Verdächtige Mailanhänge risikolos untersuchen und entschärfen

Erfolgreicher Exorzismus

Wie Sie verdächtige Mailanhänge risikolos untersuchen und entschärfen

Mailanhänge zu öffnen, ist ein riskantes Unterfangen – aber oft unumgänglich. Wir stellen Tools vor, mit denen Sie Anhänge in risikofreie Kopien verwandeln und eingehend untersuchen können, bevor Sie sie öffnen.

Von Sylvester Tremmel

Mailanhängen dürfen Sie nicht vertrauen. Doch egal wie vorsichtig Sie Ihren Posteingang auf Phishing-Attacken untersuchen und wie misstrauisch Sie E-Mails begegnen: Früher oder später taucht ein Anhang auf, dessen Absichten unklar sind und den Sie nicht ignorieren können, weil der Inhalt verspricht, wichtig zu sein.

Also müssen Sie irgendwie das Risiko verringern, das von dem Anhang ausgeht, bevor Sie ihn öffnen. Dazu haben Sie eine Reihe von Handlungsoptionen; die einfachste vorweg: Sehen sie nach, ob ein Online-Virenscanner wie virustotal.com den Anhang kennt. Allerdings nicht, indem Sie dort einfach die Datei hochladen, sonst haben Sie allzu leicht ein Datenschutzproblem am Hals (siehe dazu den Artikel auf S. 18). Berechnen Sie

stattdessen lokal einen eindeutigen Hash der Datei und geben Sie diesen in die Suche von VirusTotal ein. Aus dem Hash lassen sich keine Daten rekonstruieren, aber falls es sich um eine bereits bekannte Datei handelt, bekommen Sie so eine Einschätzung des Dienstes. Viren-Dokumente werden in der Regel breit gestreut, mit etwas Glück liegt daher zu einer verseuchten Datei bereits ein Report vor.

Auf VirusTotal muss man nicht unbedingt eigene Dateien hochladen. Man kann auch per Hash nach bereits bekannten Dateien suchen.
Einen passenden Hash berechnen Sie am schnellsten auf der Kommandozeile, unter Windows mit dem PowerShell-Befehl Get-FileHash DATEI, unter Linux per sha256sum DATEI und unter macOS mit shasum -a 256 DATEI. Es gibt aber auch diverse Tools mit grafischer Oberfläche, die Hashes berechnen können; VirusTotal findet Hashwerte der Verfahren MD5, SHA-1 und SHA-256. (Nutzen Sie am besten das letzte, es gilt als uneingeschränkt sicher.)

Wenn gleich mehrere namhafte Scanner bei VirusTotal anschlagen, sollten Sie den Anhang direkt in den Orkus schicken. Falls der Onlinedienst die Datei nicht kennt oder darin nichts findet, dann ist das nur ein erster Hinweis, aber noch keine Unbedenklichkeitserklärung, und Sie sollten

weiterforschen.

Ab in die Quarantäne

Zum Beispiel, indem Sie eine von Ihrem Arbeitsrechner isolierte Umgebung nutzen, aus der Malware nicht ausbrechen kann. Dafür eignet sich unter anderem eine virtuelle Maschine (VM). Wenn man darin ein bösartiges Dokument öffnet, geht höchstens diese VM zugrunde. Zwar gibt es auch in VM-Software Lücken, aber das Risiko, dass eine Malware aus der Virtualisierung herauskommt, ist sehr, sehr gering.

VMs sind gut, um gelegentlich eine Datei zu analysieren. Dann bootet man darin am besten ein frisches Spezialsystem wie Kali Linux oder Parrot Security [1, 2] und löscht nach der Analyse die ganze VM. Sie können virtuelle Maschinen auch zur Absicherung der täglichen Arbeit nutzen, zum Beispiel, indem Sie darin ein wartungsarmes Linux wie Debian [3] installieren und damit Ihre Mails abrufen. Das ist eine gute Methode, aber wenn man täglich so arbeitet, stößt man schnell an die Grenzen, die durch die Isolierung entstehen. Wer dann keine eiserne Disziplin zeigt, bohrt über kurz oder lang Löcher in die Isolation, um leichter Dateien in die VM hinein und aus ihr heraus zu bekommen. Schlimmstenfalls wird aus der Isolations-VM allmählich die normale Arbeitsumgebung und der Schutzeffekt ist perdu.

Praktikabler sind Tools, die automatische Isolationsumgebungen nutzen, um Dateien zu entschärfen, wie das Werkzeug Dangerzone (https://dangerzone.rocks). Es steht für Windows, macOS und Linux zur Verfügung und nutzt Container zur Isolation. Unter Windows und macOS kommt dafür Docker Desktop zum Einsatz unter Linux podman. Container bieten eine weniger gute Isolation als echte virtuelle Maschinen, stellen für Malware aber dennoch eine massive Hürde dar.

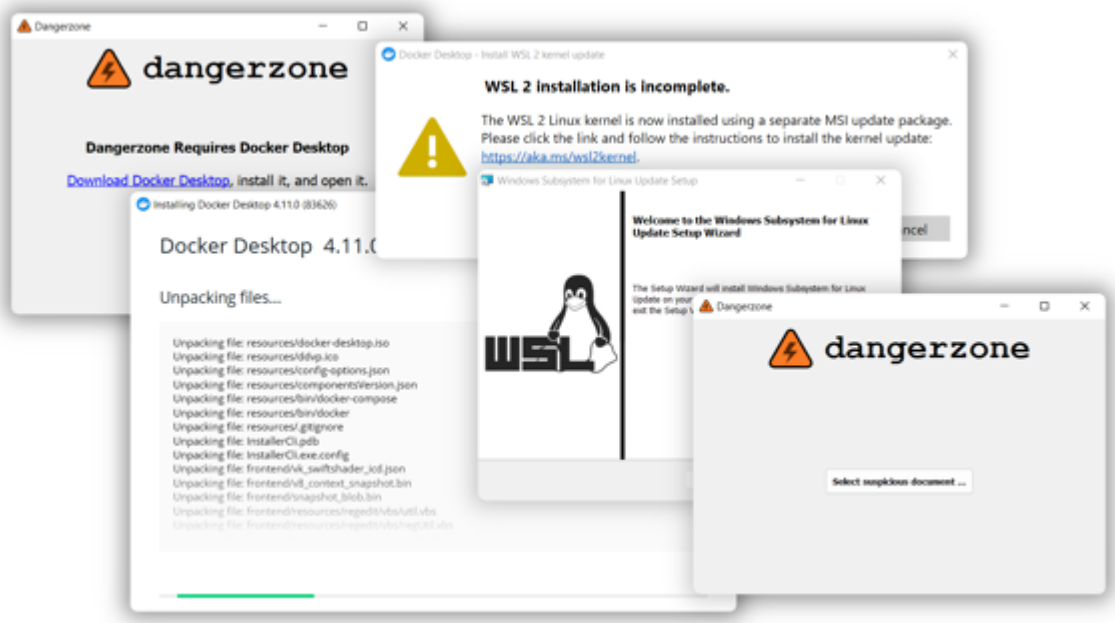
Die isolierten Container nutzt Dangerzone, um einen Anhang zu öffnen und in Bilddaten zu konvertieren. Malware können diese

Pixelbilder nicht enthalten und nur diese Daten lässt Dangerzone aus dem Container. In einem zweiten Schritt wird aus den Pixeldaten ein PDF erzeugt, damit man keine lose Bildsammlung als Ergebnis erhält. Das Resultat ist ein PDF mit optisch gleichem Inhalt wie das Eingangsdokument, aber garantiert ohne Malware, Makros, versteckte Inhalte, verheimlichte Linkziele und viele andere Arten von Bedrohung. Als Betriebssystem im Container nutzt Dangerzone Linux (auch unter Windows und macOS). Da die meisten Schädlinge auf Windows abzielen, ist es unwahrscheinlich, dass etwaiger Schadcode überhaupt ausgeführt wird, selbst wenn die Programme im Container Sicherheitslücken aufweisen sollten. Und auch wenn Malware die Software im Container kompromittiert und mit Linux zurande kommt, dann müsste sie immer noch aus dem Container ausbrechen, um Schaden anzurichten.

Bei so vielen Hürden kann man es verschmerzen, dass sich die Software im Container leider nicht leicht aktualisieren lässt: Der Installer von Dangerzone bringt ein fertiges Containerimage mit, damit die Software auch auf Rechnern ohne Internetzugang funktioniert. Wer sich nicht zutraut, das Containerimage selbst neu zu bauen – und eventuelle Inkompatibilitäten zu beheben –, bekommt erst mit einer neuen Dangerzone-Version ein neues Image. Das ist ein akzeptabler Kompromiss, aber wem er nicht reicht: Nichts spricht dagegen, noch eine Barriere hinzuzufügen und Dangerzone innerhalb einer VM zu betreiben.

Die Installation von Dangerzone erfordert unter Windows und macOS diverse Schritte, aber die sind relativ simpel: Zuerst laden Sie den Installer herunter und führen ihn aus. Danach können Sie Dangerzone bereits starten, erhalten aber den Hinweis, dass die Applikation Docker Desktop erfordert, sofern es nicht bereits installiert ist. Also folgen Sie dem angezeigten Link, laden Docker Desktop herunter und führen auch diesen Installer aus, was unter macOS mit ein paar Sicherheitsabfragen einhergeht, die Sie bestätigen müssen.

Danach starten Sie Docker und sind unter macOS nach ein paar Sekunden Startzeit einsatzbereit.



Die Installation von Dangerzone erfordert zwar eine Reihe von Schritten, ist aber nicht kompliziert.
Unter Windows beschwert sich Docker Desktop eventuell, falls das „Windows Subsystem for Linux 2“ (WSL 2) nicht bereitsteht. Aber auch in diesem Fall zeigt die Problemmeldung direkt den nötigen Link an. Sie müssen also nur eine weitere Runde aus Klick, Download und Installation drehen und nun ist Docker auch unter Windows zufrieden und zur Arbeit bereit. Nach einem Klick auf „Check again“ merkt das auch Dangerzone und macht sich daran, das Container-Image zu installieren. Das geht vollautomatisch vonstatten.

Die Installation unter Linux ist leichter oder schwerer, je nachdem, um welche Distribution es geht. Für einige Distributionen betreiben die Dangerzone-Entwickler eigene Repositories, was die Installation sehr einfach macht. Unter Debian genügen beispielsweise folgende Befehle:

```
curl -s https://packagecloud.io/install/repositories/firstlookmedia/code/script.deb.sh | sudo bash
sudo apt update
sudo apt install -y dangerzone
```

Ein Skript per curl herunterzuladen und direkt auszuführen, gilt allerdings zu Recht als höchst fragwürdige Installationsmethode. Wer dem Braten nicht traut, kann die Repositories manuell einrichten, die Dokumentation von Dangerzone erklärt, wie das geht (siehe ct.de/yw2x).

Leider unterstützt Dangerzone im Moment nur bei Debian aktuelle Versionen (11 und 12), bei Ubuntu und Fedora funktionieren von Haus aus nur etwas ältere Ausgaben (20.10, 21.04 und 21.10 beziehungsweise 33, 34 und 35). Auch bei anderen Distributionen sollten Sie sich nicht zu früh freuen: Beispielsweise findet sich Dangerzone zwar im User Repository von Arch Linux, allerdings ist das Paket aktuell nicht funktionstüchtig.

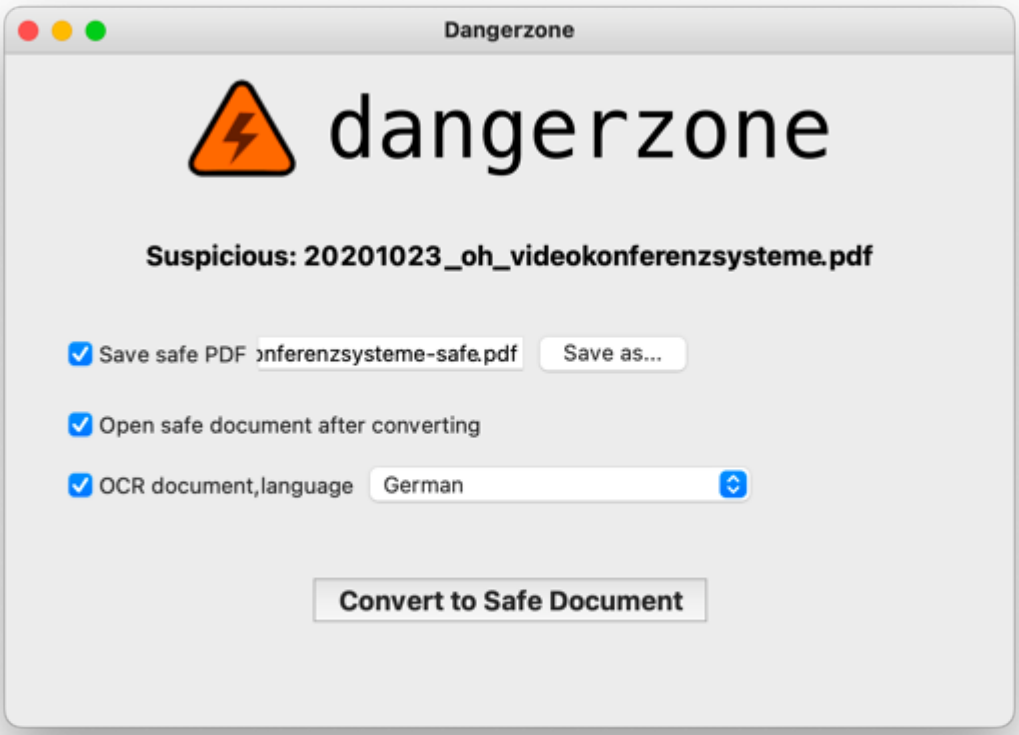
Statt sich unter Linux mit dem Paketbau oder Versionsinkompatibilitäten herumzuschlagen, bietet es sich an, einfach eine Debian-VM aufzusetzen und Dangerzone darin zu betreiben.

In der Gefahrenzone

Einmal fertig installiert, fällt die Bedienung von Dangerzone sehr leicht: Das Programm präsentiert nach dem Start nur eine Schaltfläche, die Sie drücken, um eine Datei zu konvertieren. Dangerzone kann diverse Office-Formate unschädlich machen, die ein Haupteinfallstor für Malware sind. Dazu startet das Programm im Container LibreOffice, um aus dem Office-Dokument ein PDF zu machen. Aus dem PDF werden dann Pixelgrafiken und daraus wieder ein – garantiert harmloses – PDF. Daneben können Sie mit Dangerzone auch PDFs und sogar Bilddateien entschärfen. Von letzteren geht nur eine geringe Gefahr aus, aber sicher ist sicher.

Nachdem Sie ein Dokument ausgewählt haben, bietet das Programm noch ein paar Einstellungen an. Dangerzone hat eine Texterkennung integriert (Optical Character Recognition, OCR) und fragt dafür nach der Sprache, in der das Dokument

vermutlich verfasst ist. So kann das Tool im zweiten Schritt die Bilddaten analysieren, um den Textinhalt eines Dokumentes zu rekonstruieren. OCR erhöht den Komfort erheblich, weil Sie dadurch im sicheren PDF Texte wieder markieren und kopieren können. Ein Klick auf „Convert to Safe Document“ stößt die Umwandlung an. Unter Linux und macOS erlaubt Dangerzone darüber hinaus, das Ergebnis-PDF automatisch zu öffnen, was Ihnen noch ein paar Klicks erspart.



Ein Klick und Dangerzone erzeugt eine garantiert harmlose Dateikopie mit dem gleichen (sichtbaren) Inhalt. So wird beispielsweise aus einem verseuchten Word-Dokument eine entschärfte PDF-Version.
Diese Bequemlichkeit können Sie unter Windows leicht nachrüsten, indem Sie die Kommandozeilenvariante von Dangerzone einspannen. Die wurde automatisch mitinstalliert, Sie können sie in der Eingabeaufforderung mit dem Befehl dangerzone-cli (für „command-line interface“) starten. Der Aufruf dangerzone-cli DATEI erstellt aus DATEI ein sicheres PDF, mit den Parametern --ocr-lang deu und --output-filename NEU.PDF schalten Sie die Texterkennung für Deutsch ein und

legen den Namen der Ergebnisdatei fest.

Damit kann man leicht ein Skript basteln, das Dateien konvertiert und öffnet. Unter ct.de/yw2x haben wir Ihnen drei Varianten bereitgestellt: Eine Batch-Datei, ein AutoHotkey-Skript und eine daraus erstellte EXE-Datei. Es ist eine gute Idee, eines der Skripte als Standardanwendung für Office-Dateien festzulegen. In Zukunft genügt dann ein Doppelklick auf die Datei, um Dangerzone zu starten, eine sichere Version zu generieren und diese zu öffnen. So vermeiden Sie auch, gefährliche Dateien versehentlich direkt zu öffnen. Bei Bedarf können Sie die Originaldokumente über das Kontextmenü weiterhin mit der üblichen Anwendung öffnen – wenn Sie sicher wissen, dass sie harmlos sind.

Qubes OS

Wenn man willens ist, aus Sicherheitsgründen das Betriebssystem zu wechseln, stehen noch bessere Lösungen als Dangerzone zur Verfügung. Nahe am Nonplusultra liegt Qubes OS, das VMs nutzt, um das gesamte System in Sicherheitszonen zu unterteilen. Im Detail haben wir Qubes OS in Ausgabe 11/2022 vorgestellt [4].

Unter Qubes OS können Sie beliebige Dateien weitgehend gefahrlos öffnen, indem Sie im Kontextmenü „View in disposable“ oder „Edit in disposable“ auswählen. Das System startet dann automatisch eine aktuelle VM und öffnet darin den Anhang mit der Standardanwendung. Wenn Sie die schließen, verwirft Qubes OS die komplette VM. Einzig die Änderungen an der Datei werden zurückgeschrieben, sonst nichts, und auch die Änderungen nur, wenn Sie die „Edit“-Option gewählt haben.

Schon das liefert mehr Sicherheit und Komfort, als man mit normalen VM-Lösungen erreicht. Zusätzlich gibt es die Tools qvm-convert-pdf und qvm-convert-img. Diese Werkzeuge waren die Vorlage für Dangerzone und funktionieren im Prinzip genauso. Allerdings nutzen die Qubes-OS-Befehle echte VMs und keine

Container. Das bietet noch mehr Schutz und ist leicht implementiert, wenn das Betriebssystem ohnehin alles in VMs verpackt.

Mit spitzen Fingern

Trotz solcher Helferlein ist Dangerzone mit Einschränkungen verbunden. Zum einen stellt das LibreOffice im Container Office-Formate nicht unbedingt so dar, wie Microsoft Office unter Windows sie anzeigt; zum Beispiel, weil im Container Schriftarten fehlen. Sie müssen also damit leben, dass die Ausgabedokumente von Dangerzone eventuell ein bisschen anders aussehen, als die Eingabedateien.

Zum anderen holpert die Texterkennung von Dangerzone gelegentlich, besonders wenn die Schrift im Dokument schlecht lesbar ist, etwa weil es sich um eine schnörkelige Schreibschrift handelt. Längere kopierte Passagen sollten Sie daher Korrektur lesen.

Das Hauptproblem von Dangerzone folgt aber aus seiner Funktionsweise: Als Ergebnis erhalten Sie immer ein PDF. Das reicht, wenn Sie das Dokument nur betrachten wollen, aber wenn Sie ein Word-Dokument bearbeiten, eine Excel-Tabelle für Berechnungen nutzen oder ein PDF-Formular ausfüllen wollen, dann kommen Sie so nicht weiter.

Immerhin können – und sollten – Sie in solchen Fällen das Dokument erst einmal mit Dangerzone konvertieren und öffnen, um den Inhalt auf Plausibilität zu prüfen. Ein angeblicher Geschäftsbericht gehört direkt in die Tonne, wenn der sichtbare Inhalt laut Dangerzone nur aus einem aufwendigen Banner besteht, das Sie auffordert, Makros zu aktivieren.

Aber was, wenn der Dateiinhalt plausibel aussieht? In diesem Fall kommen Sie nicht darum herum, das Dokument zu öffnen – allerdings nicht mit der Standardanwendung! Als absolutes Minimum können Sie beispielsweise den PDF-Reader im Browser

statt des Adobe Reader einspannen oder LibreOffice statt Microsoft Office. Das verringert zumindest die Chance, dass eventuell im Dokument eingebetteter Schadcode korrekt ausgeführt wird (siehe S. 21).

Deutlich sicherer ist es aber, verdächtige Dateien mit Werkzeugen zu öffnen, die den Inhalt analysieren und nicht direkt anzeigen. Was für Werkzeuge sich dafür eignen, hängt vom Typ der fraglichen Datei ab. Wir beschränken uns im Folgenden auf die beiden verbreitetsten Arten von Anhängen: Office- und PDF-Dateien. Bilder werden zwar ebenfalls sehr häufig verschickt, aber von üblichen Formaten wie JPG oder PNG geht nur eine geringe Gefahr aus. Wer solche Dateien weiterverarbeiten will, kann sie – nach einer Inspektion per Dangerzone – in der Bildbearbeitung seiner Wahl öffnen. Das verbleibende Restrisiko ist sehr gering.

Zur Analyse von PDFs und Office-Dateien stellen wir Ihnen zwei Werkzeugsammlungen vor, die beide auf der Kommandozeile laufen. Lassen Sie sich davon nicht abschrecken, eine erste Analyse ist wirklich nicht schwer.

PDF-Tools

Der Sicherheitsforscher Didier Stevens hat eine Reihe von Werkzeugen geschrieben, um PDF-Dateien zu analysieren und bietet sie auf seiner Webseite als Zip-Archive zum Download an (siehe ct.de/yw2x). Um eine Datei grob einzuschätzen, eignet sich das Tool pdfid. Laden Sie das zugehörige Archiv von Didiers Website und entpacken Sie den Inhalt in ein beliebiges Verzeichnis. Das Tool ist in Python geschrieben; wie Sie die dafür nötige Laufzeitumgebung installieren, haben wir in c’t 5/2022 ausführlich erklärt [5].

Wenn Sie zum Beispiel die PDF-Datei verdaechtig.pdf mit

```
python pdfid.py verdaechtig.pdf
```

öffnen, gibt das Programm eine Liste von Schlüsselwörtern

zurück, die es im PDF gefunden hat:

```
PDFiD 0.2.8 verdaechtig.pdf
 PDF Header: %PDF-1.1
 obj                    9
 endobj                 9
 stream                 2
 endstream              2
 xref                   1
 trailer                1
 startxref              1
 /Page                  1
 /Encrypt               0
 /ObjStm                0
 /JS                    1
 /JavaScript            1
 /AA                    0
 /OpenAction            1
 /AcroForm              0
 /JBIG2Decode           0
 /RichMedia             0
 /Launch                0
 /EmbeddedFile          1
 /XFA                   0
 /URI                   0
 /Colors > 2^24         0
```

Im Grunde sucht pdfid lediglich in der Datei nach diesen Schlüsselwörtern, die als ASCII-Zeichen vorliegen müssen. Wie so oft ist es in Praxis komplizierter: PDFs erlauben die Zeichenketten unterschiedlich zu kodieren, womit pdfid aber zurande kommt.

Achten sollten Sie besonders auf die Schlüsselwörter /JS und /JavaScript, die einen Wert größer 0 anzeigen, wenn das PDF vermutlich JavaScript-Code enthält. JavaScript kommt auch in einigen gutartigen PDFs vor, wo es beispielsweise Formulareingaben validiert. Nichtsdestotrotz sollten Sie JavaScript-Code als deutliches Warnsignal betrachten.

Ebenfalls Warnsignale stellen die Schlüsselwörter /AA,

/OpenAction und /AcroForm dar. Werte größer 0 bedeuten dort, dass der PDF-Reader automatische Aktionen starten soll, wenn man ein Dokument öffnet. Auch das kann harmlos sein und den Reader beispielsweise anweisen, eine bestimmte Seite des Dokuments anzusteuern – oder es führt Skriptcode aus und platziert Malware auf dem Rechner.

Wenn Sie auch nur eines dieser Schlüsselwörter entdecken, löschen Sie das verdächtige PDF, um auf Nummer sicher zu gehen. Wenn es dafür zu wichtig und dringend ist, dann hilft der Parameter --disarm (oder -d) von pdfid:

```
python pdfid.py -d verdaechtig.pdf
```

Das Programm produziert damit eine Kopie der Datei mit der Endung „.disarmed.pdf“. In der Kopie ist die Groß- und Kleinschreibung kritischer Schlüsselwörter vertauscht, aus /JavaScript wird /jAVAsCRIPT, aus /OpenAction wird /oPENaCTION und so weiter. So geschrieben handelt es nicht um gültige Schlüsselwörter und PDF-Reader sollten sie ignorieren. Diese entwaffnete Variante der Datei können Sie risikoarm öffnen.

Wem auch das nicht reicht, der kommt um eine detaillierte Analyse der internen Struktur des Dokuments nicht herum. Nur so findet man gefahrlos heraus, welche Aktionen genau ausgeführt würden und was genau der JavaScript-Code täte. Das erfordert allerdings Programmierkenntnisse, Wissen über den internen Aufbau von PDFs und mehr Platz, als dieser Artikel bietet. Wir werden in einer der folgenden Ausgaben zeigen, wie man bei so einer Analyse vorgeht.

Office-Dateien

Auch um Office-Dateien zu untersuchen, gibt es Kniffe und Werkzeuge in der Art von pdfid, aber nicht immer benötigen Sie dergleichen: Microsofts neuere Formate, die auf X enden (DOCX, XSLX, PPTX), sind im Grunde Zip-Archive, die lediglich einen speziellen Inhalt haben. Das hilft, falls Sie beispielsweise nur an den Bildern in einem Word-Dokument interessiert sind.

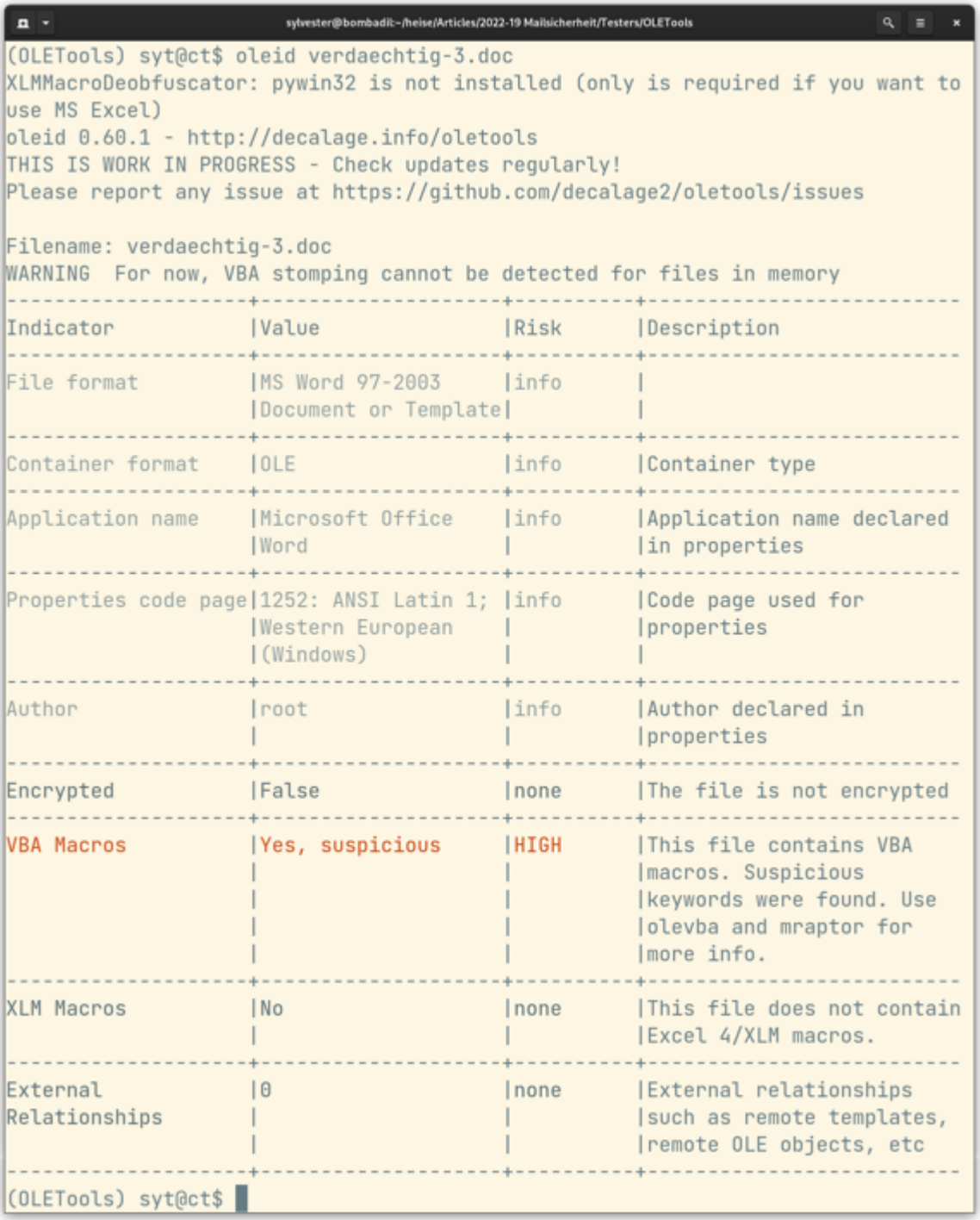
Dann ändern Sie einfach die Endung von .docx in .zip, öffnen das Archiv mit dem Zip-Programm Ihrer Wahl und inspizieren die Bilder im entpackten Verzeichnis /word/media/.

Wenn Sie die Office-Dateien aber auf Unbedenklichkeit prüfen und letztlich in Word oder Excel bearbeiten wollen oder wenn es um ältere Formate geht (DOC, XLS …), dann funktioniert dieser Trick nicht. Was funktioniert, sind die oletools des Programmierers Philippe Lagadec (siehe ct.de/yw2x). Auch dieser Werkzeugkasten nutzt Python, am einfachsten installieren Sie ihn über die Paketverwaltung pip [5]:

```
pip install -U oletools[full]
```

Die oletools lesen sowohl die alten Office-Binärformate (wie DOC) als auch die aktuelleren auf XML-Basis (etwa DOCX). Für eine Einschätzung einer verdächtigen Datei ist das Programm oleid gedacht. Wie pdfid gibt es einen Überblick über relevante Aspekte einer Office-Datei. Statt einer bloßen Liste liefert oleid allerdings eine Tabelle samt Risikoeinschätzung der Elemente und schreibt im Fall der Fälle auch noch Handlungsanweisungen dazu (siehe Bild auf S. 31) Einer Word-Datei ohne Makros, externe Objekte oder andere Spezialitäten attestiert das Programm beispielsweise ein geringes Risiko: In der Spalte Risk sind alle Werte „info“ oder „none“.

Im Testdokument des heise Mailchecks (siehe S. 21) erkennt oleid korrekterweise ein VBA-Makro und bewertet es mit dem Risiko „Medium“. In der letzten Spalte steht, warum und was Sie jetzt tun können: „No suspicious keyword was found. Use olevba and mraptor for more info.“ Es wurden also keine Alarmsignale im Makro selbst gefunden, für Details soll man die Werkzeuge olevba oder mraptor nutzen.

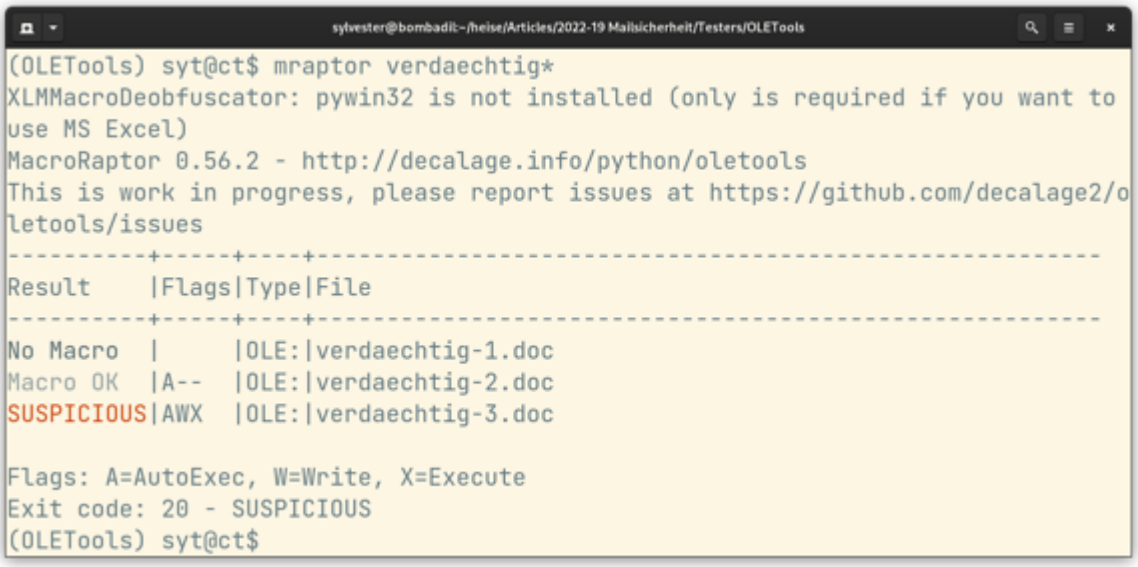


„VBA Macros: Yes, suspicious; Risk: HIGH“ meldet oleid und hat recht. Diese Datei ist tatsächlich höchst suspekt.
Ein Dokument mit einem höchst suspekten Makro, das versucht, eine Datei auf die Festplatte zu schreiben, bewertet oldid in Rot als „suspicious“ (verdächtig) und warnt in Großbuchstaben vor dem hohen Risiko, weil es verdächtige Schlüsselwörter im Makro gefunden hat.

Der wieder empfohlene Aufruf von mraptor erklärt den Verdacht näher: Das Makro wird automatisch ausgeführt („AutoExec“),

schreibt Daten („Write“) und versucht etwas außerhalb des Makro-Codes aufzurufen („Execute“). Folgerichtig kommt mraptor zu dem Schluss, dass die Datei verdächtig ist.

Wer es noch genauer wissen will, greift zum Werkzeug olevba. Es zeigt den enthaltenen Makrocode an, was aufschlussreich ist, wenn man Programmierkenntnisse hat. Zudem liefert olevba eine noch detailliertere Tabelle mit gefundenen problematischen Schlüsselwörtern und was sie bedeuten (siehe Listing auf S. 32).



mraptor kann man auch mehrere Dateien auf einmal vorwerfen. Er liefert dann eine Tabelle, ob Makros gefunden und als verdächtig bewertet wurden.

Listing: Output von olevba

```
+----------+--------------------+----------------------------
---------------+
|Type      |Keyword             |Description
|
+----------+--------------------+----------------------------
---------------+
|AutoExec  |AutoOpen            |Runs when the Word document
is opened      |
|Suspicious|Environ             |May read system environment
variables      |
|Suspicious|Open                |May open a file
```

```
|
|Suspicious|Write                  |May write to a file (if
combined with Open)  |
|Suspicious|Put                    |May write to a file (if
combined with Open)  |
|Suspicious|Binary                 |May read or write a binary
file (if combined |
|              |                                |with Open)
|
|Suspicious|CreateObject           |May create an OLE object
|
+----------+---------------------+-----------------------------
----------------+
```

Das Helferlein olevba extrahiert nicht nur Makrocode aus Office-Dateien (hier nicht gezeigt), sondern meldet auch, welche interessanten Begriffe sich im Code finden und worauf sie hindeuten.

Fazit

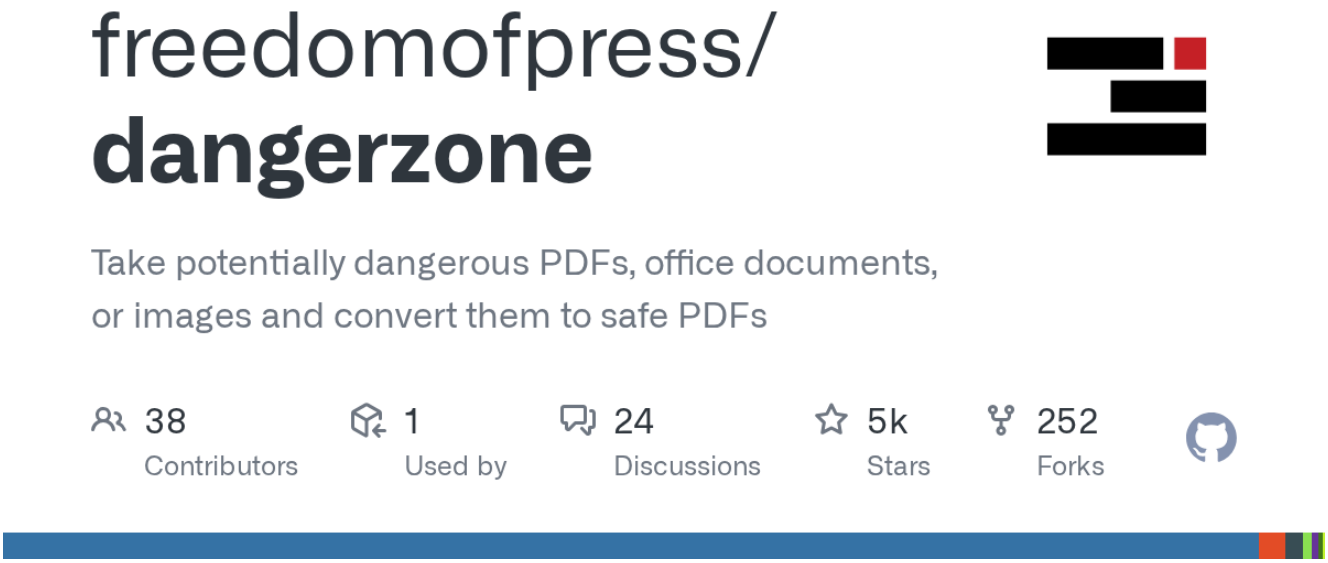
Auch ohne weitere Analyse müssen Sie keine Angst vor bösartigen Anhängen haben, wenn Sie die in diesem Artikel vorgestellten Werkzeuge einsetzen. Das Risiko, dass etwas den Filter von Dangerzone passiert, ist extrem gering. Übrigens sammeln sich unter Windows und macOS mit der Zeit immer mehr „Containers“ (mit Status „Exited“) und „Volumes“ in Docker Desktop an, zwei für jeden Aufruf von Dangerzone. Sie können die Einträge einfach ignorieren – oder aufräumen, wenn Sie die Unordnung stört. Löschen Sie einfach alle Exited-Container, die zugehörigen Volumes entsorgt Docker Desktop gleich mit. Dangerzone benötigt lediglich den Eintrag unter „Images“ und falls sie diesen versehentlich löschen sollten, legt das Programm ihn automatisch neu an.

Wenn Sie ein Dokument doch im Original öffnen müssen, dann reichen pdfid, oleid und Konsorten, um Gefahren zu wittern, bevor es zu spät ist. Das genügt für den Eigenschutz, aber wenn Sie die Neugierde packen sollte, dann sehen Sie sich

weiter in den Werkzeugkisten von Stevens und Lagadec um. Die enthalten noch viele weitere Programme, mit denen man den Inhalten von Office- und PDF-Dateien auf den Grund gehen kann. Ein Beispiel dafür werden wir in einer der kommenden Ausgaben beschreiben. (*syt@ct.de*)

1. Literatur
2. Ronald Eikenberg, Hacking-Stick, Kali Linux auf USB-Stick einrichten, c’t 23/2021, S. 30
3. David Wolski, Buntes Hacker-Linux, Linux-Distribution: Parrot Security für Pentester und Hacker, c’t 14/2020, S. 98
4. Sylvester Tremmel, Neue Stammkneipe, Wie Sie die passende Distribution für sich finden, c’t 3/2022, S. 30
5. Knut von Walter, Von Snowden empfohlen, Das sicherheitsorientierte Betriebssystem Qubes OS im Test, c’t 11/2022, S. 94
6. Ronald Eikenberg, Jan Mahn, Draufgebeamt, Python schnell und einfach einrichten, c’t 5/2022, S. 20

Downloads: ct.de/yw2x



Installing Dangerzone · freedomofpress/dangerzone Wiki

Take potentially dangerous PDFs, office documents, or images and convert them to safe PDFs – Installing Dangerzone · freedomofpress/dangerzone Wiki

PDF Tools

Here is a set of free YouTube videos showing how to use my tools: Malicious PDF Analysis Workshop. pdf-parser.py This tool will parse a PDF document to identify the fundamental elements used in the…

GitHub – decalage2/oletools: oletools – python tools to analyze MS OLE2 files (Structured Storage, Compound File Binary Format) and MS Office documents, for malware analysis, forensics and debugging.

oletools – python tools to analyze MS OLE2 files (Structured Storage, Compound File Binary Format) and MS Office documents,

for malware analysis, forensics and debugging. – GitHub – decalage2/oleto…