**Wenn sich in der PDF-Datei ein Linux tummelt**

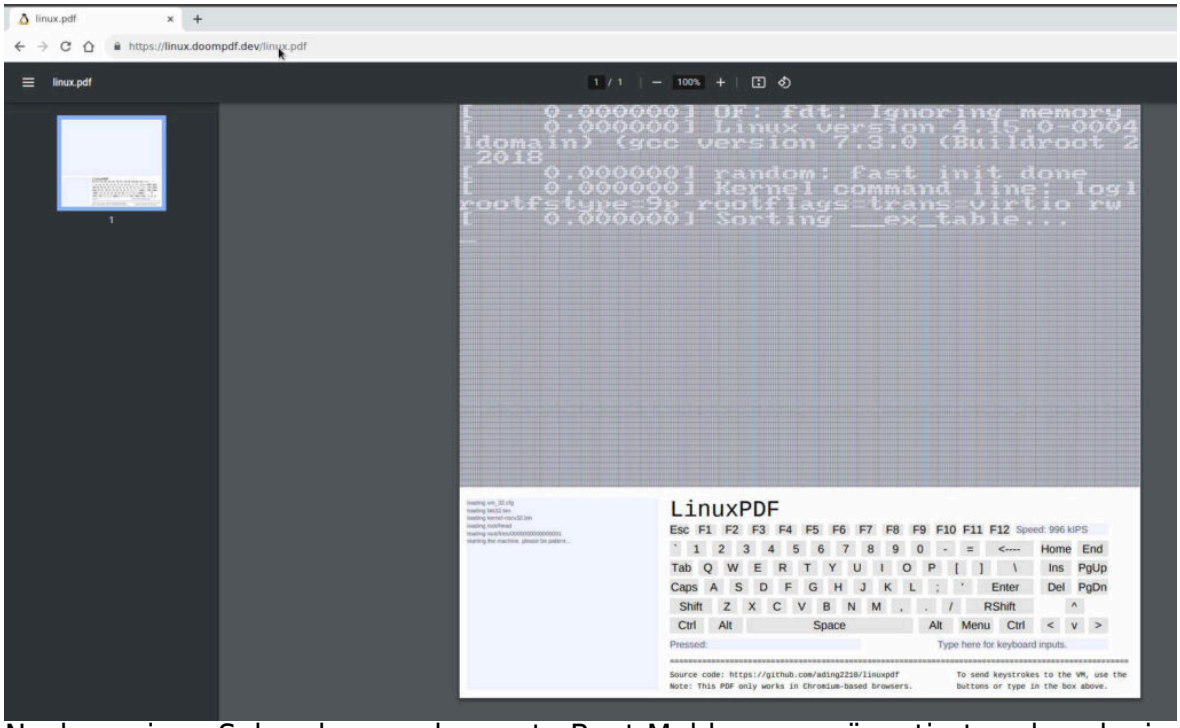
***Egg, 7. Februar 2025:*** *Mit der Version 2025/II wurde im letzten Blog die neue Mail-Archivierung vorgestellt. Ebenfalls neu ab Version 2025/II gelten neue Regeln, wie Dokumente verarbeitet werden. Am Beispiel einer PDF-Datei, in der soviel JavaScript-Code steckt, dass ein gesamtes Linux ausgeführt wird, soll veranschaulicht werden, warum es Sinn ergibt, bei der Entgegennahme von Dokumenten viel Sorgfalt walten zu lassen — und warum die ArchivistaBox in dieser Hinsicht einzigartig ist.*

**Betriebssystem läuft in PDF-Datei**

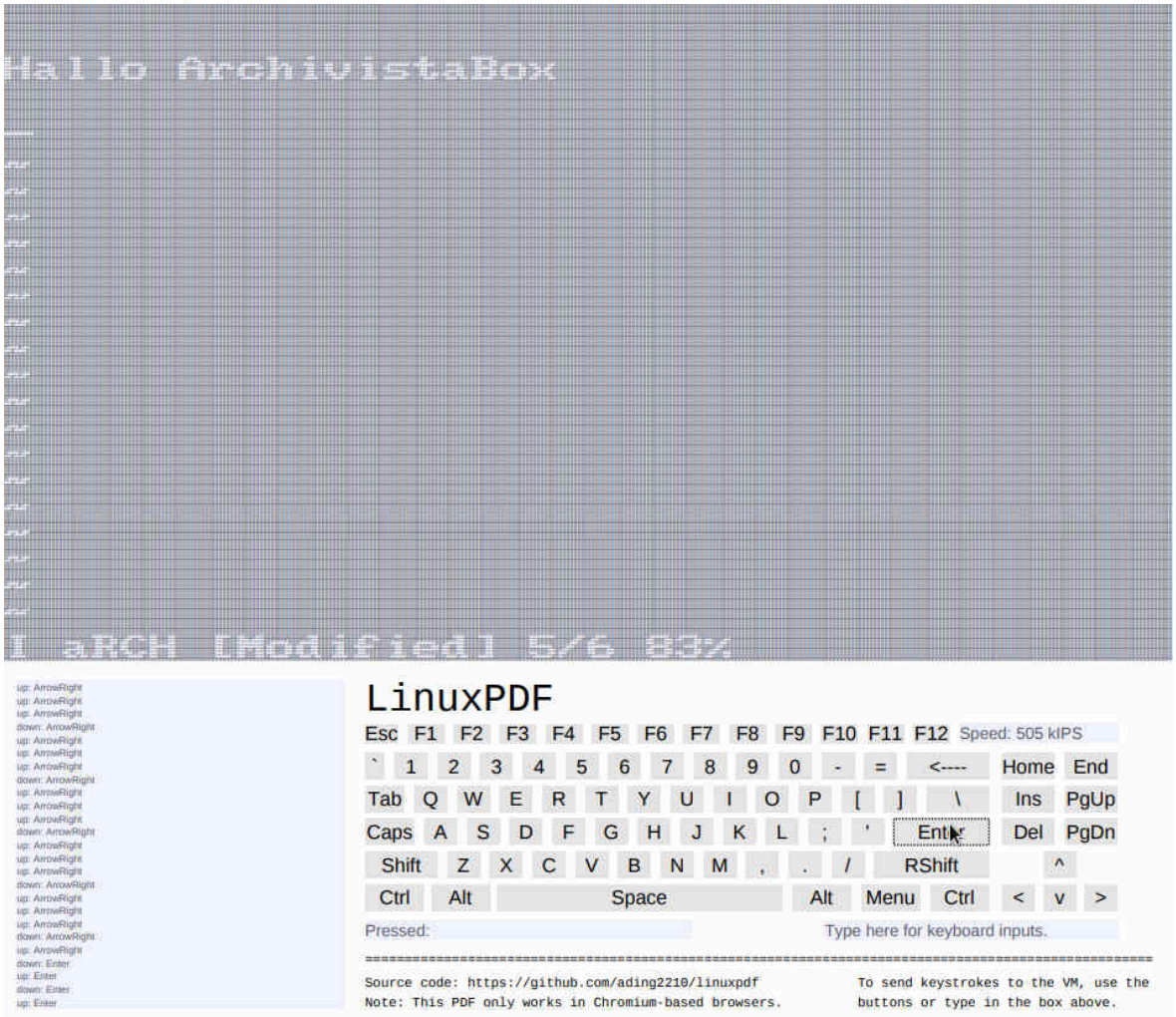
PDF-Dateien gelten allgemein als das Austauschformat, wenn es darum geht, Inhalte über das Netz auszutauschen. Üblicherweise kann davon ausgegangen werden, dass diese Dokumente auf fast jeder Plattform (Windows, Mac, Linux, iOS und Android) dargestellt werden können.

Nur wenige wissen hingegen, dass PDF-Dateien auch ausführbare Programme enthalten können (Stichwort JavaScript). Gemäss einem Artikel bei **heise.de** hat ein findiger Tüftler ein komplettes Mini-Linux in eine PDF-Datei verfrachtet. Die **entsprechende PDF-Datei (ca. 5 MByte) kann auf unserem Server** bezogen werden.

Damit das Mini-Linux starten kann, ist ein Browser mit «Chrome-Engine» erforderlich, mit Firefox oder Safari arbeitet der JavaScript-Code nicht. Um Linux in der PDF-Datei zu starten, genügt es, die Datei linux.pdf mit Chrome (oder Chromium, Edge nicht geteste) zu starten:




Nach wenigen Sekunden werden erste Boot-Meldungen präsentiert und nach ein bis zwei Minuten steht eine Linux-Konsole zur Verfügung. Wer mag, kann mit dem Texteditor ‚vi' auch eine kleine Grussbotschaft verfassen:




Es wäre vermessen zu behaupten, Linux in der PDF-Datei laufe mit hohem Tempo, das Gegenteil ist der Fall, es geht sehr gemütlich zu und her. Von daher kann das Beispiel als nette Spielerei abgehakt werden, oder etwa doch nicht?

**Was sind Daten, was ein Programm ?**

Fassen wir zusammen, in einer ca. 5 MByte grossen PDF-Datei läuft ein Linux. Handelt es sich dabei nun um eine Datei oder ist es ein Programm oder neudeutsch gar eine App? Wird es bald einen PDF-Store geben? Oder allgemeiner gefragt, wie können Daten von Programmen unterschieden werden? Allgemein kann gesagt werden, Daten sind Informationen, die nicht ganz flüchtig sind. Sie können zwar geändert werden, aber dies erfolgt von Zeit zu Zeit, d.h. es gibt statische Zustände. Bei Programmen geht es darum, nach einer mehr oder grossen Vielzahl von Aktionen (statische) Resultate zu erhalten, die dann (optional) als Daten gespeichert werden.

Die Programme selber sind meist eine Art BlackBox. Selbst wenn der Code offengelegt ist, nur die wenigsten dürften wissen, wie Programme im Detail arbeiten. Kleine Nebenbemerkung: Leider ist es auch oft so, dass nicht einmal die Programmierer/innen später noch wissen, wie ihr Code funktioniert...

Unschwer lässt sich erkennen, eine 100% Abgrenzung gibt es nicht, denn ein Programm ist (als ausführbare) Datei statisch. Und weil es auch bei Dokumenten meistens darum geht, dass diese geändert werden können, so kann es durchaus Sinn ergeben, für diese Daten wiederkehrende Abläufe als Makro oder Skript in eine Datei zu integrieren.

Was wäre heutzutage eine Homepage (html-Datei) ohne irgendwelche Schnippsel an JavaScript? Von daher betrachtet darf angemerkt werden, dass JavaScript in PDF-Dateien nicht grundsätzlich ein Problem darstellen müssten.

**Wenn Programme ungefragt gestartet werden**

Jedoch, das Linux-PDF-Dokument zeigt die Problematik, denn wer die Datei öffnet, der wird ja nicht gefragt, ob Linux gestartet werden soll oder nicht. Der Code wird automatisch gestartet. Natürlich kann hier wieder angeführt werden, dass Benutzer/innen auch nicht gefragt werden, ob JavaScript-Programme auf Webseiten gestartet werden oder nicht.

Das mag schon sein, aber eine Homepage ist ja auch nicht primär eine statische Datei, dies im Unterschied zum PDF-Dokument, wo doch im Grundsatz Informationen in Bild- oder Schriftform statischer Art ausgetauscht werden. Das Problem bei JavaScript in PDF-Dateien liegt primär darin, dass der Code ohne Interaktion beim Öffnen direkt gestartet wird.

**Sicherheit bei der ArchivistaBox**

Nun gehören PDF-Daten zum täglichen Brot einer jeden ArchivistaBox bzw. wohl auch eines jeden Dokumenten Management Systems (DMS). Mit Version 2025/II wurde die Art und Weise, wie Dokumente verarbeitet werden, weiter verfeinert. Bislang war es so, dass, sobald die ArchivistaBox ein Dokument nicht einem passablen Dateiformat zuordnen konnte, einfach mit LibreOffice PDF-Dateien aus den originären Daten erstellt wurde.

Neu werden eingehende Dokumente auf ausführbare Programme getestet. Ist dies der Fall, versucht LibreOffice nicht mehr, die Daten lesbar aufzubereiten, indem LIbreOffice eine Text-zu-PDF-Konvertierung vornimmt.

Dies bedeutet, dass von den entsprechenden Dateien kein virtuelles Abbild erstellt wird. Und damit wären wir beim wichtigsten Vorteil, den die ArchivistaBox seit bald 30 Jahren zu bieten hat.

Der «übliche» Fall ist ja nicht, dass Programme angeliefert werden, sondern es werden (nicht abschliessend) Schriftgut, Bilder, Mails und/oder multimediale Daten verwaltet. Alle eingehenden Informationen werden dabei virtuell «abfotografiert». Damit ist es in ArchivistaDMS nicht nötig, die Daten für eine spätere Ansicht öffnen zu müssen.

Vielmehr werden die erstellten Bilddaten angezeigt. Damit ist es nicht notwendig, irgendwelche Skripte oder Programme zu starten, um z.B. über einen Viewer PDF- und/oder Office-Dokumente darzustellen, wie dies bei (allen) anderen Lösungen der Fall ist. Damit ist ArchivistaDMS nicht durch Skripte oder Programme «angreifbar». Wo keine Programme gestartet werden, gibt es auch kein Problem mit der Sicherheit.

Damit entscheiden alleine unsere Kundinnen und Kunden, welche Tools für ihre Daten zum Einsatz kommen. Bei der Archivista GmbH wird z.B. 100% ohne die Office-Pakete von Microsoft gearbeitet, ebenfalls gibt es kein Outlook oder Teams. LibreOffice wie auch der PDF-Viewer sind so konfiguriert, dass beim Öffnen von Dateien keine Programme gestartet werden.

***Praxis-Tipp:*** *All jene, die mit Windows oder Mac arbeiten, können bei den Office-Programmen und auch beim PDF-Viewer das automatische Starten von Skripten in Dateien deaktivieren. Noch besser (falls möglich) ist, das Starten von Programmen in Dokumenten ganz abzuschalten.*

**Maximal minimierte Risiken**
Solange beim Öffnen bzw. Verarbeiten von Dokumenten kein in Dokumenten liegender Code ausgeführt wird (und dies ist bei der ArchivistaBox zu 100% der Fall), besteht faktisch kein entsprechendes Risiko (Security by Design).
Ebenso einzigartig ist die Tatsache, dass die ArchivistaBox im Hauptspeicher arbeitet. Selbst wenn der Kunde (z.B. über den Desktop der ArchivistaBox) selber zusätzliche Programme (mit Schadcode) installieren sollte, so lässt sich der Schadcode durch einen Neustart der ArchivistaBox wieder entfernen, denn bei jedem Neustart wird die ArchivistaBox komplett (und automatisiert) neu im Hauptspeicher installiert.
Bleibt die Problematik, dass auch Betriebssysteme Schadcode enthalten können. Bei Open Source ist es aber doch so, dass allfällige Schwachstellen sehr schnell behoben werden können, weil der Quellcode bekannt ist. Falls notwendig, erhalten alle Kundinnen und Kunden aktuelle Versionen innert Stunden. Der Update-Prozess wird dabei stets von der ArchivistaBox aus (WebConfig) gestartet, es besteht kein automatisierter Zugriff seitens des Lieferanten. Auch dies ist ein Plus bei Sicherheit, weil damit automatisierte Angriffe nicht stattfinden können.
Aktuell arbeiten die allermeisten Informatik-Lösungen ohne Anbindung an das Internet nicht mehr. Die ArchivistaBox lässt sich problemlos ganz ohne Netz, mit Maus, Tastatur und Bildschirm betreiben. Dies mag für heutige Verhältnisse als vielleicht übertrieben erscheinen, und doch kann es für dieses Szenario nach wie vor gute Gründe geben. Als Beispiel seien selbstragene Archive genannt.
Anhand einer Abfrage werden Dokumente in eine eigene ArchivistaBox exportiert. Diese kann danach in einem «Käfig» (eigener Rechner oder mit Virtualisierung) gestartet werden, ohne dass eine Anbindung ans Netz aufgebaut werden muss.

**Mit der ArchivistaBox wäre dies nicht passiert**

Als Fazit darf hier angeführt werden, wer denkt, Angriffe über PDF-Dateien (oder ganz allgemein Dokumente) seien kein Problem oder würden nicht stattfinden, dem darf gerne (ich weiss, ist Boulevard, dafür verständlich geschrieben) die Seite **Whatsapp-Wurm** aufrufen, wo über präparierte PDF-Dateien (analog zu linux.pdf) die Geräte erfolgreich angegriffen wurden.

Nochmals, so gestartete Angriffe sind über die ArchivistaBox nicht möglich, da erstens von den PDF-Dateien Bilddaten erstellt werden und dabei die JavaScript-Fragmente nicht abgearbeitet werden. Und zweitens werden anstelle der PDF-Dateien in ArchivistaDMS die Bilddaten dargestellt. In diesem Sinne ist die ArchivistaBox bestens gerüstet für den harten Business-Alltag.

Das heisst nicht, dass keine weiteren Verbesserungen möglich sind. Gerade mit Version 2025/II wird durch den neuen Dokumentenparser sichergestellt, dass problematische Dokumente gar nicht mehr verarbeitet werden. In diesem Sinne empfehlen wir das Upgrade auf die Version 2025/II allen Kundinnen und Kunden.

**Das neu Security-Konzept der ArchivistaBox 2025/II greift aber nur, wenn die Updates auch zeitnah eingespielt werden. WebConfig aufrufen und dort Online-Update anwerfen, mehr braucht es nicht.**

Optional können wir die ArchivistaBox für unsere Kundinnen und Kunden in regelmässigen Abständen entsprechend updaten, denn das beste Security-Konzept bringt wenig, wenn die entsprechenden Anpassungen zwar bestehen, aber nie eingespielt wurden.