز صورة القرص

```
gunzip disk.img.gz
```

```
ls
```

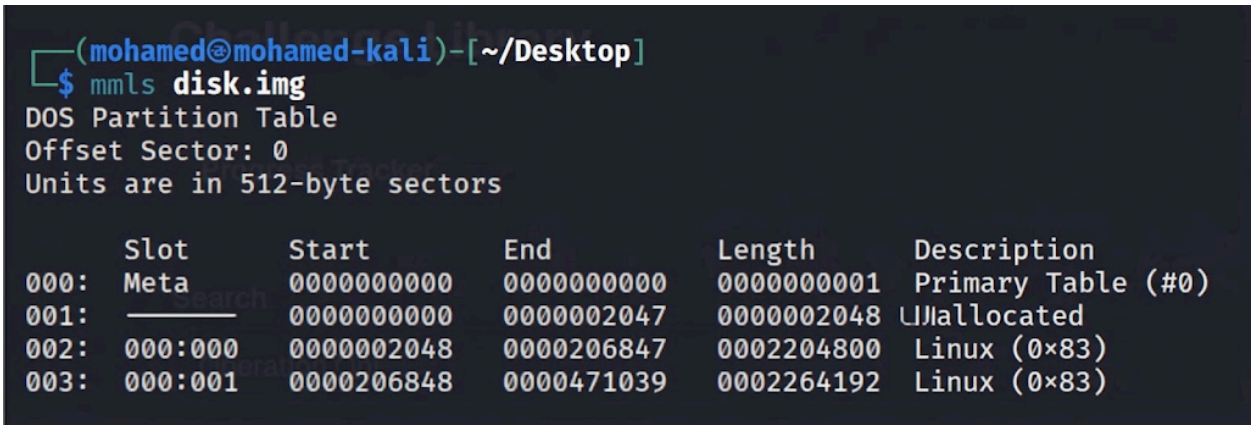


الشكل ١١ — فك ضغط صورة Operation Oni وظهور disk.img.

تحتوي قائمة الملفات أيضًا على `disk.flag.img` من الخطوات السابقة وملف `ovpn.`. لكن الأوامر التالية تستخدم **disk.img** تحديدًا. تطابق اسم الصورة هنا مع اسم صورة Intro لا يعني تطابق محتواهما؛ لكل تحدٍّ ملفاته، ويجب التحقق من مصدر الصورة لا من الاسم وحده.

٤.٣ تحديد القسم المناسب

```
mmls disk.img
```



الشكل ١٢ — جدول الأقسام وبداية القسم الثاني عند 206848.

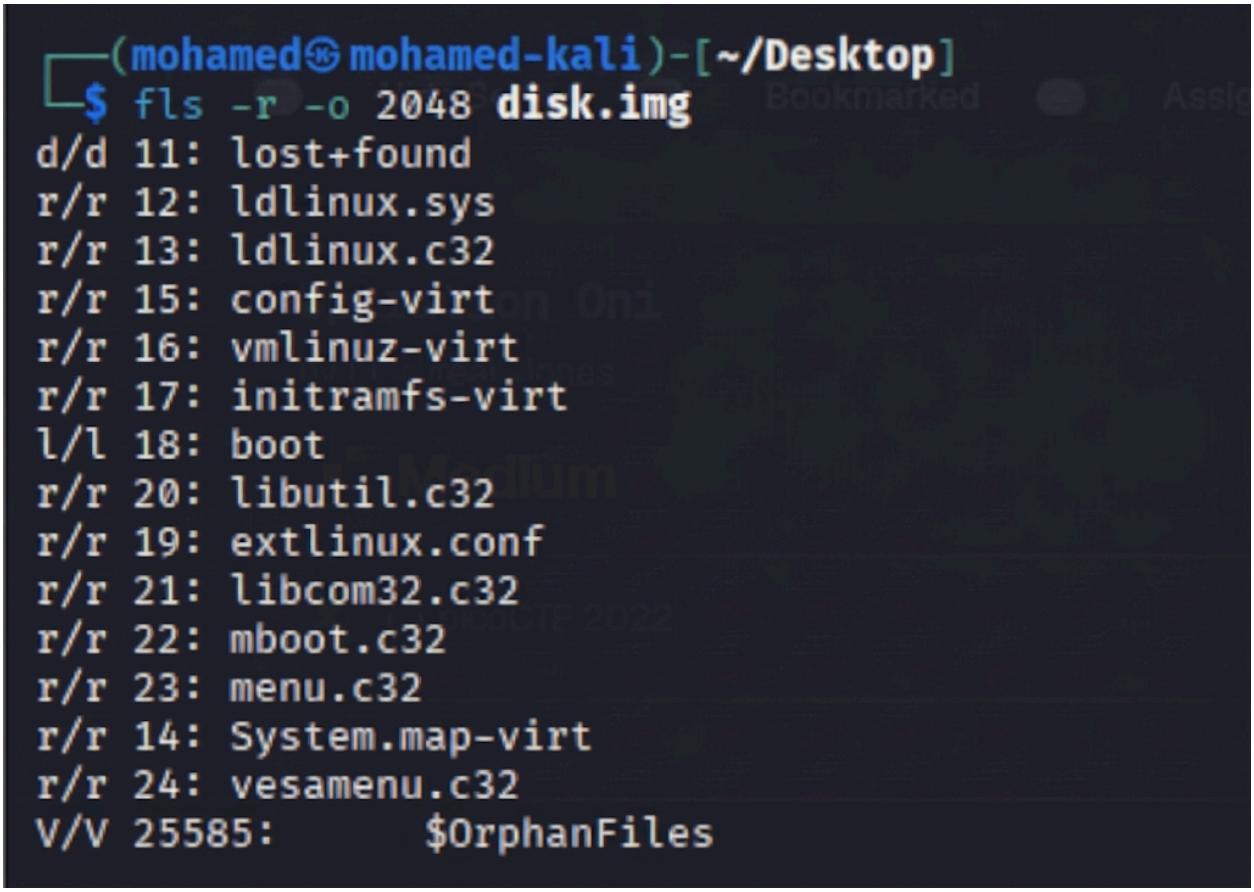
تُظهر الصورة قسمين من نوع Linux:

القسم	البداية بالقطاعات	النهاية بالقطاعات	الطول بالقطاعات
Linux الأول	2048	206847	204800
Linux الثاني	206848	471039	264192

الإزاحة **206848** هنا تختلف عن إزاحة Apprentice البالغة **360448**. نقل الإزاحات بين صور مختلفة دون قراءة جدول التقسيم قد يوجّه الأداة إلى موضع خاطئ.

٤.٤ فحص القسم الأول: محتويات مرتبطة بالإقلاع

```
fls -r -o 2048 disk.img
```



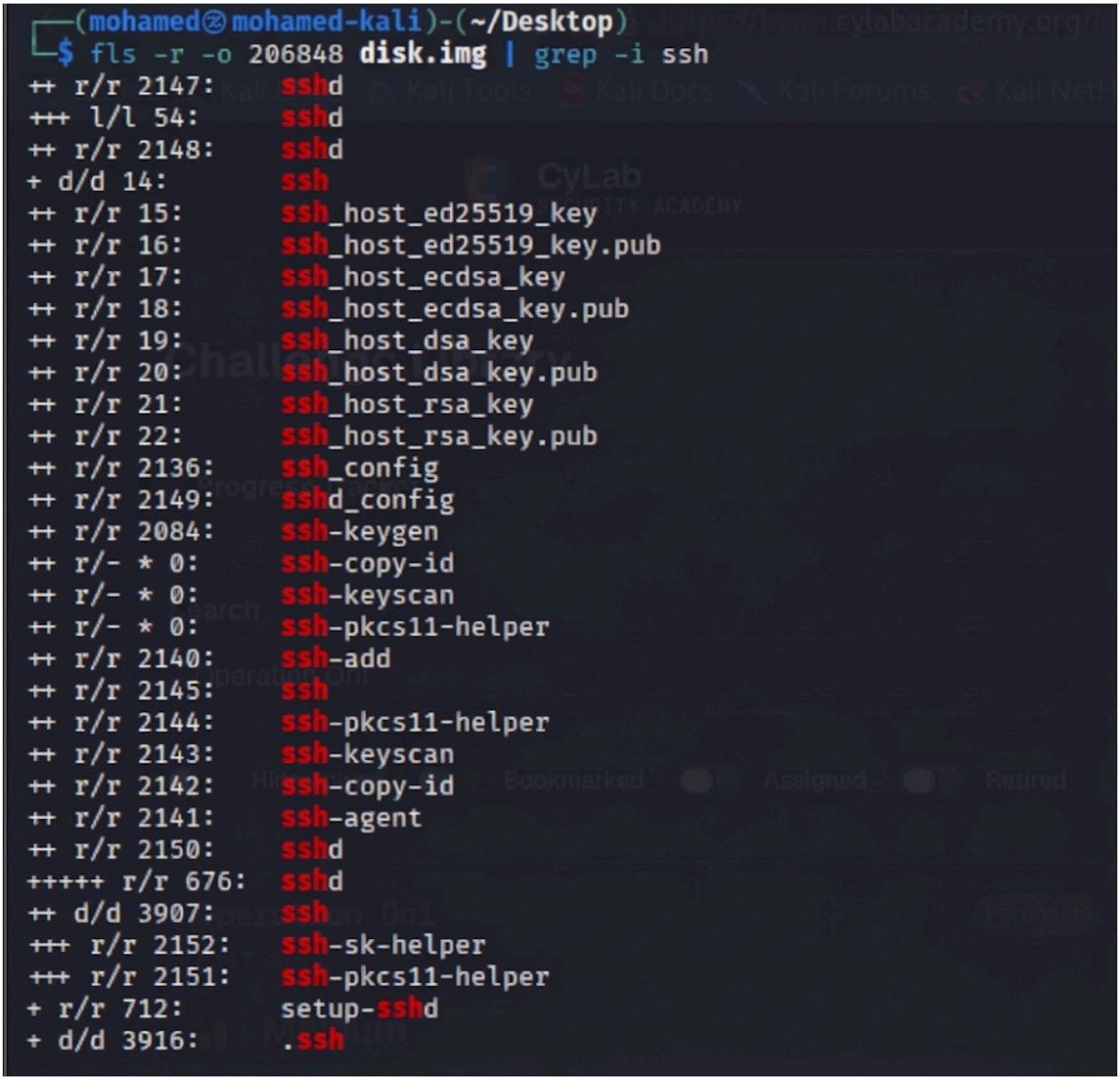
الشكل ١٣ — محتويات القسم الأول المرتبطة بالإقلاع.

تظهر أسماء مثل `ldlinux.sys` و`ldlinux.c32` و`vmlinuz-virt` و`initramfs-virt` و`extlinux.conf` و`System.map-virt` ووحدات `c32.`. هذه الأسماء تشير إلى محتويات مرتبطة بالإقلاع: نواة النظام، وبيئة التهيئة الأولية، وإعدادات ومكوّنات محمّل الإقلاع. ويظهر `boot` كرابط رمزي من النوع `l/l`.

لا تعرض هذه اللقطة مفتاح المستخدم المطلوب. لذلك ينتقل مسار الحل المصوّر إلى القسم الثاني. وهذا تفسير مبني على المخرجات الظاهرة، وليس حكمًا بأن القسم الأول خالٍ من كل دليل محتمل.

٤.٥ البحث عن العناصر المرتبطة بـ SSH

```
fls -r -o 206848 disk.img | grep -i ssh
```



الشكل ١٤ — البحث عن ملفات SSH وتحديد المجلد .ssh.

يجمع الأمر بين عمليتين: سرد نظام الملفات عند الإزاحة `206848`، ثم تمرير المخرجات بواسطة `|` إلى `grep` لتصفية السطور التي تحتوي `ssh` دون حساسية لحالة الأحرف.

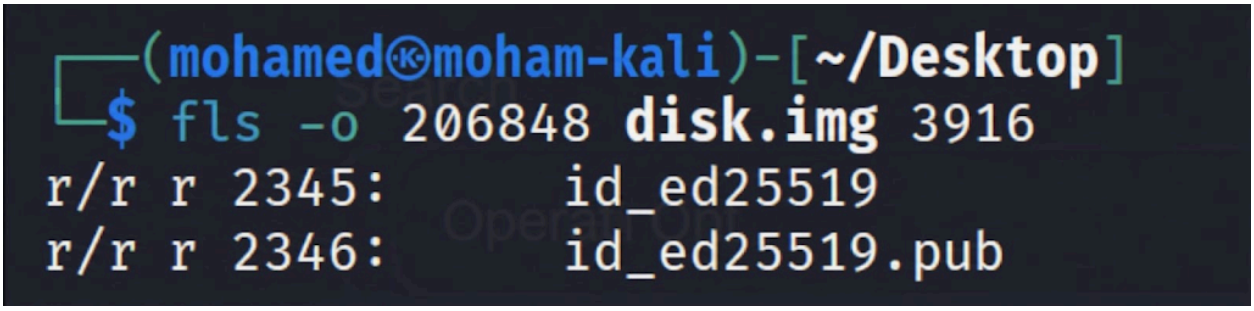
تظهر ملفات الخدمة `sshd`، وإعدادات `ssh_config` و`sshd_config`، وأدوات مثل `ssh-keygen` و`ssh-agent`، ومفاتيح المضيف `*_ssh_host`. وتظهر كذلك نتيجة مهمة هي المجلد `ssh.` ذي المعرّف **3916**.

لماذا التركيز على `ssh.`؟ لأن مفاتيح المستخدم توجد عادةً داخله. أما ملفات `*_ssh_host` فتمثل هوية خادم SSH غالبًا، وليست تلقائيًا مفاتيح دخول حساب المستخدم. يجب التمييز بين مفاتيح المضيف ومفتاح المستخدم بدل استخراج أول ملف يحتوي اسمه على كلمة `key`.

حدود التصفية: يطابق `grep` النص المعروض فقط، ولا يبحث داخل محتوى الملفات. كما قد يحذف سطور المجلدات المحيطة التي لا تطابق كلمة البحث؛ لذلك لا يثبت هذا السرد المصفّى وحده المسار الكامل للمجلد `ssh.`.

٤.٦ استعراض مجلد ssh.

```
fls -o 206848 disk.img 3916
```



الشكل ١٥ — المفتاح الخاص id_ed25519 والمفتاح العام المقابل.

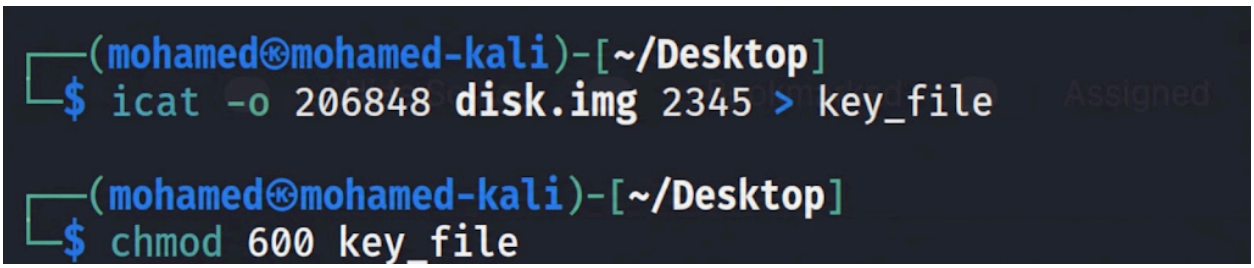
يطلب الرقم `3916` من `fls` عرض محتويات المجلد المحدّد بدل البدء بسرد عام. وتظهر نتيجتان:

- **المعرّف 2345:** `id_ed25519` — اسم المفتاح الخاص الذي سيُستخدم للمصادقة.
- **المعرّف 2346:** `id_ed25519.pub` — المفتاح العام المقابل، كما يدل الامتداد `.pub`.

تعرض اللقطة حرف `r` إضافيًا قبل المعرّفين. لا تكفي الصورة وحدها لتفسير هذه العلامة أو الجزم بحالة حذف الملف واستعادته، ولا ينبغي الخلط بينها وبين زوج الرموز `r/r` الذي يصف نوع الملف. أما نجاح الاستخدام اللاحق فهو الدليل العملي على قابلية المفتاح المستخرج للعمل في جلسة التحدي.

٤.٧ استخراج المفتاح الخاص وضبط صلاحياته

```
icat -o 206848 disk.img 2345 > key_file
chmod 600 key_file
```



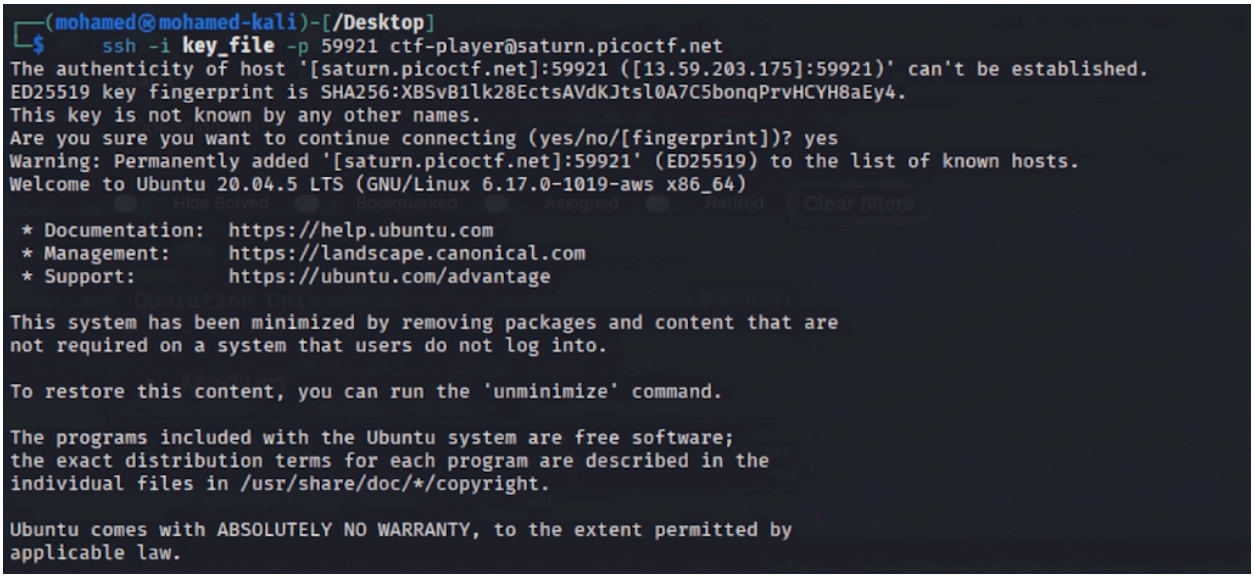
الشكل ١٦ — استخراج المفتاح الخاص وتقييد صلاحياته إلى 600.

يقرأ `icat` محتوى الملف ذي المعرّف **2345** من القسم الصحيح. ثم يعيد الرمز `>` توجيه المحتوى إلى ملف محلي باسم `key_file` بدل عرضه في الطرفية. إذا كان ملف بهذا الاسم موجودًا، فإن إعادة التوجيه بهذه الصيغة تستبدل محتواه.

يضبط `chmod 600` الصلاحيات على قراءة وكتابة للمالك فقط، دون صلاحيات للمجموعة أو للآخرين. ويُكتب ذلك رمزيًا `-------rw`. هذه خطوة مهمة لأن عميل SSH قد يرفض استخدام مفتاح خاص متاح للقراءة على نطاق أوسع.

٤.٨ المصادقة باستخدام المفتاح المستخرج

```
ssh -i key_file -p 59921 ctf-player@saturn.picoctf.net
```



الشكل ١٧ — نجاح اتصال SSH وظهور ترحيب النظام البعيد.

تفصيل عناصر الأمر:

- `-i key_file`: تحديد ملف المفتاح الخاص المحلي.
- `-p 59921`: تحديد منفذ خدمة SSH في الجلسة المصوّرة.
- `ctf-player`: اسم الحساب على خادم التحدي.
- `saturn.picoctf.net`: اسم المضيف البعيد.

تعرض الطرفية تحذير الاتصال الأول بالمضيف، وبصمة مفتاح **ED25519**، ثم موافقة `yes` وإضافة هوية المضيف إلى `known_hosts`. هذه البصمة تخص **مفتاح الخادم**، وليست المفتاح الخاص الذي استُخرج للمستخدم.

بعد ذلك تظهر رسالة **Welcome to Ubuntu 20.04.5 LTS** ومعلومات النظام والدعم ورسالة أن النظام مصغّر. هذه رسائل ترحيب تشغيلية، وليست أخطاء في عملية المصادقة. وتثبت رسالة الترحيب أن الاتصال تجاوز المصادقة إلى جلسة على النظام البعيد؛ وتؤكد اللقطة التالية إمكانية تنفيذ أوامر داخلها.

ممارسة أمنية موصى بها: في بيئة حقيقية يجب التحقق من بصمة الخادم عبر قناة موثوقة قبل الموافقة. قبول `yes` في اللقطة يوثّق ما حدث، ولا يُعد بديلًا عن التحقق المستقل من هوية المضيف.

٤.٩ قراءة العلم من الآلة البعيدة

```
ls
cat flag.txt
```



الشكل ١٨ — قراءة flag.txt ثم إغلاق الاتصال من الطرف البعيد.

يُظهر `ls` وجود `flag.txt` في مجلد الجلسة الحالي. ثم يعرض `cat` محتواه:

```
picoCTF{k3y_5l3u7h_b5066e83}
```

تظهر رسالة **Connection to saturn.picoctf.net closed by remote host** بعد قراءة العلم. وهي تعني أن الطرف البعيد أغلق الاتصال، ولا تثبت وحدها السبب؛ فقد تنتهي جلسات التحديات المؤقتة لأسباب تشغيلية، لكن لا يمكن الجزم بذلك من الصورة. وبما أن العلم ظهر قبل الإغلاق، فقد تحققت خطوة القراءة بالفعل.

تُظهر بطاقة التحدي في الشكل ١٠ إدخال العلم نفسه في حقل **FLAG**. كما في Apprentice، لا تتضمن اللقطات رسالة قبول بعد الإرسال، لذلك لا يوثّق التقرير قبولًا لم يظهر في الصور.

الخلاصة الفنية: صورة القرص ← القسم عند `206848` ← المجلد `ssh.` بالمعرّف `3916` ← المفتاح الخاص بالمعرّف `2345` ← ملف `key_file` بصلاحيات `600` ← مصادقة SSH ← قراءة `flag.txt`.

---

٥. النتائج الموثّقة وحدود الإثبات

Sleuthkit Intro

النتيجة: تحديد طول قسم Linux بقيمة `202752` قطاعًا. **الإثبات:** خدمة التحقق عرضت رسالة `!Great work` والعلم `picoCTF{mm15_f7w!}`.

Sleuthkit Apprentice

النتيجة: استخراج `flag.uni.txt` من القسم عند `360448` باستخدام المعرّف `2371`. **الإثبات:** عرض المحتوى في الطرفية، ثم إدخال `picoCTF{by73_5urf3r_2f22df38}` في المنصة. لا تظهر رسالة قبول نهائية.

Operation Oni

النتيجة: استخراج مفتاح المستخدم من القسم عند `206848` واستخدامه للوصول إلى آلة التحدي وقراءة `flag.txt`. **الإثبات:** رسالة ترحيب SSH، وتنفيذ `ls` و `cat`، وظهور `picoCTF{k3y_5l3u7h_b5066e83}`، ثم إدخاله في البطاقة. لا تظهر رسالة قبول نهائية.

---

٦. الدروس المستفادة والممارسات المهنية

٦.١ بناء سلسلة أدلة قابلة للتتبّع

ينبغي ربط كل نتيجة بالخطوة التي أنتجتها: جدول التقسيم يثبت الإزاحة، وسرد الملفات يثبت الاسم والمعرّف، واستخراج المحتوى يثبت النتيجة النصية. أما نجاح الاتصال أو قبول خدمة التحقق فيضيف طبقة تأكيد لاحقة. يمنع هذا الأسلوب القفز من لقطة جزئية إلى استنتاج لا تدعمه الأدلة.

٦.٢ عدم الخلط بين وحدات القياس والمعرّفات

طول القسم ليس عنوان بدايته، والإزاحة بالقطاعات ليست إزاحة بالبايتات، ومعرّف الملف ليس مساره النصي. تكرار هذه الفروق هو ما يجعل الأوامر قابلة للفهم وإعادة التنفيذ بدل حفظها بصورة آلية.

٦.٣ التعامل مع صور الأقراص كمصادر حساسة

يمكن أن تحتوي الصور على بيانات مستخدمين وملفات إعدادات ومواد اعتماد حتى إذا لم يكن نظام التشغيل يعمل. توضح Operation Oni أن التحليل دون تشغيل النظام قد يكشف مفتاحًا خاصًا صالحًا للمصادقة داخل سياق التحدي.

٦.٤ الحفاظ على سلامة الدليل عند التطبيق الواقعي

الخطوات التالية **توصيات مهنية وليست إجراءات ظاهرة في الصور**:

- الاحتفاظ بالأصل والعمل على نسخة، مع توثيق مصدر الملف وتاريخ الحصول عليه.
- حساب تجزئة مثل SHA-256 للتحقق من سلامة الدليل؛ لم تُعرض أي قيمة تجزئة لصورة القرص في اللقطات.
- استخدام `gunzip -k` عند الحاجة إلى إبقاء النسخة المضغوطة، لأن `gunzip` بصيغته المعتادة يستبدلها بالملف المفكوك.
- تجنّب تركيب نظام الملفات في وضع الكتابة، وتوثيق الأوامر والإصدارات والنتائج.
- حماية المفاتيح الخاصة المستخرجة وعدم نشرها، والتحقق من بصمة المضيف قبل الاتصال.
- الحصول على تفويض واضح لأي اتصال أو استخدام لمواد اعتماد، وحصره ضمن نطاق العمل المسموح.

الخاتمة

تُبرز التحديات الثلاثة كيف يتطور التحليل الجنائي الرقمي من **فهم تخطيط القرص** إلى **استخراج دليل من نظام الملفات** ثم **استخدام ذلك الدليل ضمن بيئة تدريبية مصرّح بها**. ويُعد اختيار الإزاحة الصحيحة وربط الملف بمعرّفه وفصل النتيجة الظاهرة عن الاستنتاج التفسيري أساسًا لكتابة تقرير تقني دقيق.

أُعيد ترتيب الصور الثماني عشرة ضمن سياقها التحليلي، مع الحفاظ على كل لقطة وإيضاح الأوامر والنتائج وحدود ما تثبته. يستند التقرير إلى الصور المرفقة وحدها؛ ولا يفترض تنفيذًا جديدًا أو قبولًا من المنصة لم يظهر في الأدلة.