SLAC 2023
23.–25. Mai 2023 | Berlin

Folien mit Anmerkungen ab Seite 42

PeekabooAV - Komponenten und Erfahrungen der OpenSource Sandbox-Analyse

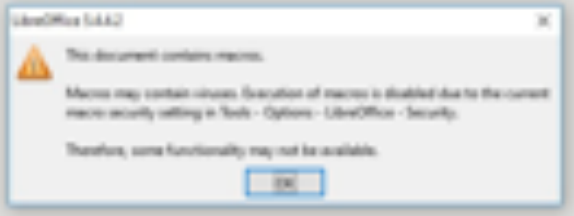
Felix Bauer
Christoph Herrmann
Michael Weiser

Gestern 6 Jahre PeekabooAV

Vor **4 Jahren** haben wir auf der **SLAC** zum zweiten Mal von PeekabooAV berichtet, wie E-Mailanhänge auf Schadsoftware geprüft werden können. **Heute** stellen wir die **gesamte Analyse-Pipeline** vor, von der Schnittstelle ins Mailsystem (**rspamd**), über das Entpacken (**expander**) bis zur Ausführung in der Sandbox mit **Windows 10**.

Wie es dazu kam, **Stand und Zukunft**!

Cuckoo ist tot, lang lebe **Cuckoo 3**, **CAPEv2** und **Drakvuf**, **Cortex** und **Karton**.


LibreOffice 5.4.6.2
This document contains macros.
Macros may contain viruses. Execution of macros is disabled due to the current macro security setting in Tools - Options - LibreOffice - Security.
Therefore, some functionality may not be available.
OK

Alles beim Alten

Wichtiges Dokument

An: felix@peekaboosv.de

Cc:

Nachricht | CheckVM.xls (37 KB)

Hallo Felix

im Anhange wie mit deinem Chef besprochen das Dokument zum Ausfüllen.
Das Formular benötigt Makros.
Bitte gleich machen und Ergebnis, wie im Dokument beschrieben, weiterleiten.

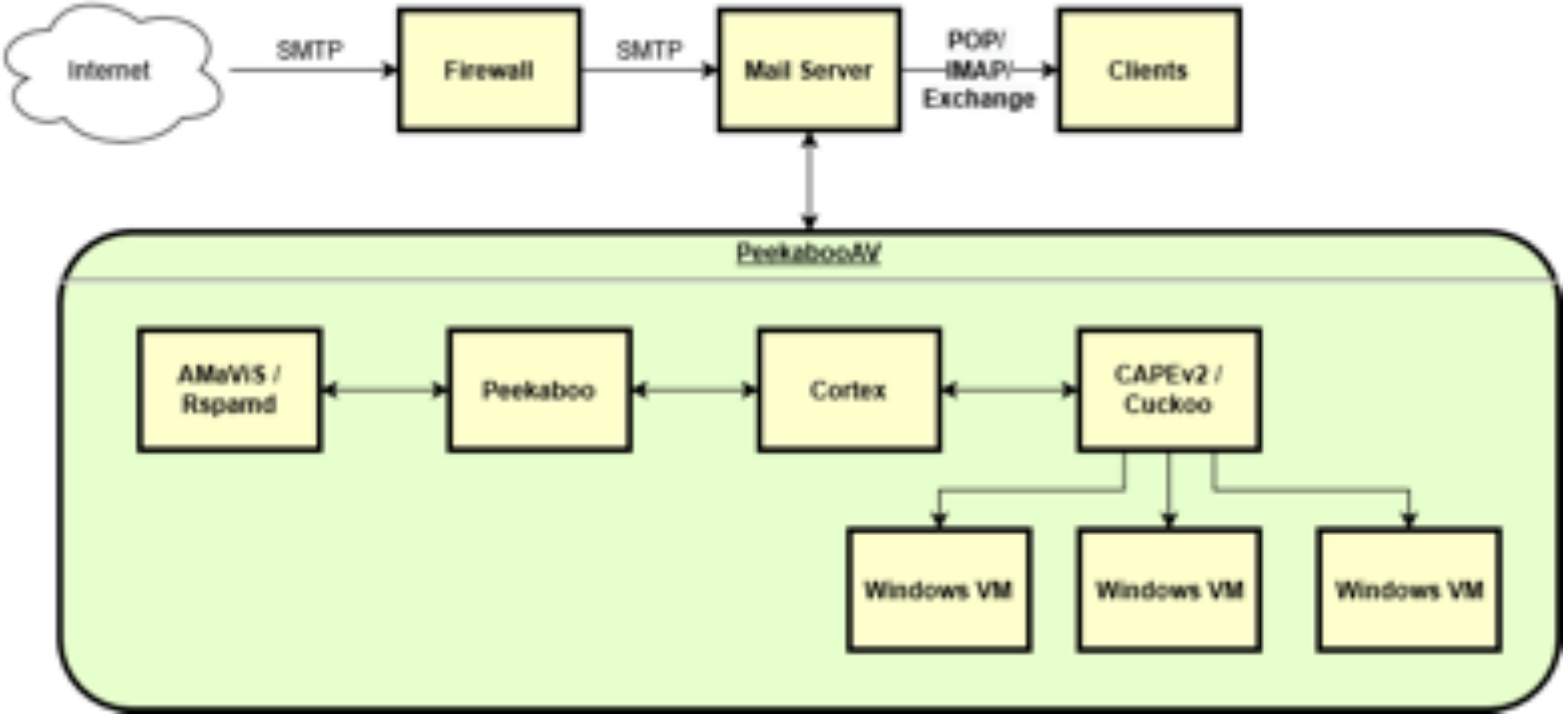
Danke
Gerda

Was ist Peekaboo?

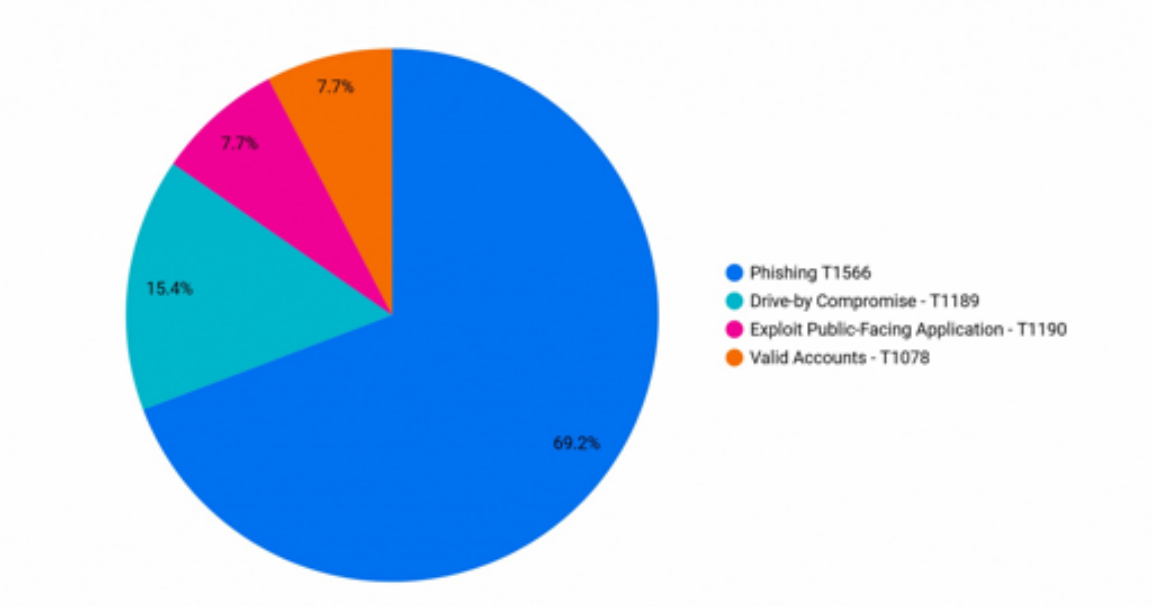
Peekaboo ist das **Bindeglied** zwischen dem **Mailsystem** und der **Sandboxanalyse**!

Peekaboo gibt die **Anhänge** zur Analyse, wertet aus und meldet die **Entscheidung** zurück ins Mailsystem.

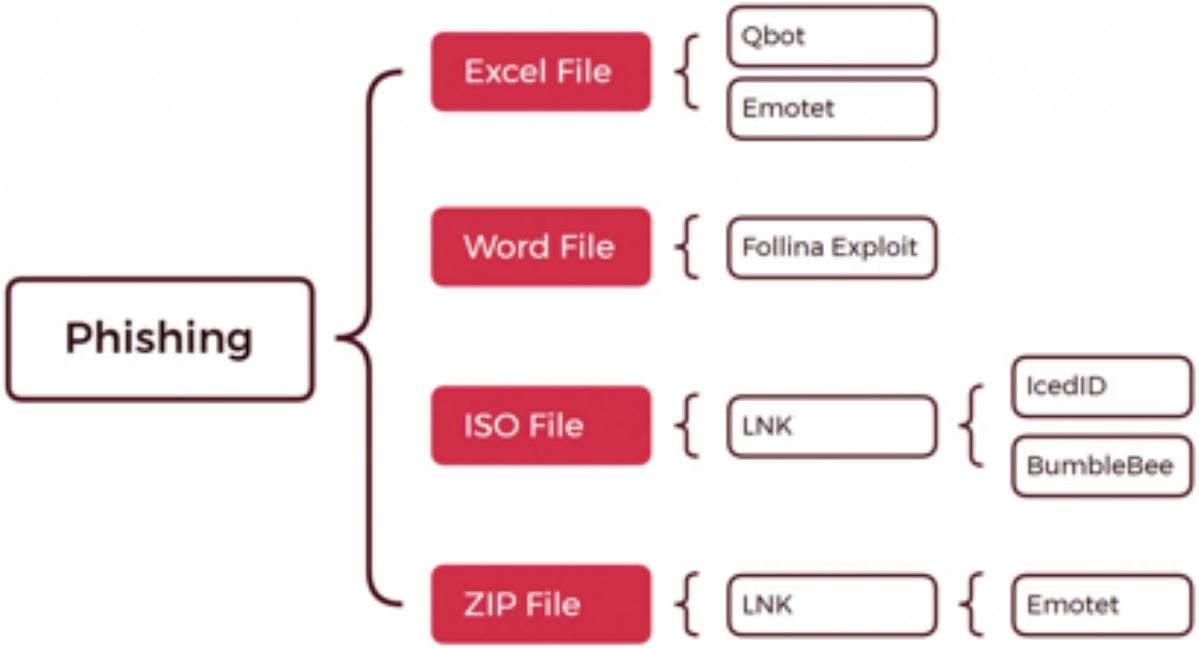
Peekaboo im Mailsystem



The DFIR report 2022 Year in Review - Initial Access



https://thedfirreport.com/2023/03/06/2022-year-in-review/



https://thedfirreport.com/2023/03/06/2022-year-in-review/

Zwei Folien von **Christoph**

Bericht aus der **Praxis**

Kuriositäten und Realitätsabgleich

```
To: "" <Hans.Hoffmann@domain.de>
Subject: MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----------------JHPn4GLt5KrvuXRamCO6TZy8"


This is a multi-part message in MIME format.
------------------JHPn4GLt5KrvuXRamCO6TZy8
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: quoted-printable


<html>
<head>
<meta http-equiv=3DContent-Type content=3D"text/html; charset=3Diso-8859-1"=
>
</head>
<body>
<br>

</body>
</html>
------------------JHPn4GLt5KrvuXRamCO6TZy8
Content-Type: application/vnd.ms-excel;
 name="IhreRechnung.xls"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="IhreRechnung.xls"


0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAEAAAA/AEAAAAAAAAA
...
```

```
Subject: =?UTF-8?B?UmU6IEplbm55IEtsZWluZW1leWVyIC0gQsO8cm8gU2lsdmlhIEJyZWg=?=
 MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----------------xF06JPJRGLwHam8aTrmtfDHD"

This is a multi-part message in MIME format.
------------------xF06JPJRGLwHam8aTrmtfDHD
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: quoted-printable

<html>
<head>
<meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3Diso-2022-=
jp">
</head>
<body>
=0DAls Anhang erhalten Sie Ihre Rechnung.<br>
<br>
<br>
<br>

Organisation<br>
Mail <a href=3D"mailto:info@domain.de">info@domain.de</a><br>
<a href=3D"http://www.domain.de">www.domain.de</a>
</body>
</html>
------------------xF06JPJRGLwHam8aTrmtfDHD
Content-Type: application/vnd.ms-excel;
 name="Scan 2022.11.11_1346.xls"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="Scan 2022.11.11_1346.xls"

0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAACAAAAtAAAAAAAAAAA
EAAA/v///wAAAAD+////AAAAALIAAACzAAAA//////////////////////////////////////
```

```
amavis[31879]: (31879-01) Decoding of p002 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of files
amavis[31879]: (31879-01) Decoding of p002 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of files
amavis[31879]: (31879-01) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at (eval 149) line 861, <GEN39> line 4015.
amavis[31879]: (31879-01) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at (eval 149) line 861, <GEN39> line 4015.
amavis[31879]: (31879-01) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at (eval 149) line 861, <GEN39> line 4015.
amavis[31879]: (31879-01) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at (eval 149) line 861, <GEN39> line 4015.
amavis[15558]: (15558-01-2) Decoding of p039 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of file
amavis[15558]: (15558-01-2) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[15558]: (15558-01-2) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[15102]: (15102-01) Decoding of p039 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of files
amavis[15102]: (15102-01) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[15102]: (15102-01) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[16347]: (16347-01) Decoding of p039 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of files
amavis[16347]: (16347-01) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[16347]: (16347-01) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[15242]: (15242-01) Decoding of p039 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of files
amavis[15242]: (15242-01) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[15242]: (15242-01) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[15004]: (15004-01) Decoding of p039 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of files

amavis[16974]: (16974-01) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[16974]: (16974-01) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[6665]: (06665-01) (!)Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[6665]: (06665-01) Decoding of p002 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Exceeded storage quota 31
amavis[6665]: (06665-01) NOTICE: Virus scanning skipped: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[6665]: (06665-01) (!)NOTICE: HOLD reason: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[8086]: (08086-01) (!)Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[8086]: (08086-01) Decoding of p002 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Exceeded storage quota 31
amavis[8086]: (08086-01) NOTICE: Virus scanning skipped: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[8086]: (08086-01) (!)NOTICE: HOLD reason: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[6561]: (06561-01) (!)Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[6561]: (06561-01) Decoding of p002 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Exceeded storage quota 31
amavis[6561]: (06561-01) NOTICE: Virus scanning skipped: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[6561]: (06561-01) (!)NOTICE: HOLD reason: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[8101]: (08101-01) (!)Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[8101]: (08101-01) Decoding of p002 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Exceeded storage quota 31
amavis[8101]: (08101-01) NOTICE: Virus scanning skipped: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[8101]: (08101-01) (!)NOTICE: HOLD reason: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
```

Vielen Dank, Christoph

So fühlt sich das also in der Realität an

Frontend

rspamd

“rspamd is ja voll toll” kann deutlich mehr als **amavis** und nimmt Peekaboo da auch viel Arbeit ab

rspamd **filtert** durch seinen ganzheitlichen Ansatz viel viel Mist und Malware heraus

Änderungen an Peekaboo

REST API

vorher über **socket** mit zeilenorientiertem Protokoll

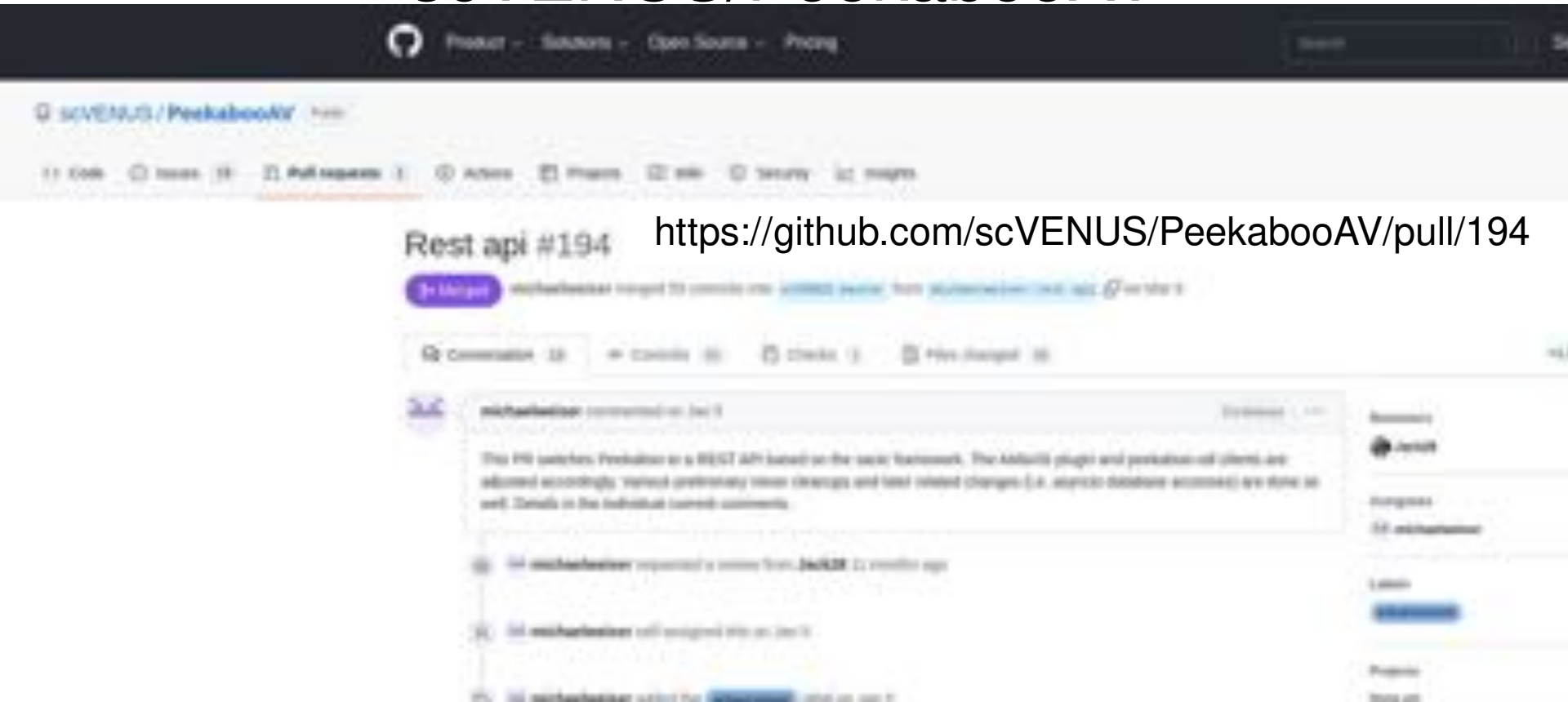
Jetzt **json** über HTTP mit zwei Endpunkten:

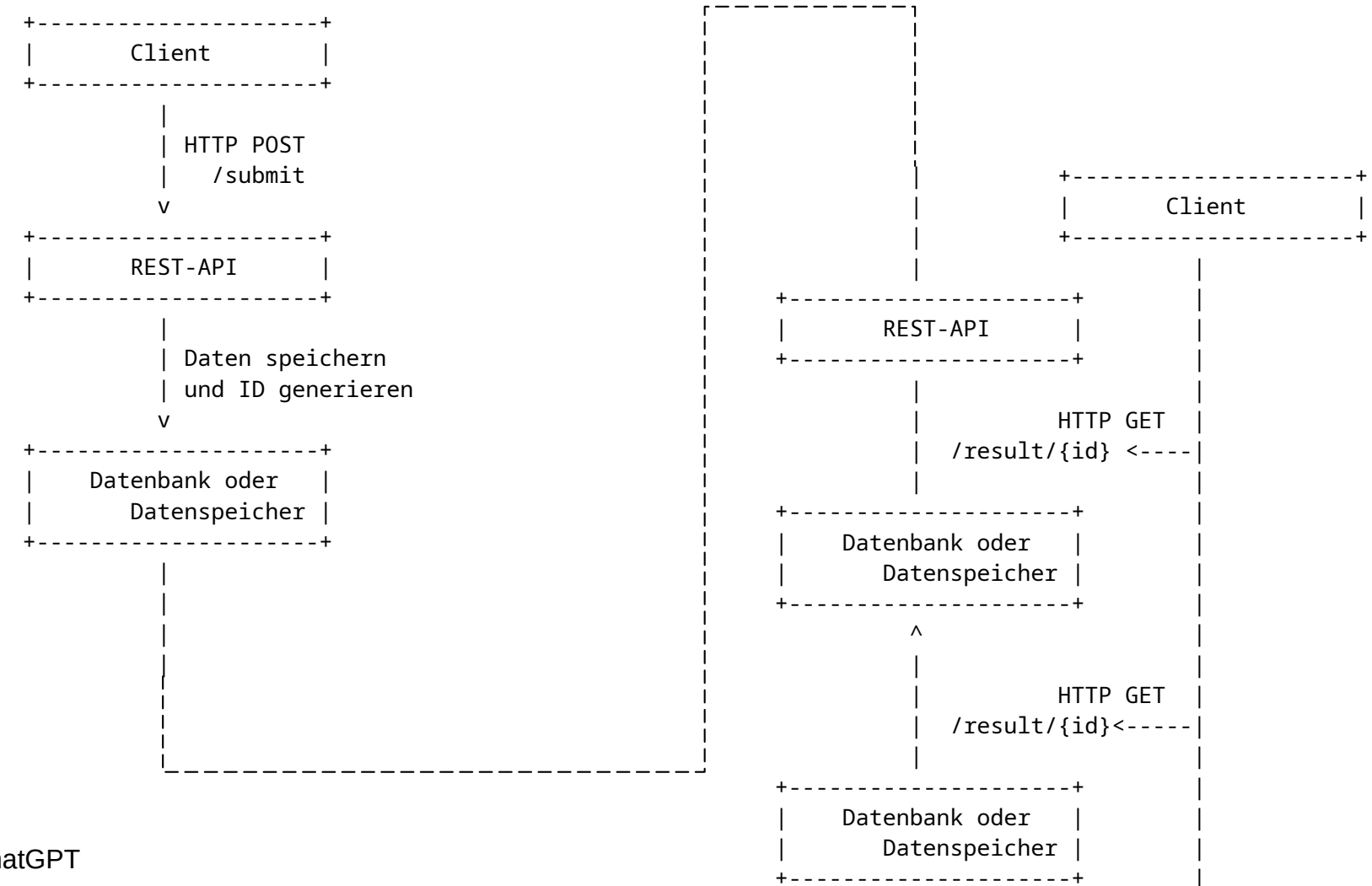
Submit und polling des **Result**ats.

+Implementierung für Amavis

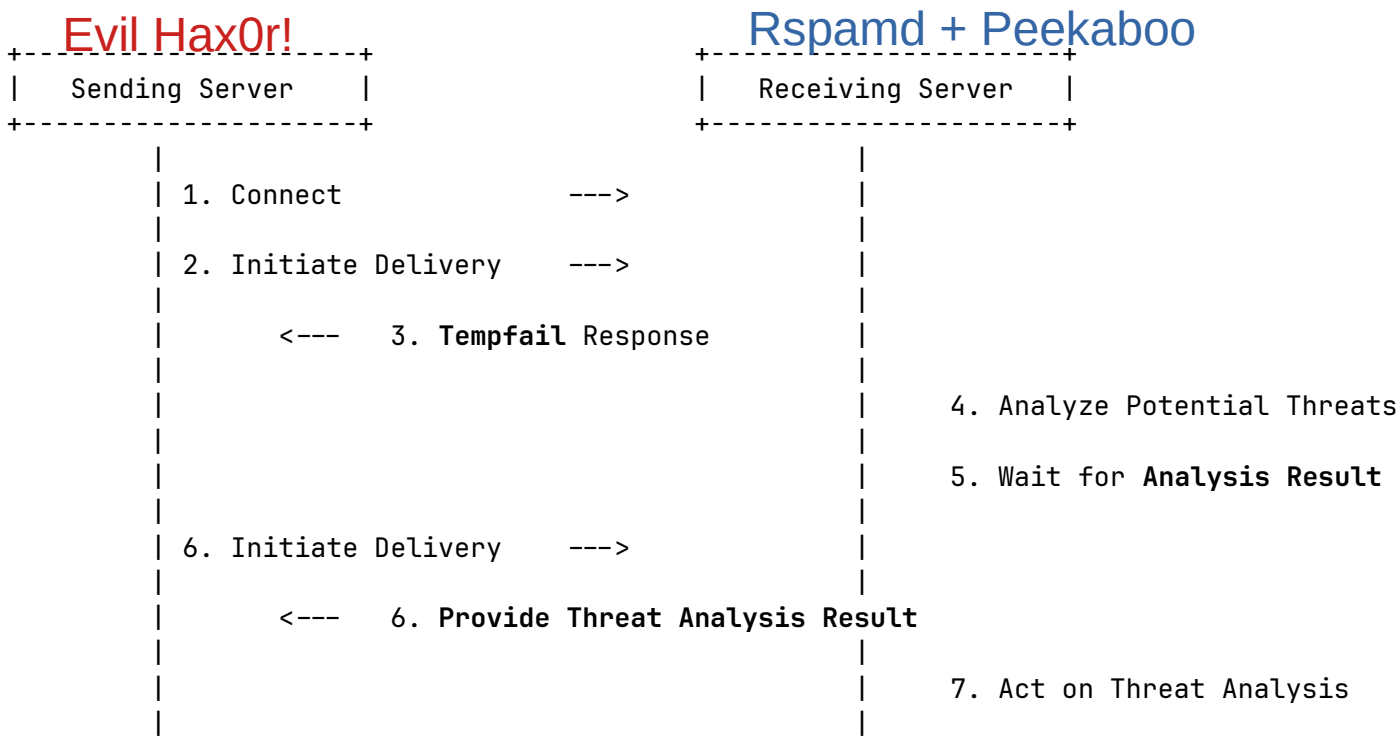
Carsten Rosenberg von heinlein **Support hat das Plugin für Rspamd geschrieben DANKE**

Rest api · Pull Request #194 · scVENUS/PeekabooAV





Quelle: ChatGPT



Quelle: ChatGPT

expander

Im Unterschied zu Amavis packt rspamd Archive nicht aus

https://github.com/science-computing/expander

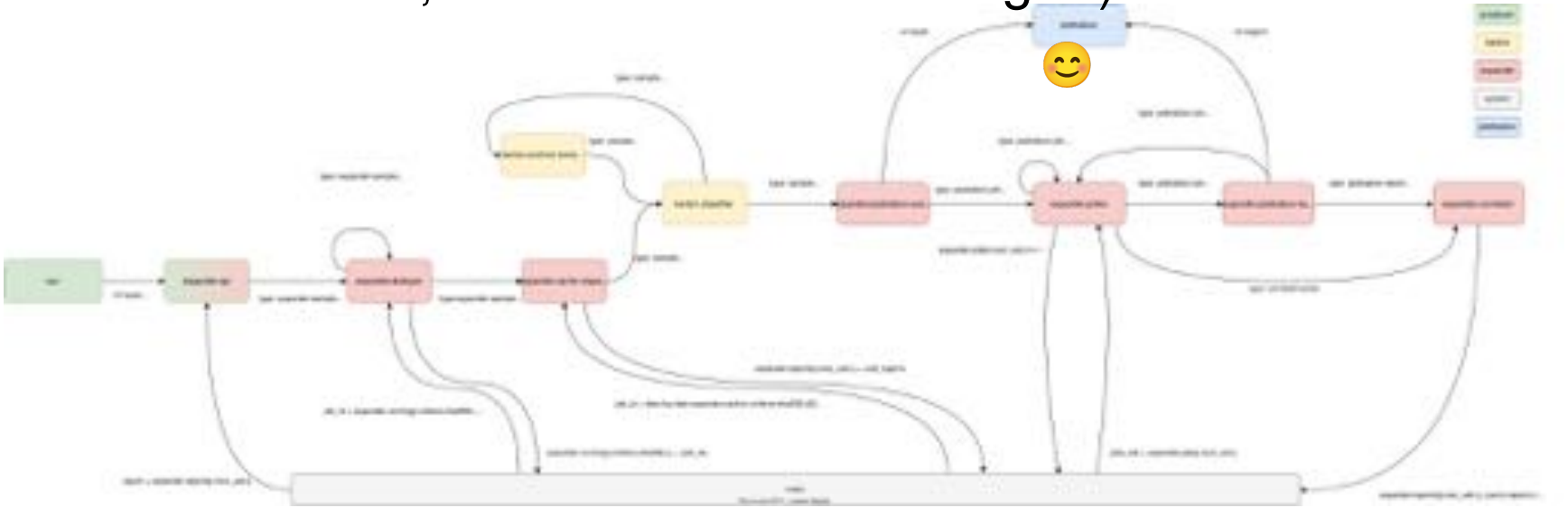
Expander Logo

Powered by:
CERT-Polska

Karton

„Distributed malware processing framework based on Python, Redis and S3.“

(**Schaubild vom Expander**, Karton Mechanik **ausgereizt**. Dabei belassen, Publikum nicht abhängen)



Expander Karton Schematische Darstellung

expression rules

- Flexibilität
- kein Scoring

```
expression.3  : sample.name_declared == 'signature.asc'
                and sample.type_declared in {
                    'application/pgp-signature'
                } -> ignore
expression.4  : sample.file_extension in {
                    'doc', 'docm', 'dotm', 'docx', 'rtf', 'rtx',
                    'ppt', 'pptm', 'pptx', 'potm', 'ppam', 'ppsm',
                    'xls', 'xlsm', 'xlsx' }
                and olereport.has_office_macros == True
                and cuckooreport.score > 4 -> bad
#expression.5  : cortexreport.VirusTotalQueryReport.n_of_all == 0
#                 and cortexreport.VirusTotalQueryReport.level == 'safe'
#                 ->  unknown
```

- aufpassen, den statischen Virenscanner nicht ersetzen
- schöne Möglichkeit verschiedene Reports/Analyzer miteinander zu verknüpfen

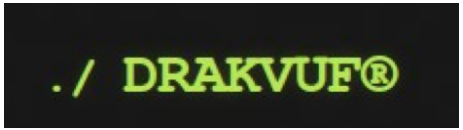
Beispiele: - von “alles in die Sandbox” ... keine **verschlüsselten Mails** … viele **ZIP basierte Dateiformate** (Numbers, Pages … Adobe …)

Backend

cortex

- Meta (multi) Analyzer
- Eine Anbindung -> Universum von Analyzern
- -> MISP -> Mitre Datenbanken
- -> Automatisierung -> Cases -> Profit/Win

https://drakvuf.com (GPL License)
by Tamas K Lengyel



drakvuf

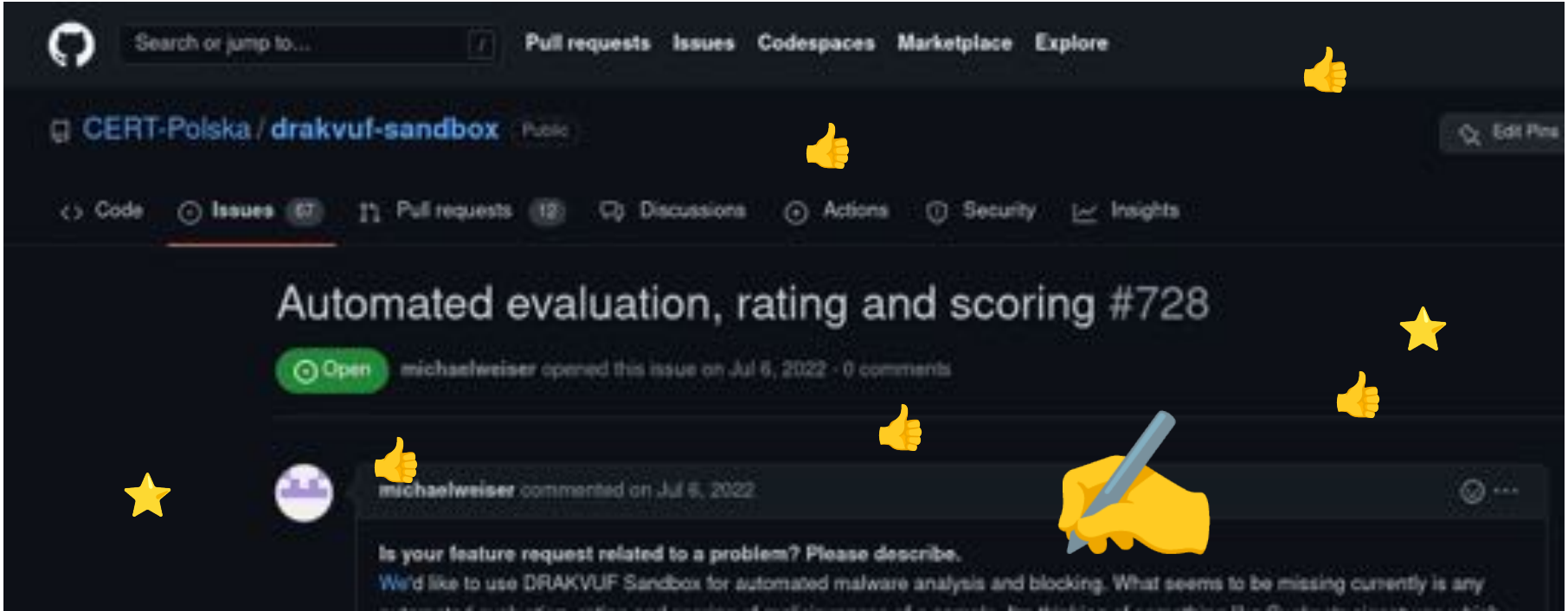
Der Ansatz am **Hypervisor** Daten zu sammeln. Ist technisch sehr interessant. Verwenden auch das Karton Framework. Skalierbar, Flexibel. **Keine Signature Engine**, es wird nur eine Liste von Aktionen protokolliert, aber keine Bewertung! **Idee!** Man könnte die **Cuckoo Signaturen** portieren und als Karton in die Pipeline hängen

-> 5 abstrakte Signaturen anstatt 500 MB json

Drakvuf-Sandbox by CERT.PL

Automated evaluation, rating and scoring · Issue #728 · CERT-Polska/drakvuf-sandbox

https://github.com/CERT-Polska/drakvuf-sandbox/issues/728



cuckoo 3

- Gemacht für **eine!!1!** Version von Windows 10
- Der Agent wurde zum Treiber

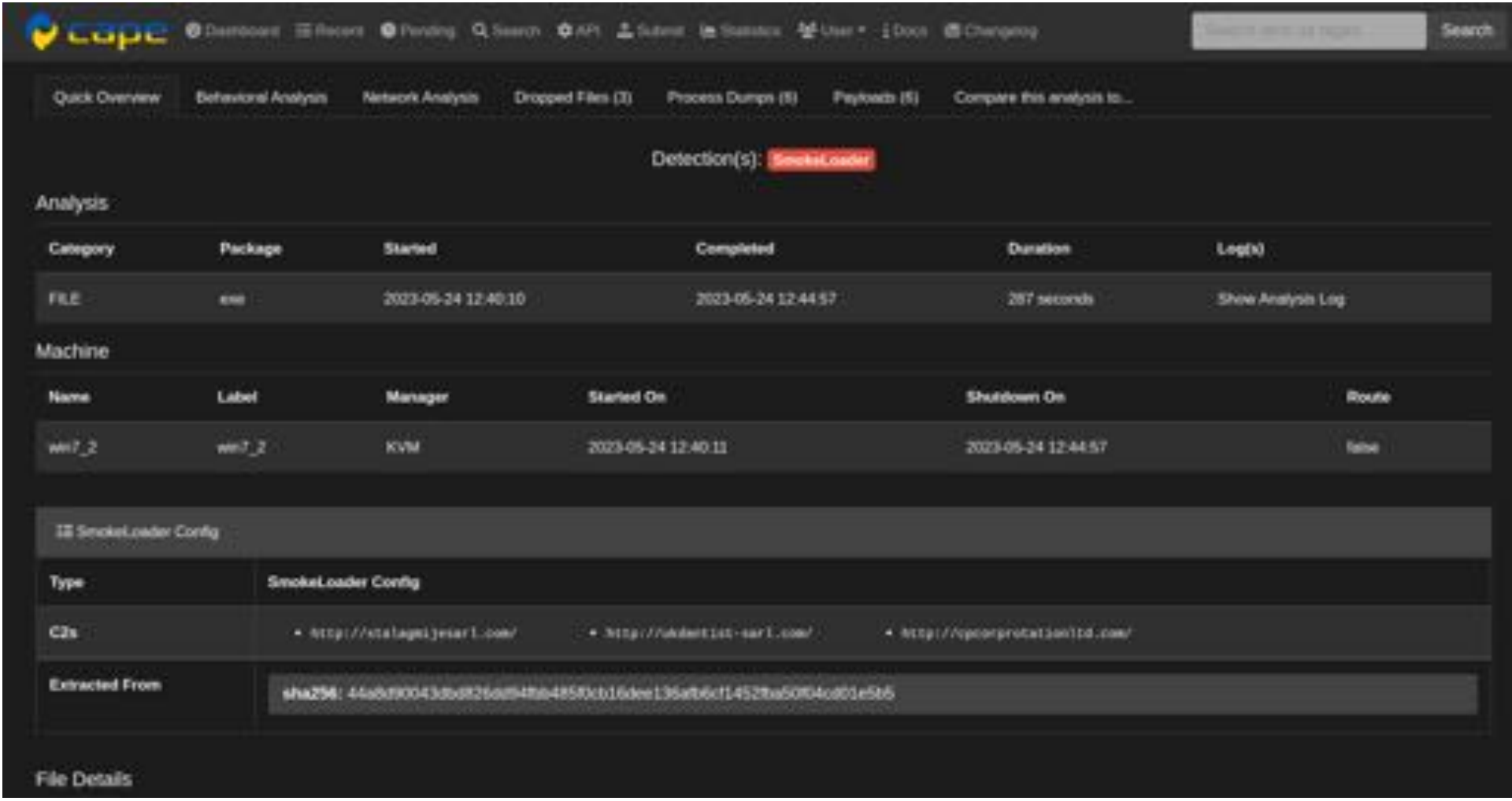
- **Eine** Veröffentlichung
- Hat immer noch nicht richtig abgehoben

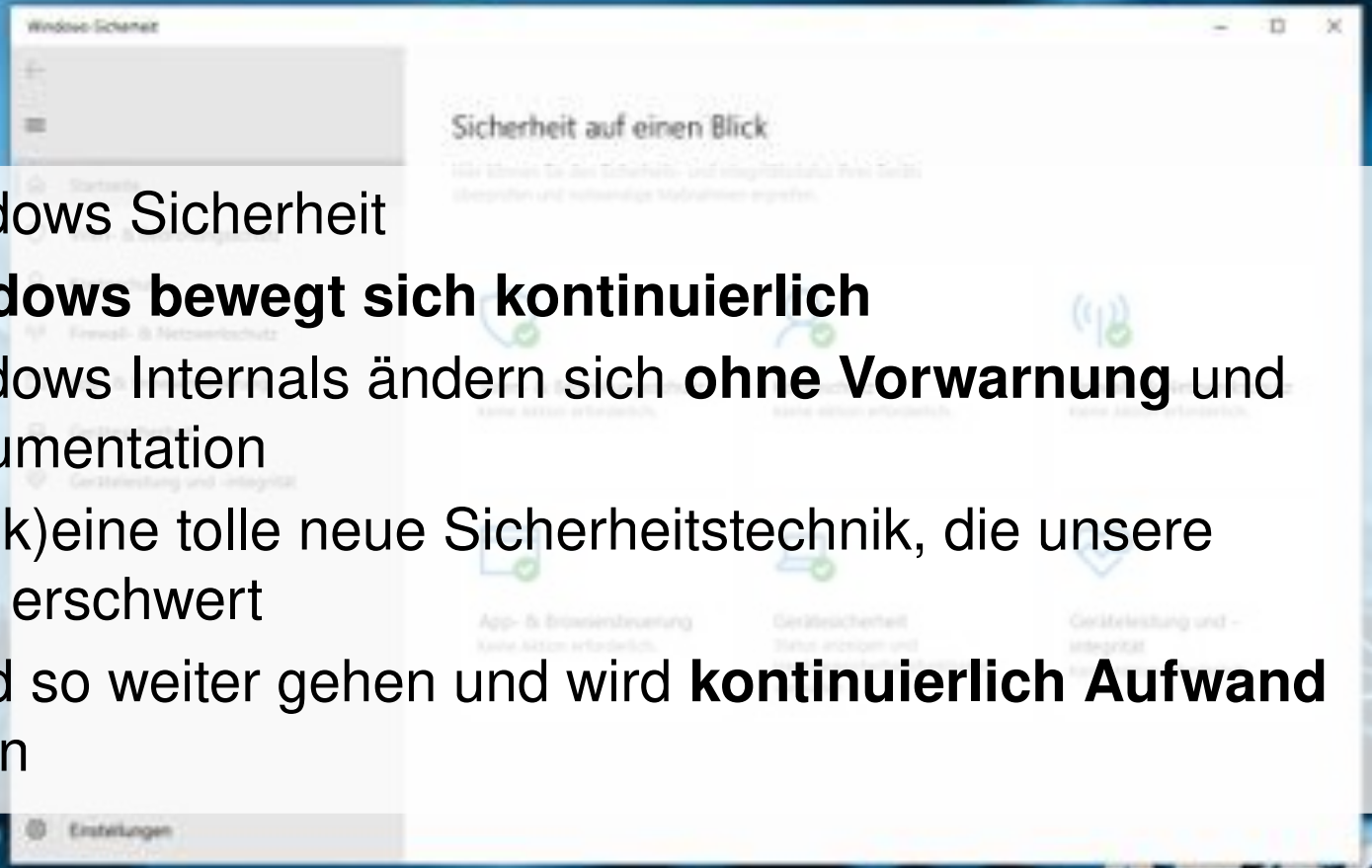
cape by

CAPEv2

- vielversprechendste Kandidatin
- Fork von cuckoo 2
- Python3
- für die interaktive Malware-Analyse, Payload Extraction,
- eher unstable

Wir bräuchten für die Automatisierung Stable Releases, das passt aber nicht zu Windows 10.

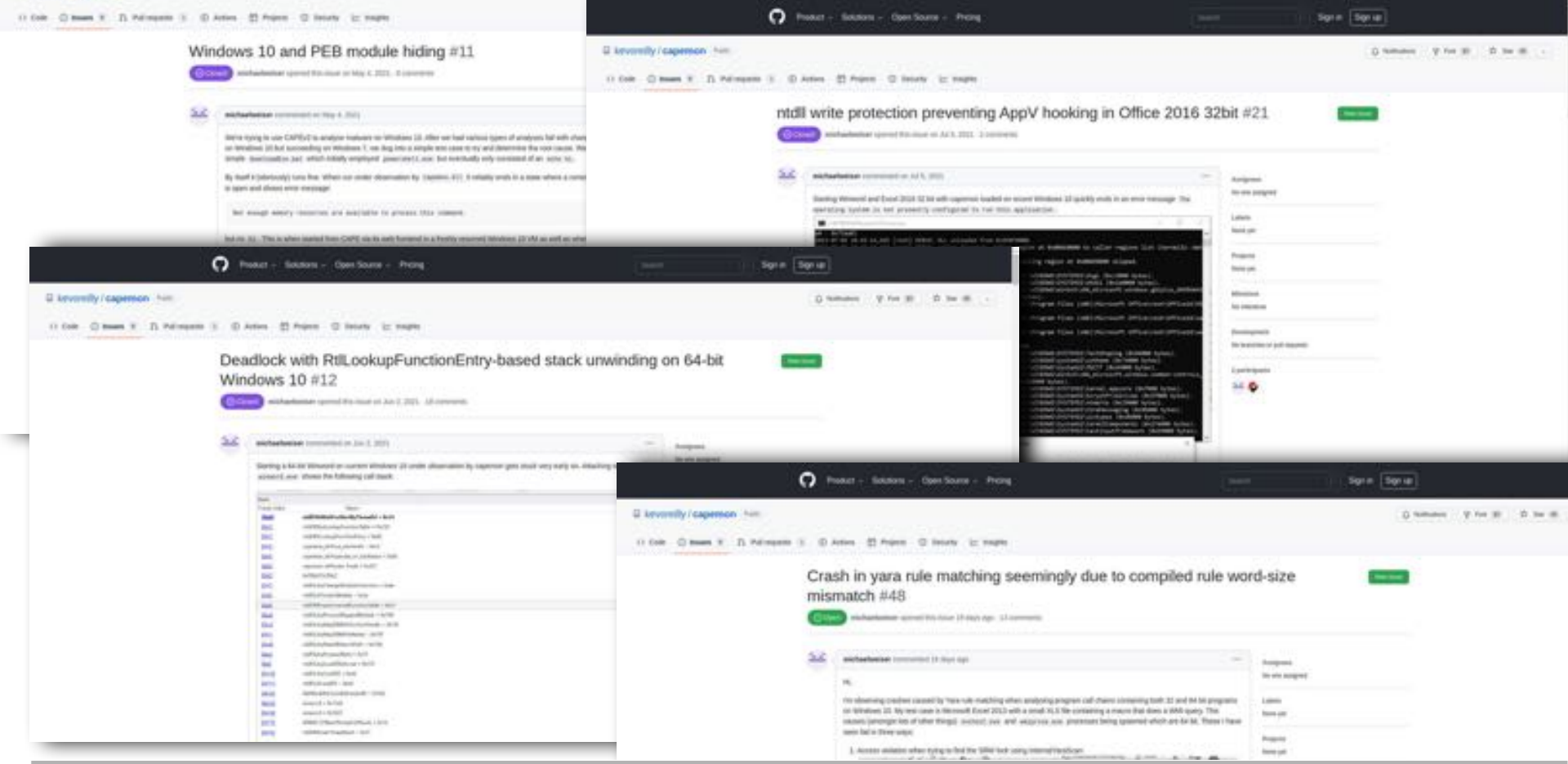

cape
Dashboard
Recent
Pending
Search
API
Submit
Statistics
User
Docs
Changelog
Search
Quick Overview
Behavioral Analysis
Network Analysis
Dropped Files (3)
Process Dumps (6)
Payloads (6)
Compare this analysis to...
Detection(s): SmokeLoader
Analysis
Category
Package
Started
Completed
Duration
Log(s)
FILE
exe
2023-05-24 12:40:10
2023-05-24 12:44:57
287 seconds
Show Analysis Log
Machine
Name
Label
Manager
Started On
Shutdown On
Route
win7_2
win7_2
KVM
2023-05-24 12:40:11
2023-05-24 12:44:57
false
SmokeLoader Config
Type
SmokeLoader Config
C2s
http://xtalagmijesarl.com/
http://ukdentist-sarl.com/
http://cpcorporationltd.com/
Extracted From
sha256: 44a8d90043dbd826a8f4ffbb48580cb16dee136afb6cf1452fba50f04cd01e5b6
File Details



- Windows Sicherheit
- **Windows bewegt sich kontinuierlich**
- Windows Internals ändern sich **ohne Vorwarnung** und Dokumentation

Es gibt (k)eine tolle neue Sicherheitstechnik, die unsere Analyse erschwert

Das wird so weiter gehen und wird **kontinuierlich Aufwand** erzeugen



https://github.com/kevoreilly/capemon/issues?q=is%3Aissue+author%3Amichaelweiser

Wie funktioniert CAPE, Windows, Office? Und **wie lange?**

- Mit **Windows 7** ist die Erkennung auch gut (kauft aber keiner)
- **Windows 10** bewegt sich zu schnell

Sandbox	Windows 10?	Zukunftsfähig ?	Todos
Cuckoo 2	nein	nein	Python3 ...
drakvuf	ja	jein	Signaturen
Cuckoo 3	eine Version	(noch) nicht	Windows
CAPEv2	jein	am ehesten	Windows

Tabelle, Gegenüberstellung der Optionen

Wir warten noch auf die Gewinnerin

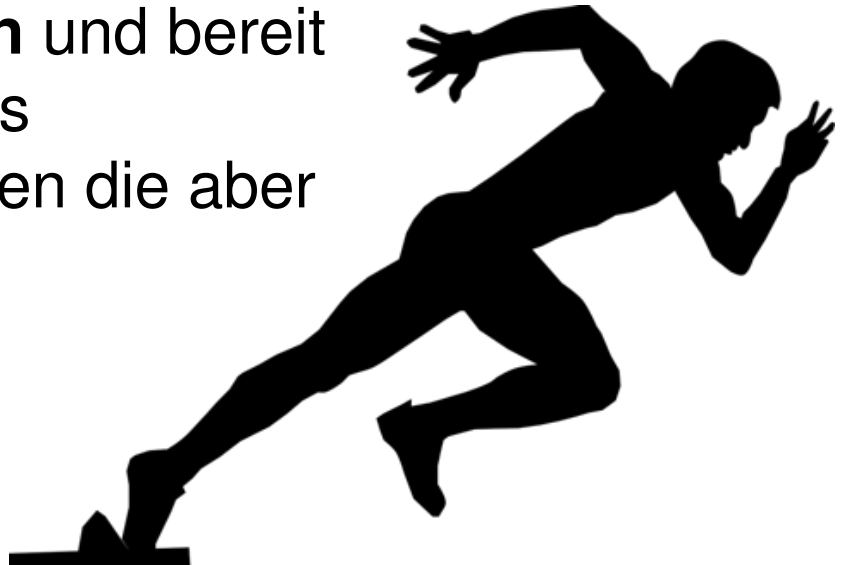
Nach Emotet und Konsorten in neuer Form und Stärke

Wie müssen sich Analyse-Sandboxen weiterentwickeln um mit der Malware mithalten zu können?

- Die User:inneninteraktion ist zur Erkennung notwendig
- Das Befolgen von in der Mail enthaltenen Instruktionen
- KI Usersimulation

Auf die Plätze! Fertig! ??

- Soweit so **gut**
- Die Hoffnung ist, Ihr konntet was **lernen**
- Gerne **Feedback, Ideen, Anregungen** …
- Wir stehen in den **Startlöchern** und bereit für den Tag, an dem eine neues **Sandboxlösung** kommt (können die aber **alleine nicht** stemmen)



Vielen Dank für Eure Aufmerksamkeit

Kommt gerne auf uns zu:

- Christoph, Felix, Michael
- @peekabooav@fosstodon.org
- info@peekabooav.de

SLAC 2023

23.–25. Mai 2023 | Berlin

Interna

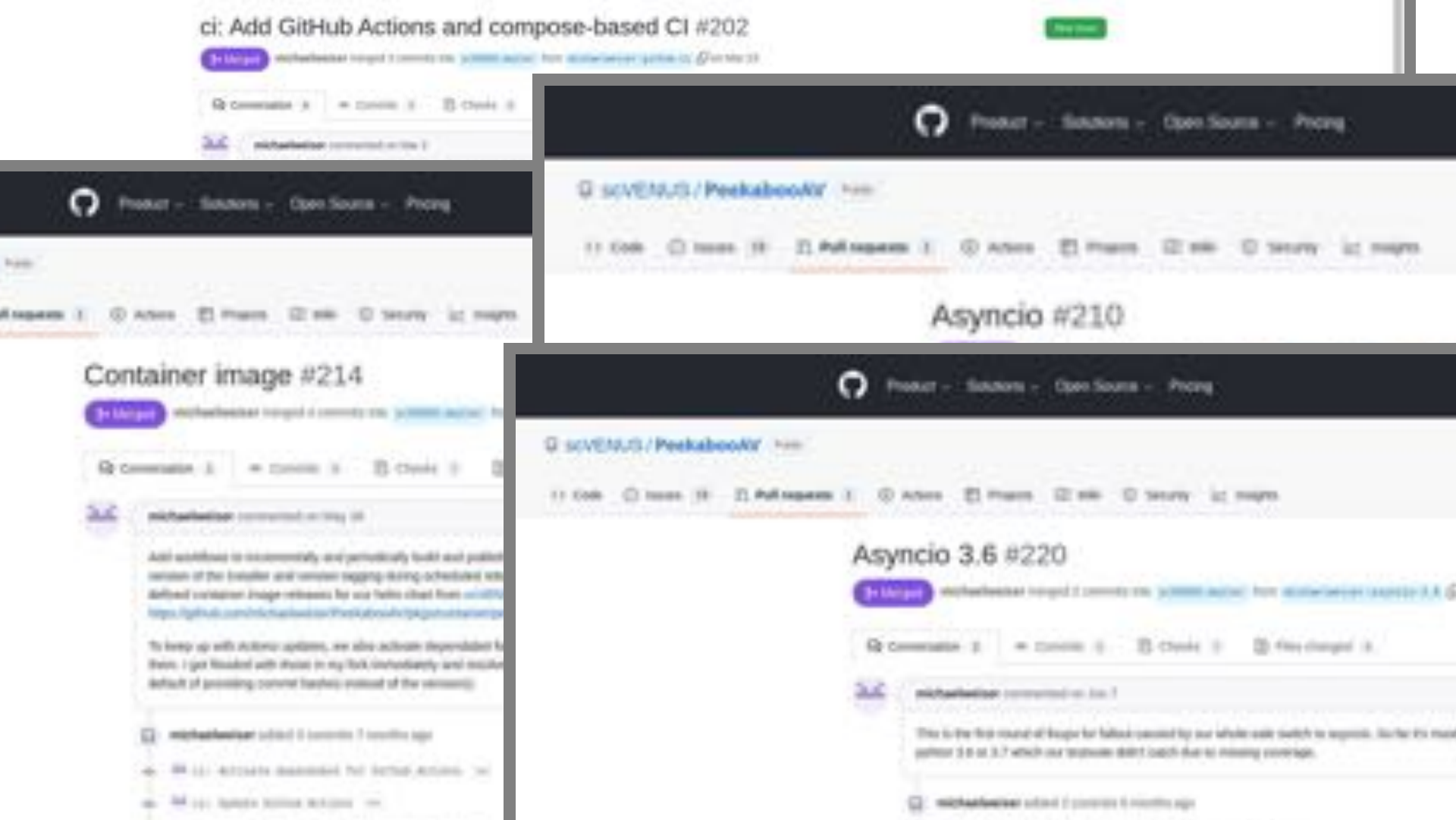
“übrigens”

asyncio

container

kubernetes. HELM

ci/cd


ci: Add GitHub Actions and compose-based CI #202
Product
Solutions
Open Source
Pricing
scVENUS / PeekabooAV
Asyncio #210
Container image #214
Asyncio 3.6 #220

SLAC 2023
23.–25. Mai 2023 | Berlin

PeekabooAV - Komponenten und Erfahrungen der OpenSource Sandbox-Analyse

Felix Bauer
Christoph Herrmann
Michael Weiser

24.05.2023

Am heutigen 24.05.2023 halten wir diesen Vortrag auf der Secure Linux Administration Conference in Berlin.

Vielen Dank an Heinlein Support für die Kooperation und den angeregten Austausch.

Gestern 6 Jahre PeekabooAV

Vor **4 Jahren** haben wir auf der **SLAC** zum zweiten Mal von PeekabooAV berichtet, wie E-Mailanhänge auf Schadsoftware geprüft werden können. **Heute** stellen wir die **gesamte Analyse-Pipeline** vor, von der Schnittstelle ins Mailsystem (**rspamd**), über das Entpacken (**expander**) bis zur Ausführung in der Sandbox mit **Windows 10**.

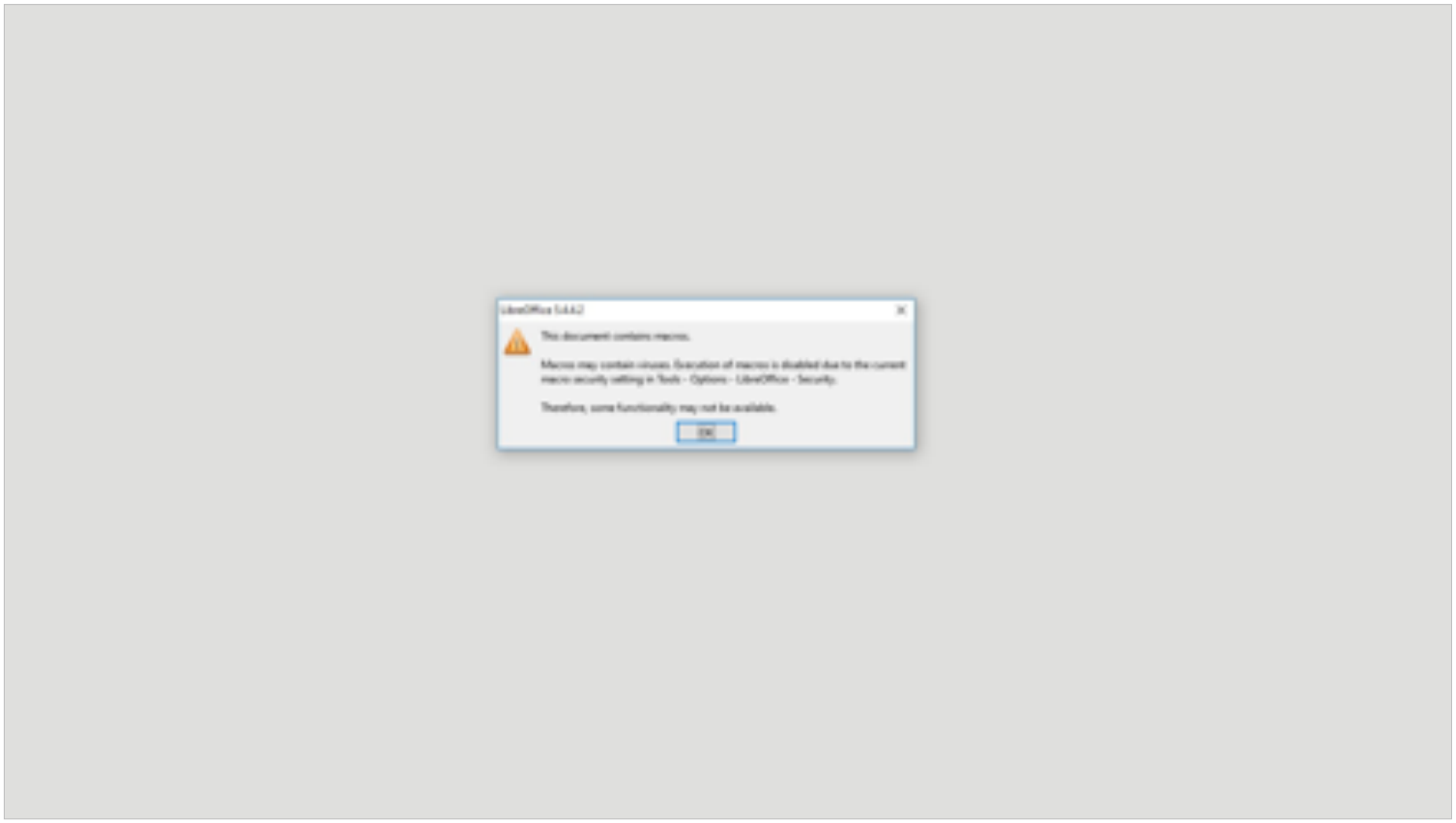
Wie es dazu kam, **Stand und Zukunft**!

Cuckoo ist tot, lang lebe **Cuckoo 3**, **CAPEv2** und **Drakvuf**, **Cortex** und **Karton**.

Wir sind zum dritten Mal hier und freuen uns sehr.

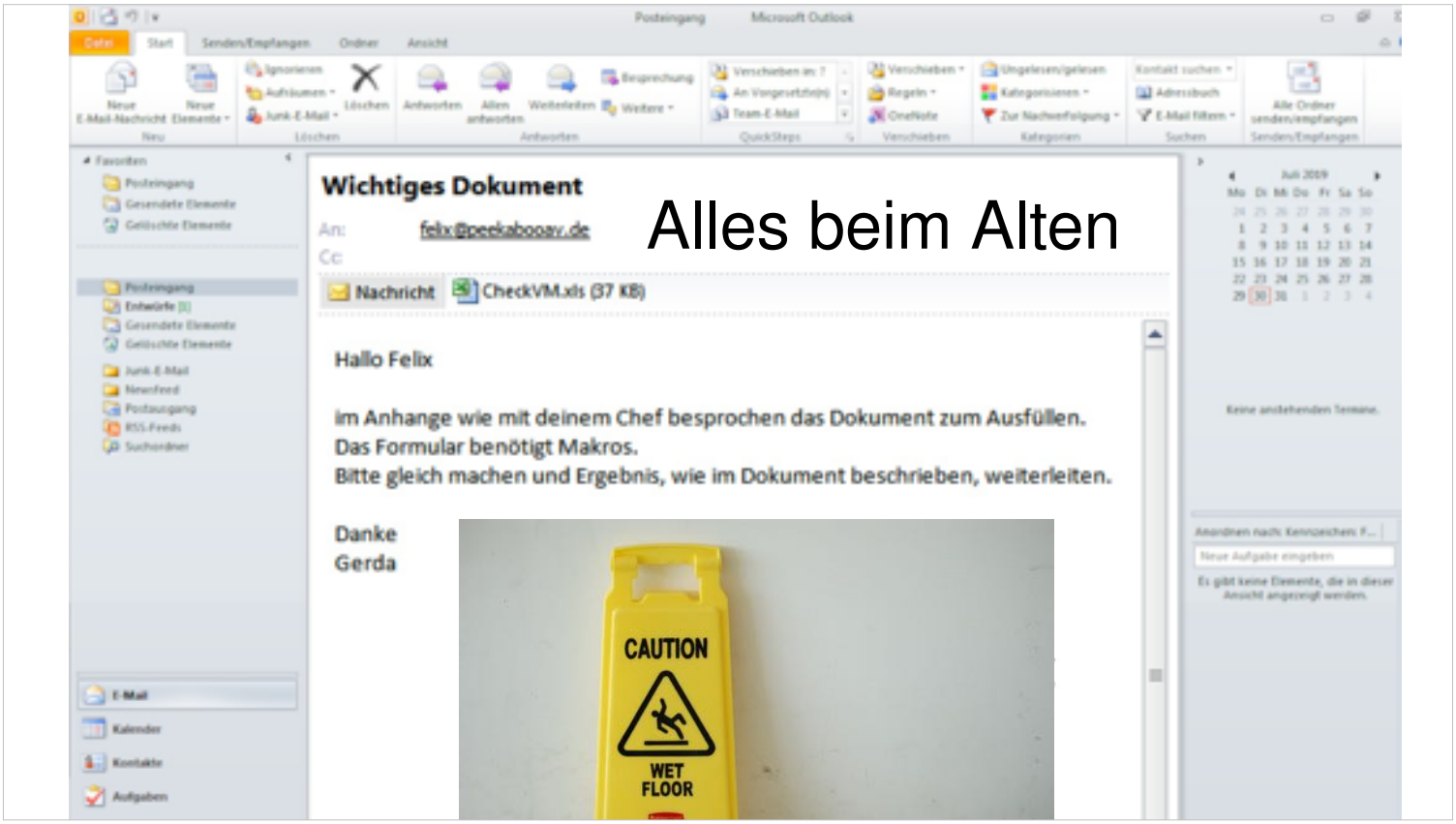
Schön, dass so viele gekommen sind.

2017 auf der SLAC haben wir unseren Open Source release gemacht, im Jahr darauf von den Entwicklungen und Erkenntnissen berichtet
und jetzt wollen wir zeigen was die letzten 4 Jahre gebracht haben.



Manche Dinge haben sich nicht verändert.

Trotz, dass Makros jetzt per Gruppenrichtlinie deaktiviert werden können ist es nach wie vor ein gern genutztes Feature für Anwender:innen und Angreifer:innen

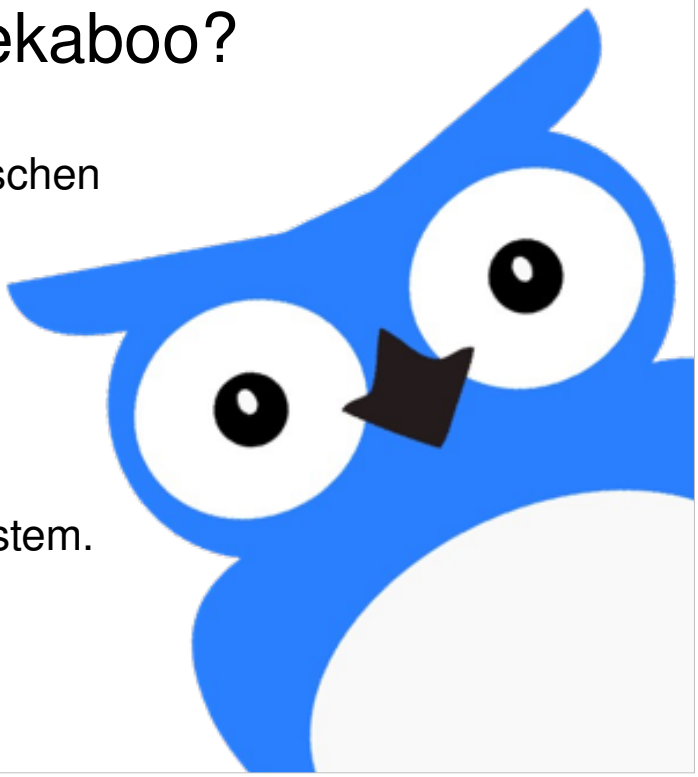


Die Tricks werden ausgefuchster und immer schwerer zu erkennen. Die Malware antwortet auf bestehende Konversationen und macht es so leicht dem schädlichen Anhang zu vertrauen.

Was ist Peekaboo?

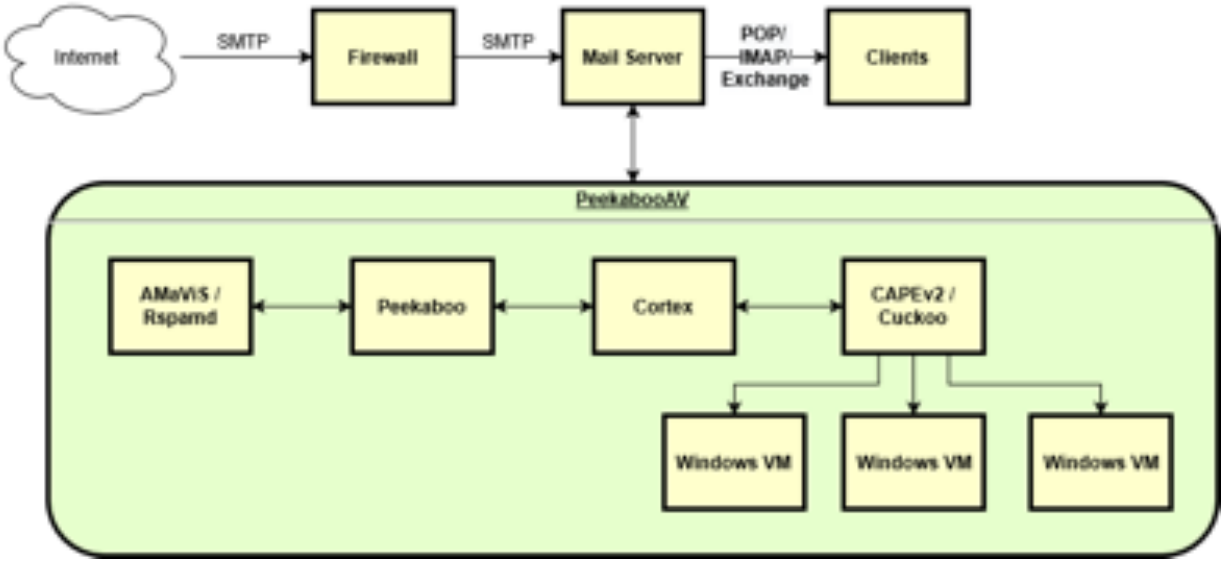
Peekaboo ist das **Bindeglied** zwischen dem **Mailsystem** und der **Sandboxanalyse**!

Peekaboo gibt die **Anhänge** zur Analyse, wertet aus und meldet die **Entscheidung** zurück ins Mailsystem.



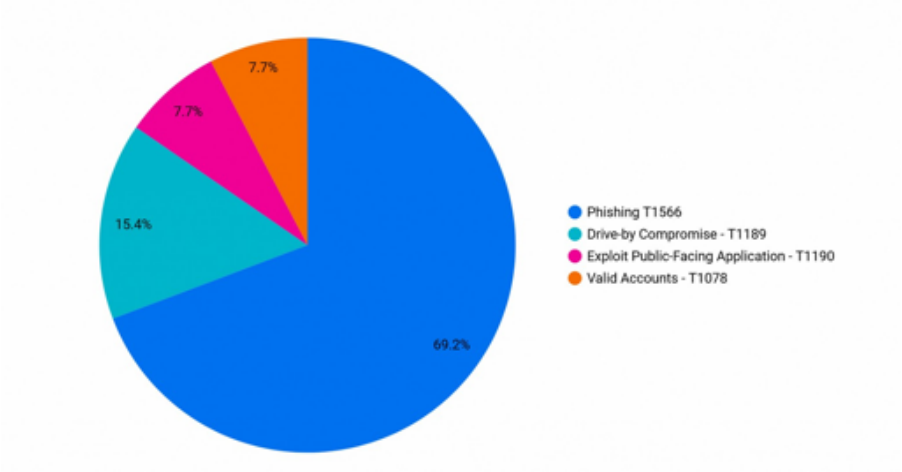
Peekaboo ist eine Serveranwendung welche aus dem Mailsystem heraus Anhänge analysiert.
Dazu verwenden wir ein sehr mächtiges Regelwerk, das wir später noch einmal sehen werden.
Der Kern von PeekabooAV ist die Verhaltensanalyse in der Sandbox. Der Anhang wird in einer VM geöffnet und deren Verhalten/Aktionen protokolliert.
Die gesammelten Reports der Analyzer (die Sandbox ist nur einer) führen dann zum Ergebnis, dass die Datei schädlich ist, oder dass keine Schadroutine erkannt werden konnte und die Mail zugestellt werden kann.

Peekaboo im Mailsystem



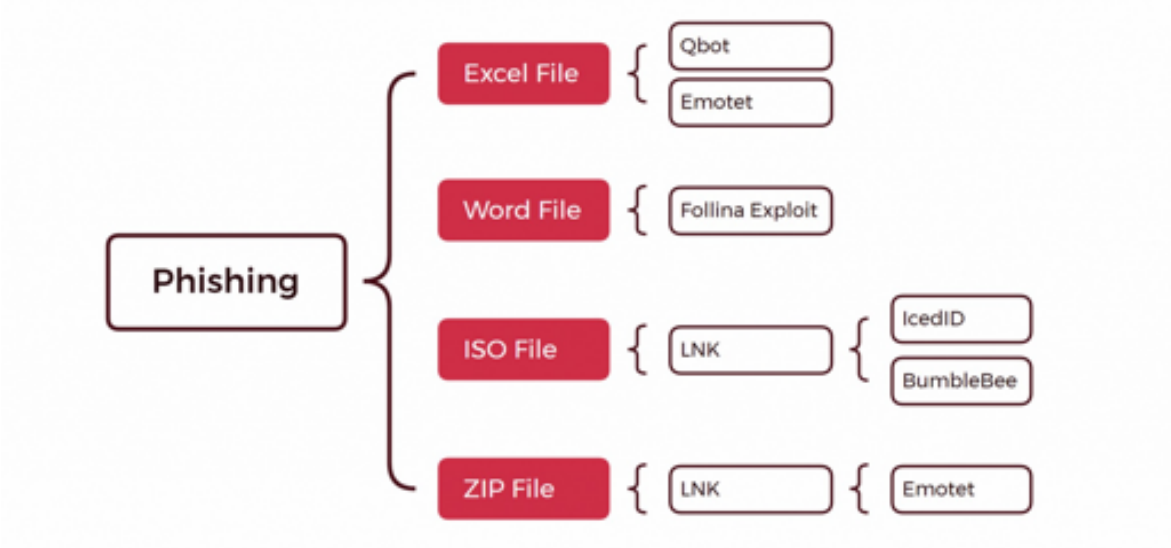
PeekabooAV ist sehr sehr flexibel und lässt sich an alle Gegebenheiten und Bedürfnisse anpassen.
Hier eine schematische Darstellung der Anbindung von PeekabooAV und seinen Komponenten in die eMail-Umgebung.

The DFIR report 2022 Year in Review - Initial Access



https://thedfirreport.com/2023/03/06/2022-year-in-review/

Die Wenigsten sollte es überraschen, das Phishing nach wie vor Nummer eins für die initiale Kompromittierung eines Unternehmens ist.



https://thedfirreport.com/2023/03/06/2022-year-in-review/

Die schädlichen Anhänge sind oft Microsoft Office Dateien mit Malware wie Qbot, Emotet oder Follina.
Neuer ist, dass auch ISO-Dateien (normalerweise für CD/DVD images) verwendet werden um die Malware einzuschleusen. Microsoft Windows erlaubt das Einbinden und damit den Zugriff auf diese Dateien als virtuelle CD-ROM-Laufwerke.
ZIP-Dateien werden häufig passwortverschlüsselt mit Instruktionen in der eMail.

Eine weitere Kapselung konnte beobachtet werden, wo sich eine HTML-Datei im Anhang befindet, welche mittels Java-Script eine enthaltene Zeichenkette dekodiert und als ZIP zum Download anbietet.

Zwei Folien von **Christoph**

Bericht aus der **Praxis**

Kuriositäten und Realitätsabgleich

Christoph betriebt unsere beiden größten Kundenumgebungen.

„Die Systeme laufen ohne größere Aufwände ruhig vor sich hin, der Ressourcenaufwand hält sich in Grenzen und die Ergebnisse sind gut“

„Aufwändig ist die Installation und Anpassung an die Gegebenheiten, um individuell bestmögliche Ergebnisse zu erzielen“

```
To: "" <Hans.Hoffmann@domain.de>
Subject: MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----------------JHPn4GLt5KrvuXRamCO6TZy8"


This is a multi-part message in MIME format.
------------------JHPn4GLt5KrvuXRamCO6TZy8
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: quoted-printable


<html>
<head>
<meta http-equiv=3DContent-Type content=3D"text/html; charset=3Diso-8859-1"=
>
</head>
<body>
<br>

</body>
</html>
------------------JHPn4GLt5KrvuXRamCO6TZy8
Content-Type: application/vnd.ms-excel;
 name="IhreRechnung.xls"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="IhreRechnung.xls"


0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAEAAAA/AEAAAAAAAAA
...
```

Hier ein Beispiel einer Mail, die so empfangen wurde.

Es ist gut zu erkennen, dass sich die Mail nicht an das normale Format hält, diverse Header fehlen und plump eine Datei „IhreRechnung.xls“ angehängt wurde.

Außerdem fällt auf, dass es sich um das alte Format von Microsoft Office Excel hält (die Dateiendung wäre sonst xlsx bzw. um Makros zu erlauben xlsm)

```
Subject: =?UTF-8?B?UmU6IEplbm55IEtsZWluZW1leWVyIC0gQsO8cm8gU2lsdmlhIEJyZWg=?=
 MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----------------xF06JPJRGLwHam8aTrmtfDHD"

This is a multi-part message in MIME format.
------------------xF06JPJRGLwHam8aTrmtfDHD
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: quoted-printable

<html>
<head>
<meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3Diso-2022-=
jp">
</head>
<body>
=0DAls Anhang erhalten Sie Ihre Rechnung.<br>
<br>
<br>
<br>

Organisation<br>
Mail <a href=3D"mailto:info@domain.de">info@domain.de</a><br>
<a href=3D"http://www.domain.de">www.domain.de</a>
</body>
</html>
------------------xF06JPJRGLwHam8aTrmtfDHD
Content-Type: application/vnd.ms-excel;
 name="Scan 2022.11.11_1346.xls"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="Scan 2022.11.11_1346.xls"

0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAACAAAAtAAAAAAAAAAAA
EAAA/v///wAAAAD+////AAAAALIAAACzAAAA//////////////////////////////////////
```

Ein weiteres Beispiel einer Phishing Mail, diesmal mit einem Anhang: Scan 2022.11.11_1346.xls

```
amavis[31879]: (31879-01) Decoding of p002 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of files
amavis[31879]: (31879-01) Decoding of p002 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of files
amavis[31879]: (31879-01) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at (eval 149) line 861, <GEN39> line 4015.
amavis[31879]: (31879-01) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at (eval 149) line 861, <GEN39> line 4015.
amavis[31879]: (31879-01) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at (eval 149) line 861, <GEN39> line 4015.
amavis[31879]: (31879-01) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at (eval 149) line 861, <GEN39> line 4015.
amavis[15558]: (15558-01-2) Decoding of p039 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of file
amavis[15558]: (15558-01-2) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[15558]: (15558-01-2) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[15102]: (15102-01) Decoding of p039 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of files
amavis[15102]: (15102-01) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[15102]: (15102-01) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[16347]: (16347-01) Decoding of p039 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of files
amavis[16347]: (16347-01) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[16347]: (16347-01) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[15242]: (15242-01) Decoding of p039 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of files
amavis[15242]: (15242-01) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[15242]: (15242-01) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[15004]: (15004-01) Decoding of p039 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Maximum number of files

amavis[16974]: (16974-01) NOTICE: Virus scanning skipped: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[16974]: (16974-01) (!)NOTICE: HOLD reason: do_7zip: Maximum number of files (250) exceeded at /usr/sbin/amavisd-new line 9843.
amavis[6665]: (06665-01) (!)Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[6665]: (06665-01) Decoding of p002 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Exceeded storage quota 31
amavis[6665]: (06665-01) NOTICE: Virus scanning skipped: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[6665]: (06665-01) (!)NOTICE: HOLD reason: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[8086]: (08086-01) (!)Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[8086]: (08086-01) Decoding of p002 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Exceeded storage quota 31
amavis[8086]: (08086-01) NOTICE: Virus scanning skipped: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[8086]: (08086-01) (!)NOTICE: HOLD reason: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[6561]: (06561-01) (!)Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[6561]: (06561-01) Decoding of p002 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Exceeded storage quota 31
amavis[6561]: (06561-01) NOTICE: Virus scanning skipped: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[6561]: (06561-01) (!)NOTICE: HOLD reason: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[8101]: (08101-01) (!)Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[8101]: (08101-01) Decoding of p002 (Zip archive data, at least v2.0 to extract) failed, leaving it unpacked: do_7zip: Exceeded storage quota 31
amavis[8101]: (08101-01) NOTICE: Virus scanning skipped: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
amavis[8101]: (08101-01) (!)NOTICE: HOLD reason: do_7zip: Exceeded storage quota 314572800 bytes by do_7zip-pre; last chunk 368865792 bytes
```

Wie leicht zu erkennen ist haben wir es hier mit einer ZIP-Datei im Anhang zu tun welche (in der oberen Hälfte) sehr sehr viele Dateien beinhaltet und deswegen nicht komplett extrahiert werden kann

und unten, einer ZIP-Datei dessen Inhalt sich zu mehreren hundert Megabyte extrahiert und deswegen der Vorgang abgebrochen wird.

Das sind Vorgehen der Angreifer:innen um den Analysevorgang zu umgehen.

Vielen Dank, Christoph

So fühlt sich das also in der Realität an

Frontend

rspamd

“rspamd is ja voll toll” kann deutlich mehr als **amavis** und nimmt Peekaboo da auch viel Arbeit ab

rspamd **filtert** durch seinen ganzheitlichen Ansatz viel viel Mist und Malware heraus

Bis letztes Jahr hatten wir ausschließlich Amavis im Einsatz um Peekaboo ins Mailsystem einzubinden.

Rspamd wird immer mehr zum Standard und auch unsere Kunden sehen die Stärken von rspamd und möchten davon profitieren.

Änderungen an Peekaboo

REST API

vorher über **socket** mit zeilenorientiertem Protokoll

Jetzt **json** über HTTP mit zwei Endpunkten:

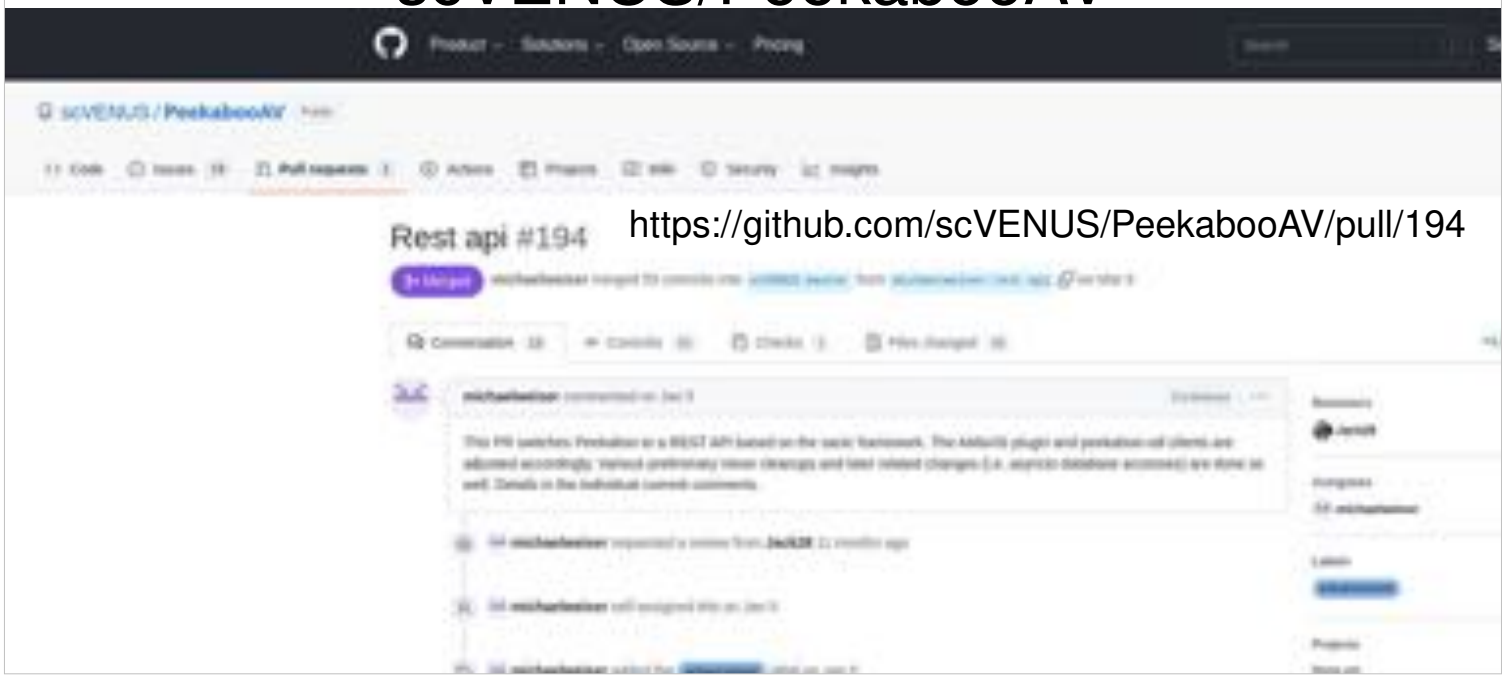
Submit und polling des **Result**ats.

+Implementierung für Amavis

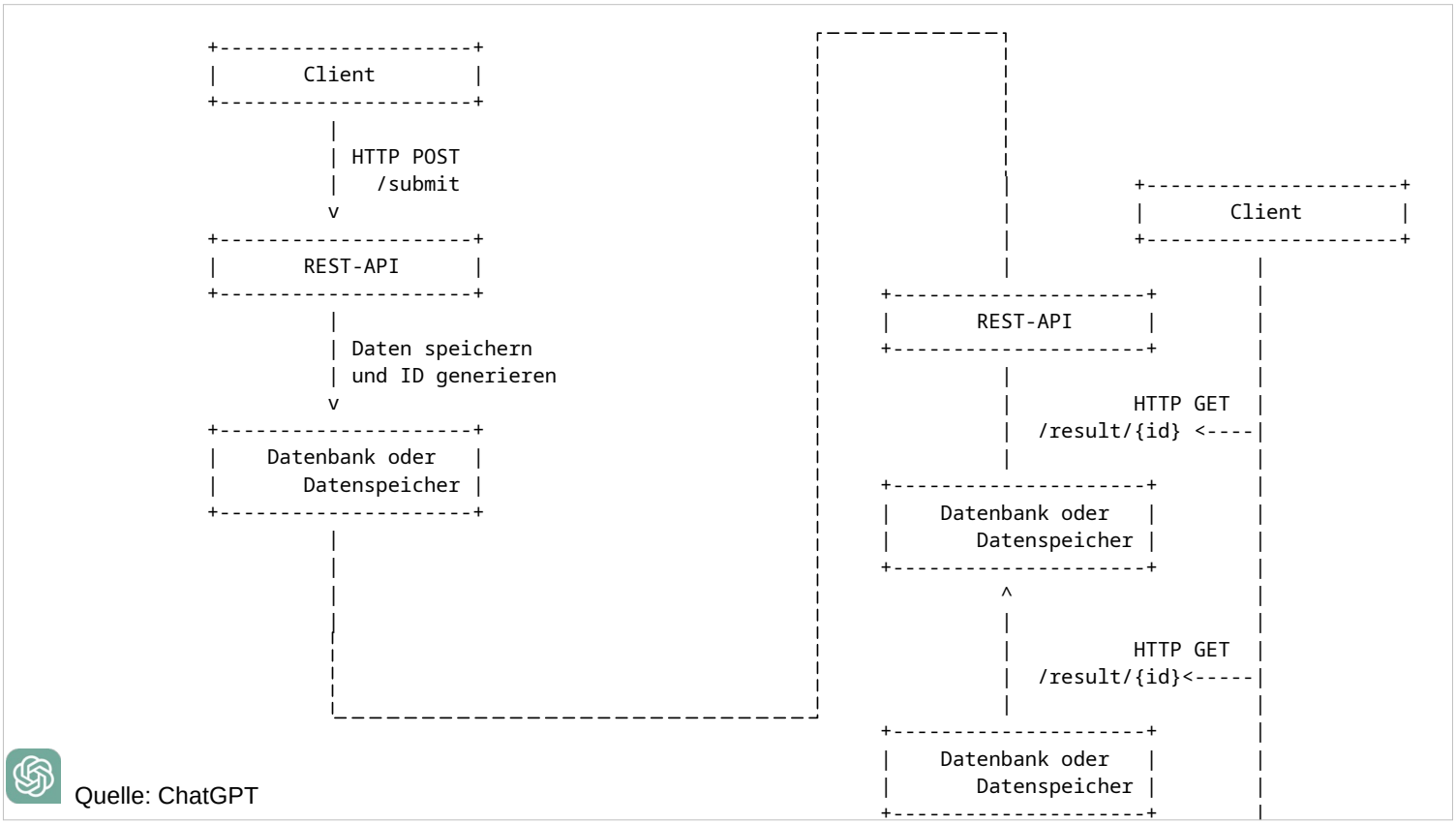
Carsten Rosenberg von heinlein Support hat das Plugin für Rspamd geschrieben DANKE

Um Peekaboo zusammen mit rspamd zu verwenden mussten wir unser Frontend anpassen und auch strukturelle Änderungen vornehmem um die REST API also solche bedienen zu können.

Rest api · Pull Request #194 · scVENUS/PeekabooAV



Mit dem Code aus dem Pull Request # 194 konnten wir hier die Grundlage schaffen um im nächsten Schritt zusammen mit Carsten Rosenberg ein Plugin für rspamd zu entwickeln.

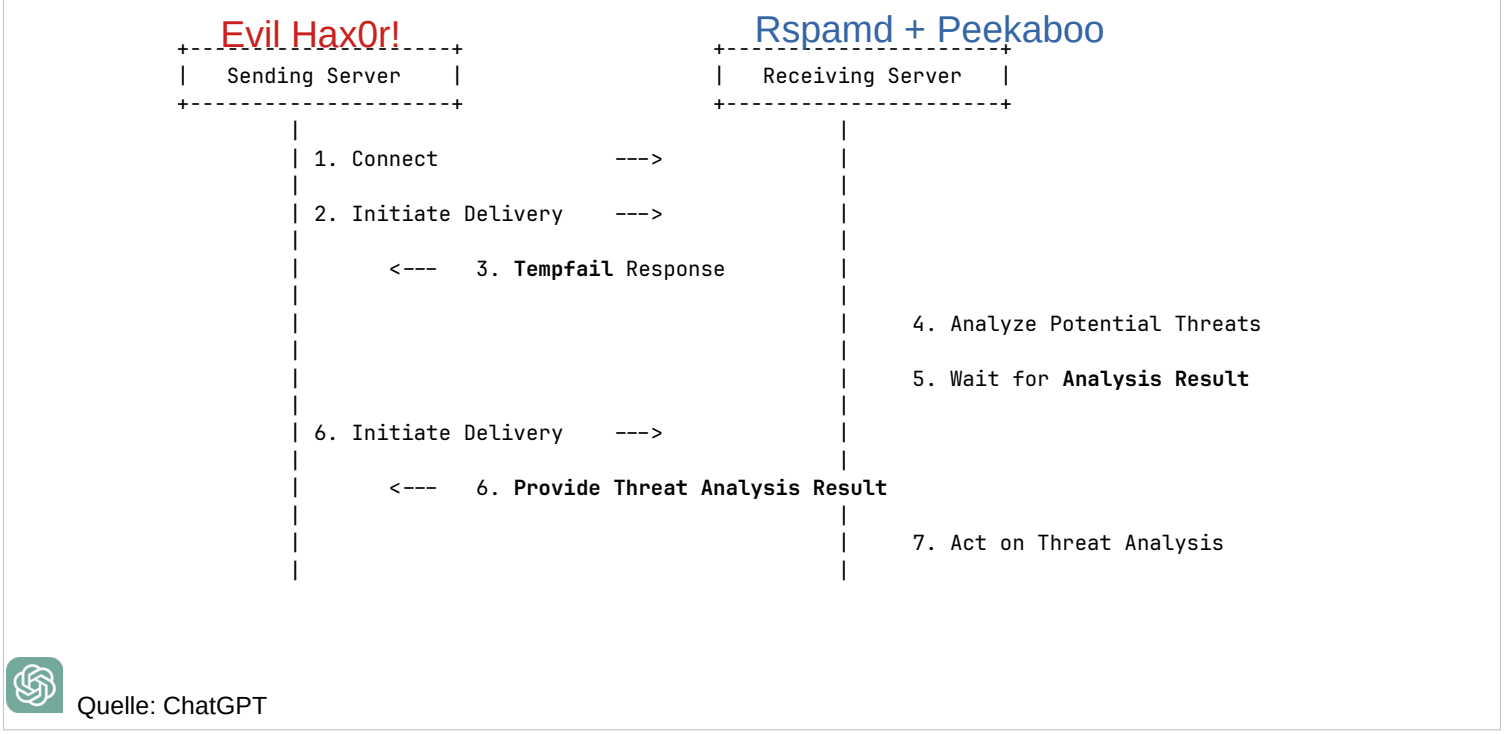


Quelle: ChatGPT

Rspamd hat die ~~Eigen~~Besonderheit, dass es nicht warten kann, dass unsere Analyse fertig ist.

Die Analyse durch Peekaboo kann mehrere Minuten dauern. Das ist nicht mal unbedingt dem Aufwand geschuldet die VM vorzubereiten, aber kommt dadurch zustande, dass das Sample über eine Zeit beobachtet werden soll.

Die REST API hat also zwei Endpunkte: Submit und Report – um die Datei einzugeben und um deren Report abzurufen (polling).



Quelle: ChatGPT

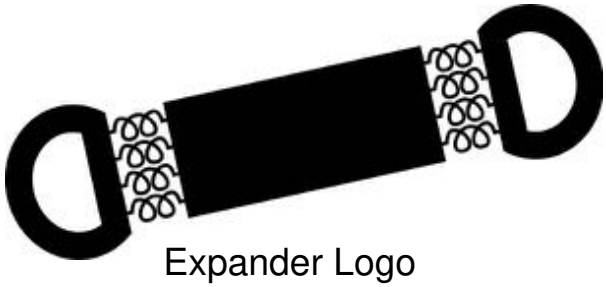
Liefert rspamd die Datei an und es liegt kein Ergebnis im Cache vor (welches sofort zur Verfügung steht) wir rspamd die Mail mit einem Tempfail zurückweisen (Greylisting).

Beim nächsten Versuch der Zustellung liegt dann ein Ergebnis vor und rspamd kann entsprechend reagieren und die Mail gänzlich abweisen oder annehmen.

expander

Im Unterschied zu Amavis packt rspamd Archive nicht aus

https://github.com/science-computing/expander



Expander Logo

Powered by:
CERT-Polska

Karton

„Distributed malware processing framework based on Python, Redis and S3.“

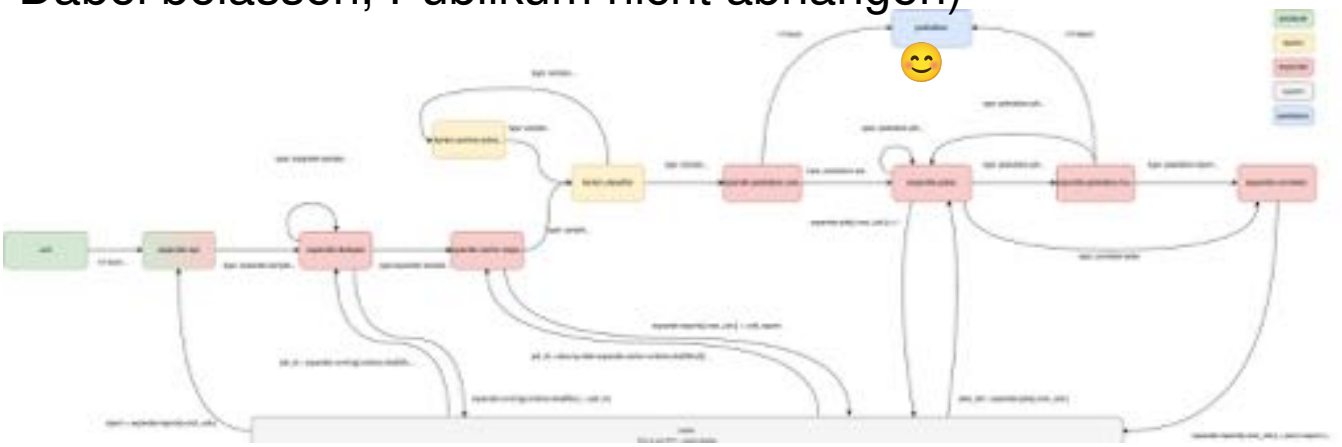
Eine weiterer Unterschied zwischen rspamd und amavis ist, dass Archive von rspamd nicht ausgepackt werden.

Auch das ist auf die hoch optimierte Performance von rspamd zurückzuführen.

Um trotzdem Inhalte von Archiven untersuchen zu können haben wir ein weiteres Open Source Projekt gestartet, den Expander Karton.

Das Karton Framework wir vom polnischen CERT entwickelt und dient ebenfalls der Dateianalyse. Verschiedene Kartons abonnieren dazu einen Messagebus und werden aktiv sobald ein für sie passender Datentyp auftaucht. Das Ergebnis wird dann wieder auf den Bus gegeben.

(**Schaubild vom Expander**, Karton Mechanik **ausgereizt**. Dabei belassen, Publikum nicht abhängen)



Expander Karton Schematische Darstellung

Wie leicht zu erkennen ist werden die Daten nach diesem Schema von links nach rechts verarbeitet, mit Peekaboo als speziellem Karton. Die Anordnung der um Peekaboo befindlichen Kartons sorgt dafür, dass sich das Polling des Ergebnisses realisieren lässt.

https://github.com/CERT-Polska/karton

expression rules

- Flexibilität
- kein Scoring

```
expression.3  : sample.name_declared == 'signature.asc'
                and sample.type_declared in {
                    'application/pgp-signature'
                } -> ignore
expression.4  : sample.file_extension in {
                    'doc', 'docm', 'dotm', 'docx', 'rtf', 'rtx',
                    'ppt', 'pptm', 'pptx', 'potm', 'ppam', 'ppsm',
                    'xls', 'xlsm', 'xlsx' }
                and olereport.has_office_macros == True
                and cuckooreport.score > 4 -> bad
#expression.5  : cortexreport.VirusTotalQueryReport.n_of_all == 0
#                and cortexreport.VirusTotalQueryReport.level == 'safe'
#                -> unknown
```

- aufpassen, den statischen Virenscanner nicht ersetzen
- schöne Möglichkeit verschiedene Reports/Analyzer miteinander zu verknüpfen

Beispiele: - von “alles in die Sandbox” ... keine **verschlüsselten Mails** … viele **ZIP basierte Dateiformate** (Numbers, Pages … Adobe …)

In der Abbildung sehen wir drei Beispielregeln, wie sie im Basisregelsatz enthalten sind.

Die Erste sorgt dafür, dass GPG-Signaturen von Mails nicht analysiert werden. Dies hätte kein Ergebnis und wäre unnötig.

Die Zweite legt fest, dass alle Dateien, deren Dateinamenserweiterungen im Set enthalten ist mit den oletools analysiert werden, sollten Makros enthalten sein wird der Cuckoo report angefordert und der gemessene Score auf größer vier geprüft. Ist dieser Ausdruck wahr wir die Datei als bad klassifiziert.

Im letzten Beispiel werden weitere Analyzer aus Cortex verwendet.

Backend

cortex

- Meta (multi) Analyzer
- Eine Anbindung -> Universum von Analyzern
- -> MISP -> Mitre Datenbanken
- -> Automatisierung -> Cases -> Profit/Win

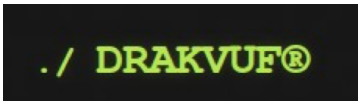
https://github.com/TheHive-Project/Cortex

Cortex stammt aus dem TheHive und MISP-Project Umfeld und dient dem Enrichment der Observables, also den Attributen eines z.B. Incident Response cases.
Cortex kann mit ganz vielen Analyzern verschiedenste Datentypen untersuchen, IP-Adressen, Dateien, Hashes, Domain-Namen …

Cuckoo ist einer dieser Analyzer.
CAPEv2 (ein Fork von Cuckoo) ein weiterer.

Auch Cuckoo3 könnte so angebunden werden.
Jedoch sind alle aus der gleichen Familie und haben ihre Probleme.

https://drakvuf.com (GPL License)
by Tamas K Lengyel



drakvuf

Der Ansatz am **Hypervisor** Daten zu sammeln. Ist technisch sehr interessant. Verwenden auch das Karton Framework. Skalierbar, Flexibel. **Keine Signature Engine**, es wird nur eine Liste von Aktionen protokolliert, aber keine Bewertung! **Idee!** Man könnte die **Cuckoo Signaturen** portieren und als Karton in die Pipeline hängen
-> 5 abstrakte Signaturen anstatt 500 MB json

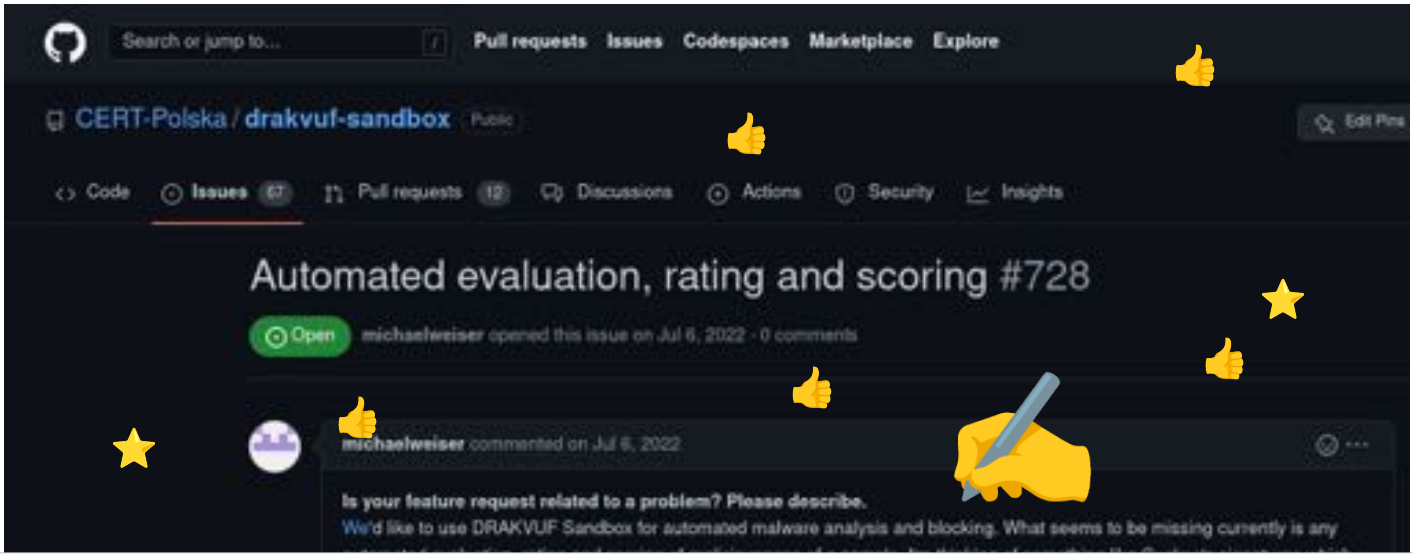
Drakvuf-Sandbox by 

Deswegen sind wir auf der Suche nach einer anderen OpenSource Sandbox.

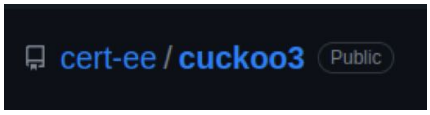
Ein Kandidat ist drakvuf.

Automated evaluation, rating and scoring · Issue #728 · CERT-Polska/drakvuf-sandbox

https://github.com/CERT-Polska/drakvuf-sandbox/issues/728



Wir haben einen Issue im Repository der drakvuf-sandbox geöffnet und gefragt ob ein solches Feature geplant ist bzw. wie es aussehen müsste oder am geschicktesten implementiert werden könnte.



cuckoo 3

- Gemacht für **eine!!1!** Version von Windows 10
- Der Agent wurde zum Treiber

- **Eine** Veröffentlichung
- Hat immer noch nicht richtig abgehoben

Weiterer Kandidat ist Cuckoo 3, entwickelt von der gleichen Firma die ursprünglich Cuckoo 2 erstellt und maintaint hat.

Initial sehr vielversprechend. Dann mussten wir leider feststellen, dass die Entwicklung auch hier stagniert

cape by

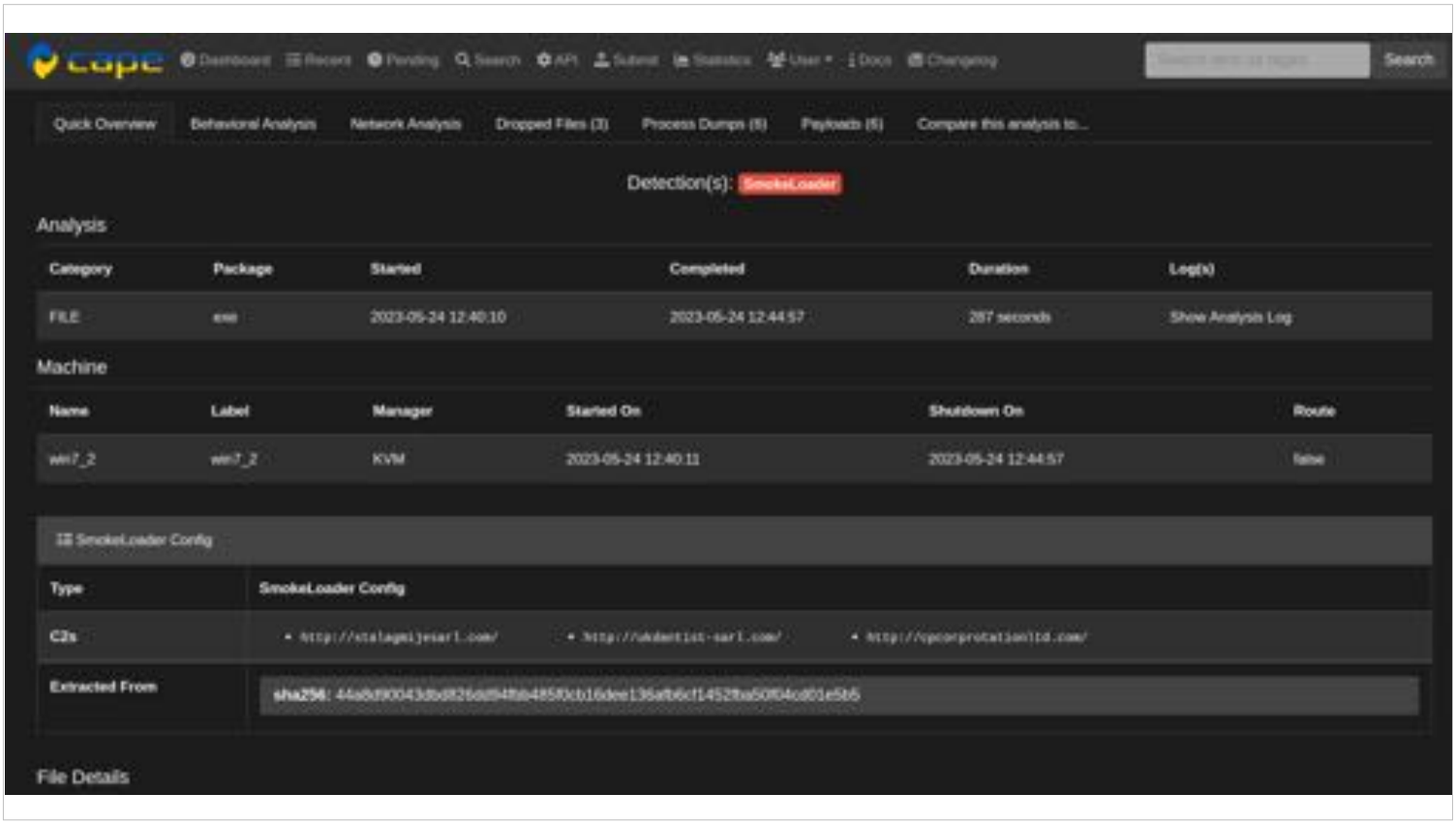
CAPEv2

- vielversprechendste Kandidatin
- Fork von cuckoo 2
- Python3
- für die interaktive Malware-Analyse, Payload Extraction,
- eher unstable

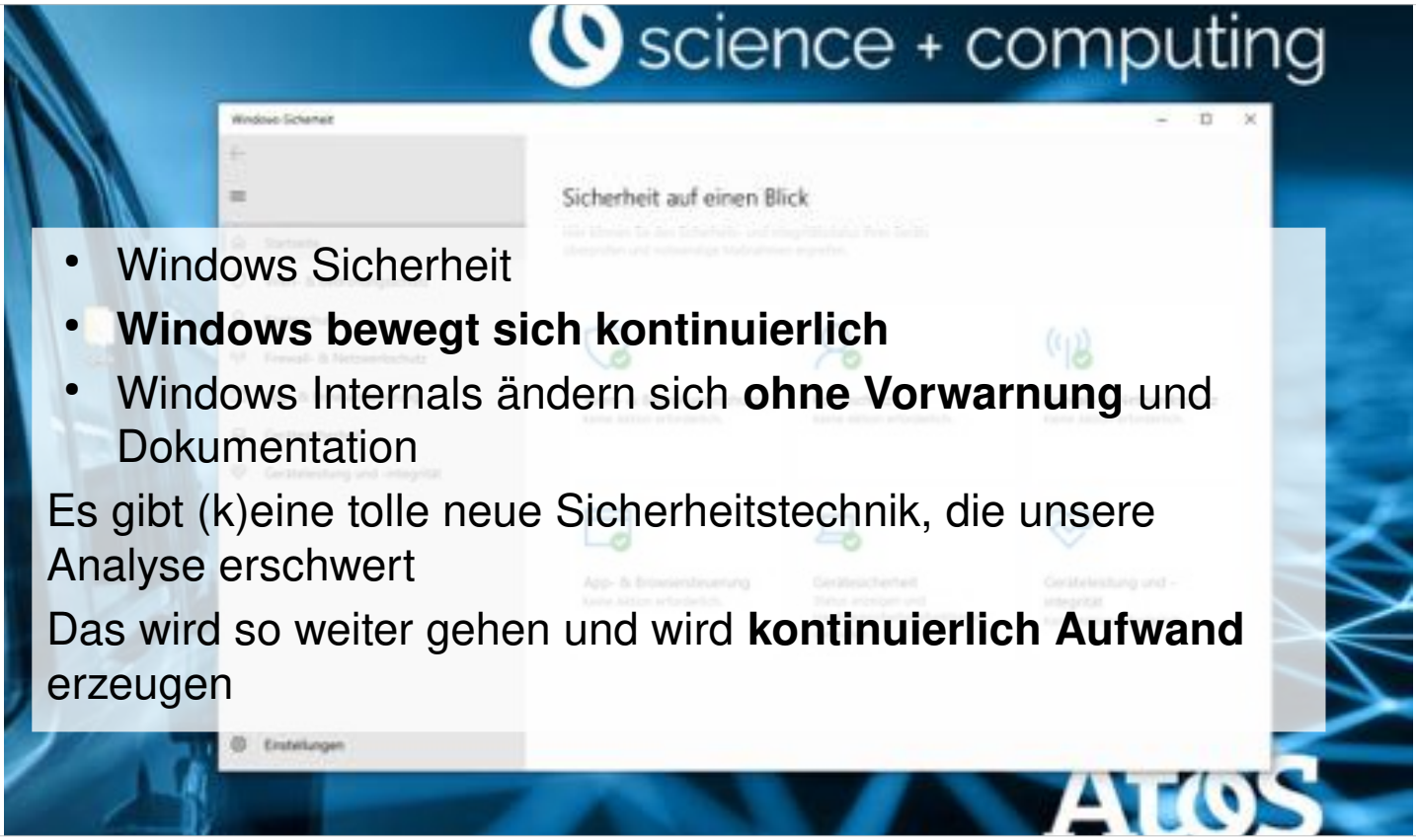
Wir bräuchten für die Automatisierung Stable Releases, das passt aber nicht zu Windows 10.

Die funktionalste der Alternativen.

Leider auch hier Instabilitäten und immer wieder Probleme bei der Analyse. Oft ausgelöst durch Aktualisierungen des Windows Betriebssystems.



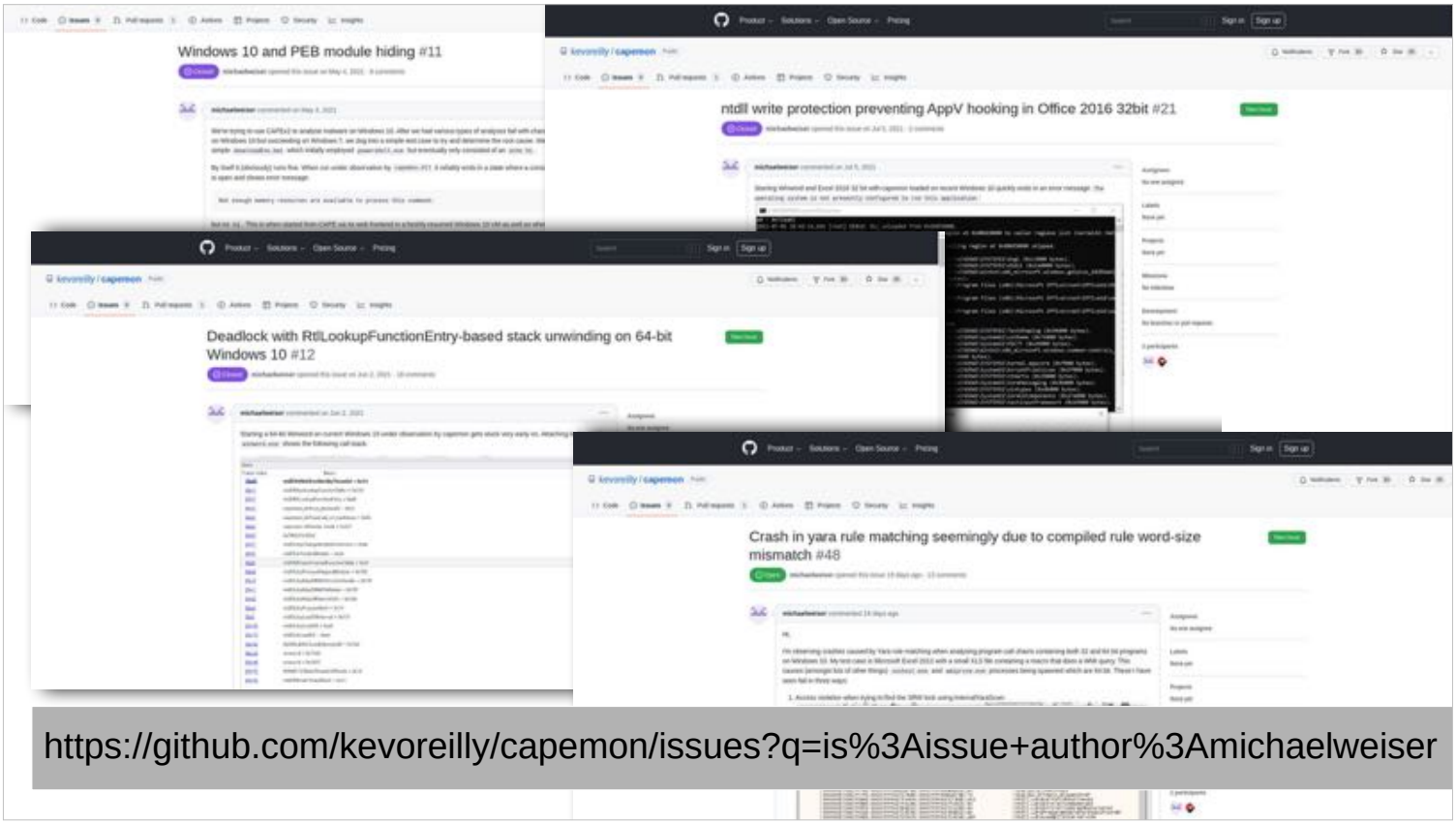
Die Oberfläche ähnelt der von Cuckoo2



- Windows Sicherheit
- **Windows bewegt sich kontinuierlich**
- Windows Internals ändern sich **ohne Vorwarnung** und Dokumentation

Es gibt (k)eine tolle neue Sicherheitstechnik, die unsere Analyse erschwert

Das wird so weiter gehen und wird **kontinuierlich Aufwand** erzeugen



Wir können CAPEv2 ausreichend verstehen und per Debugging in der Windows VM die Ursachen identifizieren und beheben, das ist aber sehr aufwändig.

Wie funktioniert CAPE, Windows, Office? Und **wie lange?**

- Mit **Windows 7** ist die Erkennung auch gut (kauft aber keiner)
- **Windows 10** bewegt sich zu schnell

Das Zusammenspiel von Indirektionen in Microsoft Windows und das Hooking und Debugging des beobachteten Prozesses sorgen für Instabilitäten und zu Problemen, schon bei kleinen Änderungen und Abweichungen (Systemsprache, Patch Level, Office Version)

Sandbox	Windows 10?	Zukunftsfähig ?	Todos
Cuckoo 2	nein	nein	Python3 ...
drakvuf	ja	jein	Signaturen
Cuckoo 3	eine Version	(noch) nicht	Windows
CAPEv2	jein	am ehesten	Windows

Tabelle, Gegenüberstellung der Optionen

Eine Gegenüberstellung der uns bekannten Sandboxen.

Wir warten noch auf die Gewinnerin

Eine OpenSource Sandbox welche verlässlich mit Windows 10 Analyseergebnisse produziert, welche abstrakt genug sind um sinnvoll im Regelwerk gesucht werden zu können.

„Creates executable in filesystem“
„Registers to automatically be started on boot“
„Tries to download executable dll file“

Nach Emotet und Konsorten in neuer Form und Stärke

Wie müssen sich Analyse-Sandboxen weiterentwickeln um mit der Malware mithalten zu können?

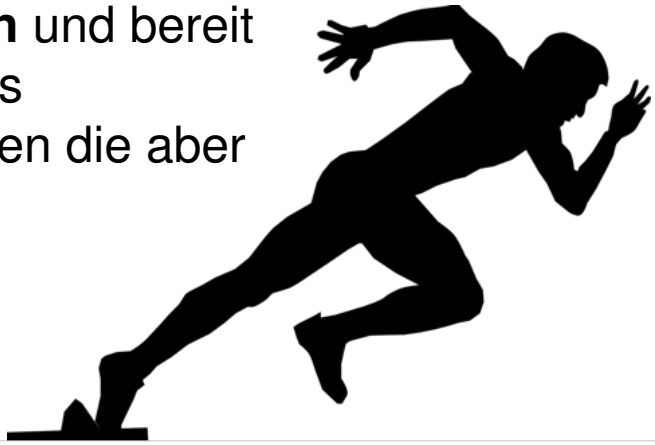
- Die User:inneninteraktion ist zur Erkennung notwendig
- Das Befolgen von in der Mail enthaltenen Instruktionen
- KI Usersimulation

Schon in Cuckoo ist die grundlegende Funktionalität implementiert:
- die Maus wird möglichst zufällig bewegt
- Betriebssystem-Dialoge werden bestätigt

Da bräuchten wir deutlich mehr.

Auf die Plätze! Fertig! ??

- Soweit so **gut**
- Die Hoffnung ist, Ihr konntet was **lernen**
- Gerne **Feedback, Ideen, Anregungen** …
- Wir stehen in den **Startlöchern** und bereit für den Tag, an dem eine neues **Sandboxlösung** kommt (können die aber **alleine nicht** stemmen)



Leider sind die Aufwände immens groß und können von uns nicht einfach so finanziert werden.

Ich hoffe wir konnten vermitteln was wir die letzten Jahre gelernt haben, gerne stehen wir für Fragen auch über den Vortrag hinaus zur Verfügung.

Vielleicht hab ja jemand die zündende Idee, kennt ein Projekt oder hat Kontakte für eine Kooperation um genügend Personenpower aufbringen zu können.

Vielen Dank für Eure Aufmerksamkeit

Kommt gerne auf uns zu:

- Christoph, Felix, Michael
- @peekabooav@fosstodon.org
- ✉️ info@peekabooav.de

SLAC 2023
23.-25. Mai 2023 | Berlin

Wir bleiben dabei: Peekaboo funktioniert!

Es existieren große Pläne um die Erkennung weiter zur verbessern und die vorhandenen Einschränkungen zu lösen.

Interna

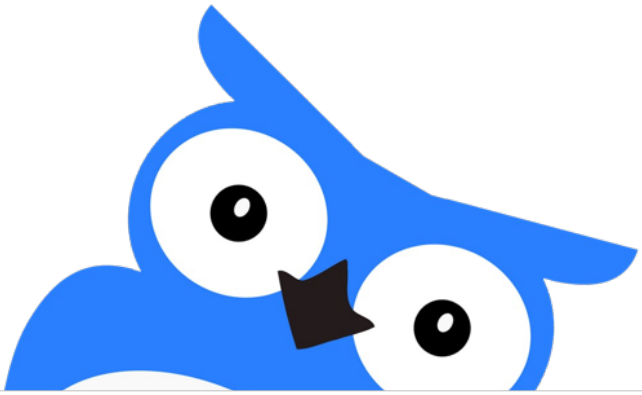
“übrigens”

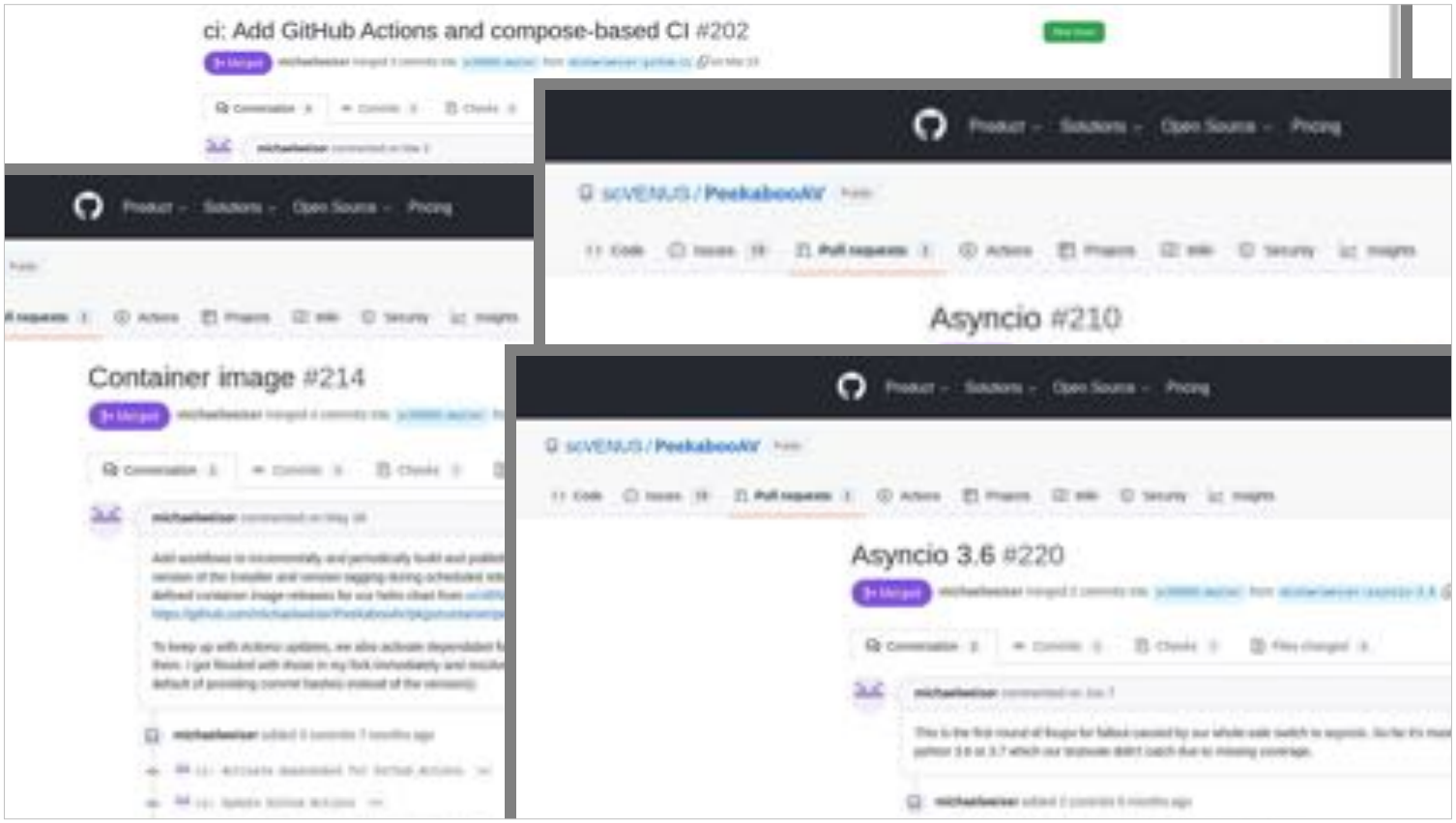
asyncio

container

kubernetes. HELM

ci/cd





Weitere Errungenschaften der letzten Jahre:

- Implementierung in Asyncio
- Containerisierung
- CI/CT