Grundlagen der IT-Sicherheit: Phishing Infrastructure Knowledge

Gina Bartenwerfer, Bernhard Birnbaum, Vincent Donat, Felix Gretschel, Abdalla Khalil, Michelle Kirst

Zusammenfassung—Im Zentrum dieses Projekts steht die Fragestellung, welche Daten im Zusammenhang mit Phishing-Attacken den Dienstanbietern zur Verfügung stehen (könnten), welche zunehmend von Kriminellen ausgenutzt werden. Weiterhin soll daraus geschlossen werden, in welchem Umfang diese Dienstanbieter solche Angriffe erkennen bzw. unterbinden könnten.

Index Terms—Open Source, Privacy, Forensic Design, Phishing

1 Motivation [Gina]

Phishing-Attacken ziehen sich durch das gesamte Netz und dienen Kriminellen als finanzielle Einnahmequelle. Phisher tarnen ihre E-Mails und geben sich beispielsweise als eine Bank aus und leiten Empfänger auf täuschend echte Webseiten weiter, auf denen Daten wie Passwörter eingegeben werden sollen [1].
Wenn ein Phishing-Angriff erfolgreich ist, kann das schwerwiegende Konsequenzen haben. Die Konsequenzen reichen von Gelddiebstahl und Installation von Malware, bis hin zu der Übernahme der Konten und Identitätsdiebstahl [2].
Im Rahmen des Softwareprojekts „Grundlagen der IT-Sicherheit“ beschäftigen wir uns mit der Frage, was Dienstanbieter von Phishing-Attacken wahrnehmen können und was E-Mail-Clients von eingehenden Mails wahrnehmen können.

2 Stand der Technik [Gina]

2.1 Phishing

Laut BSI ist Phishing eine Betrugsmasche die darauf abzielt nach Passwörtern zu „fischen“. Viele Webbrowser und E-Mail-Clients haben bereits Filter und somit eine eingebaute Phishing- und Scam-Erkennung, dennoch existieren weiterhin viele Phishing-Attacken. Google berichtet, dass sie täglich 100 Millionen schädliche E-Mails filterten. Sie seien in der Lage 99,9 Prozent der eingehenden Spam- und Phishing- Mails zu blockieren [3].
Jedoch gibt es trotzdem noch Phishing Attacken. Das Problem ist, dass viele Mitarbeiter von Unternehmen nicht wissen, wann es sich um Phishing handelt. Dadurch werden sie unbewusst zu Mithelfern der Täter [4]. In Proofpoints Report „State of the Phish Report“ wird berichtet, dass 83 Prozent der Befragten 2021 angaben, dass ihr Unternehmen Opfer von erfolgreichen E-Mail-basierten Phishing-Angriff wurden [5].
Doch wie schaffen die Täter das? Sie verwenden verschiedenste Methoden, um ihre Opfer zu „phishen“. Am häufigsten wird das nachahmen von Webseiten zum Beispiel von Banken verwendet. Die Opfer werden dann von der vermeintlichen Bank aufgefordert, die Bankdaten auf einer täuschend echt aussehenden Seite einzugeben. Eine andere Methode ist es HTML-Anhänge an eine E-Mail zu binden, die die Leute ebenfalls bitten, Daten einzugeben. Außerdem können die Phisher mit Hilfe von Skripten dafür sorgen können, das Links eine Originaladresse anzeigen, er jedoch auf eine gefälschte Seite weiterführt. Phisher können den Text in der E-Mail auch als Grafik darstellen und fälschen dafür häufig die E-Mail-Adresse des Absender. Phishingseiten haben häufig gefälschte Namen, die Banken oder Firmen ähneln. Die Seiten sind täuschend ähnlich zum Original und die Phisher nutzen Domainnamen, die vom Original kaum abweichen. Dafür können sie Umlaute verwenden. Zum Beispiel schreiben sie statt „ü“ im URL „ue“. Eine andere Möglichkeit ist kyrillische Buchstaben zu verwenden. Das kyrillische „a“ hat kaum einen Unterschied zum lateinischen „a“, jedoch führt es dann zu einer anderen Webseite [6].

Eingehende Phishing-Mails können u.a. auf Inhalt wie Semantik, Nachricht, sprachliche Mittel und technische Details z.B. Header untersucht werden, wobei letzteres manipuliert werden kann [7].

2.2 Gesetzliche Grundlagen

Bei Phishing liegt nach §269 StGB eine Straftat vor, wenn Daten so verändert werden, dass die Wahrnehmung beim Empfänger entsteht, dass es sich um eine unechte oder gefälschte Urkunde handelt. Darüber hinaus kann §240 StGB greifen, denn sollte der Sender von Phishing E-Mails mit Beispielsweise der Sperre des Kontos drohen, liegt der Tatbestand der Nötigung vor [8].

2.3 Datentypen

2.3.1 Raw Data

Raw Data (dt. Rohdaten) sind vorerst nicht klassifizierte Bitfolgen bzw. Datenströme von Teilen des Systems, die Daten der anderen Datenarten enthalten können. Rohdaten können beispielsweise Netzwerkpakete, sowie u.a. primär Speicher- und Datenträgerbilder, wie Bilder sein [9].

2.3.2 Details about Data (DT3)

Details über Daten sind sogenannte Metadaten, die Auskunft über die Nutzdaten geben, welche innerhalb oder auch außerhalb der Nutzdaten abgespeichert sein können. Hierzu gehören flüchtige, sowie persistente Metadaten wie zum Beispiel Sequenznummern von Netzwerkpaketen [9].

2.3.3 Communication Protocol Data (DT5)

Um das Kommunikationsverhalten von Systemen zu kontrollieren werden Communication Protocol Data (dt. Kommunikationsprotokolldaten) verwendet. Dazu gehören Netzwerkkonfigurationsdaten und die Inter-Prozess Kommunikation bei IT-Anwendungen [9].

2.3.4 Process Data (DT6)

Alle Daten über einen laufenden Prozess werden als Prozessdaten bezeichnet, wie zum Beispiel der Prozessstatus, Prozesseigentümer, Priorität, Speichernutzung und die dazugehörige IT-Anwendung(hier einzelne Threads bzw. Daten darüber) [9].

3 Konzept und Versuchsaufbau [Gina]

Der erste Schritt war es, uns theoretisches Wissen anzueignen, um eine Grundlage für das Projekt zu schaffen und um besser zu verstehen, woran und womit wir arbeiten.
Als nächstes haben wir die benötigten Werkzeuge installiert und eingerichtet. Dazu gehört u.a. der Testerstick und Werkzeuge wie Wireshark und Thunderbird.
Danach mussten wir Opfer von Phishing E-Mails werden, damit wir sie anschließend analysieren können. Dafür haben wir neue E-Mail Adressen auf verschiedensten Mailprovidern angelegt und uns mit dieser auf verschiedensten Webseiten bzw. Foren angemeldet.
Darauf hin haben wir die Phishing-Mails mit Wireshark und mitmproxy analysiert.
Die Analyseergebnisse wurden evaluiert, um die Frage was große Dienstanbieter von Phishing-Angriffen wahrnehmen können zu beantworten.
Anschließend fassen wir unsere Ergebnisse zusammen und bieten einen Ausblick, was in Zukunft getan werden könnte.

3.1 Grundlegende Vorbereitungen

3.1.1 Wie man Ziel von Phishing-Angriffen wird [Abdalla]

Betrüger greifen nicht jeden, der sich auf verschiedenen Webseiten oder Newslettern angemeldet an. Sie suchen sich jemanden, der aktiv auf den Angriff eingeht oder wo es finanziell lohnenswert ist.
Im Folgenden sind die von uns genutzten Möglichkeiten, um Ziel von Phishing-Angriffen zu werden [10]:

- Auf dubiosen Webseiten anmelden
- Veröffentlichung von E-Mail bzw. persönlicher Daten in sozialen Netzwerken
- Bösartige E-Mails wie z.B. Spam/Scam öffnen oder beantworten
- Links oder Anhänge von Spam-Mails anklicken bzw. herunterladen
- Abonnieren vieler Newsletter

Mit anderen Tools/Webseiten wie z.B. TemprMail (Dienst für Wegwerf-E-Mail-Adressen) kann man viele Spam- oder Phishing-Mails bekommen [11].

3.1.2 Einrichten eines E-Mail-Systems mit synthetischen Daten [Felix]

Um Phishing-Mails untersuchen zu können, muss man eben diese erst einmal bekommen. Für diesen Zweck haben wir E-Mail Accounts mit synthetischen Daten erstellt. Das die Accounts nicht mit realen Daten ausgestattet wurde dient dem Eigenschutz. Zusätzlich erlaubt das Anlegen der E-Mailadressen theoretisch eine einfachere Verfolgung der Quellen von Phishing-Mails. D.h. durch welche Website/ Aktion ist man auf der Mailingliste Krimineller gelandet.
Allerdings ist man bei der Erstellung solcher E-Mail auf bestimmte Anbieter beschränkt. Denn viele setzen eine Telefonnummer für die Anmeldung voraus. Die persönliche Nummer zu verwenden wäre hierfür nicht sinnvoll, da sie direkt gegen das eigentliche Ziel synthetischer Daten geht. Und auch sogenannte throw-away Nummern die Online angeboten werden funktionieren weitgehend nicht.

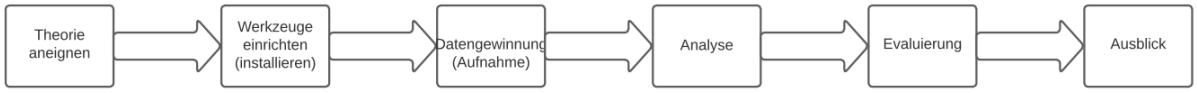


Abbildung 1: Methodik

Um E-Mails untersuchen zu können, haben wir auf unseren Virtual Machine's Thunderbird installiert und mit unseren synthetischen Daten ausgestattet. Demselben Prinzip nach wurde mit Evolution umgegangen.

3.2 Operative Vorbereitung

3.2.1 Aufbau einer rauscharmen Umgebung [Felix]

Eine rauscharme Umgebung in Bezug auf ein Computersystem bezieht sich darauf, das es wenige Verbindungen zu Webservices außerhalb der direkt Aufgerufenen gibt. Linux ist ein Beispiel für ein Betriebssystem was recht datensparsam und rauscharm arbeitet. Die Unterbindung unwichtiger Verbindungen ist für unsere Untersuchungen von äußerster Wichtigkeit, da es die Analyse der Datenaustausche wesentlich vereinfacht bzw. erst ermöglicht. Zusätzlich ist es von Vorteil das nicht zu viele Programme im Hintergrund laufen um die Analyse des Arbeitsspeichers (RAM) zu erleichtern. Ein weiterer, für uns sehr wichtiger Aspekt, ist der Selbstschutz, wodurch sich die Verwendung einer VM zusätzlich anbietet. In unseren Untersuchungen arbeiten wir viel mit Phishing Mails und deren potenziell schädlichem Anhang (z.B.: Malware). Durch die Verwendung einer VM können wir das persönliche Risiko für uns bzw unsere Daten und Systeme etwas einschränken. Dies setzt jedoch voraus das die VM regelmäßig zurückgesetzt wird.
Um die Reproduzierbarkeit und Erweiterbarkeit unserer Arbeit zu gewährleisten haben wir uns dazu entschlossen ausschließlich Open-Source-Software zu verwenden.

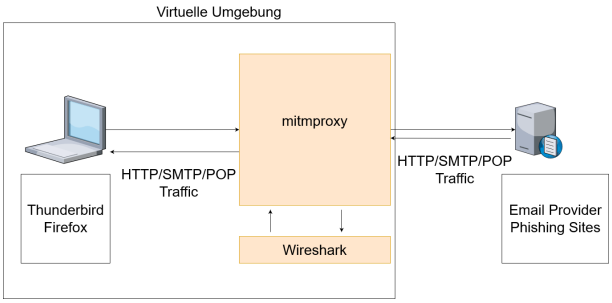


Abbildung 2: Virtualisierte IT-Infrastruktur als IT-Systemlandschaft.

3.2.1.1 Virtuelle Maschine

Als Grundlage für die Rauscharme Umgebung verwenden wir die Testerstick VM. Diese Verwendet als Betriebssystem Debian Mate und zählt somit zu den rauscharmen Umgebungen.

Für die Untersuchung laufen auf dieser Thunderbird und Evolution um Emails zu öffnen und eine Analyse zu ermöglichen. Die Verwendung der E-Mail Clients hat für unsere Arbeit entscheidende Vorteile. So lassen sich zum Beispiel Cookies klar trennen in deren Herkunft. Also ob sie von der Mail kommen und sich somit die Cookies in der Datenbank der Clients wiederfinden oder ob sie von der Phishing-Website ihren Weg auf das System geschafft haben. Als Browser, um die Phishing Links verfolgen zu können, befindet sich eine leicht abgewandelte Version von Firefox auf dem System. Diese Modifikationen dienen aber nur der Vereinfachung der Analyse und verändern nicht aktiv den Datenstrom. Außerdem ist mitmproxy und Wireshark installiert um die Netzwerkverbindungen sichtbar zu machen.

3.2.2 Werkzeuge [Gina]

Um die Phishing E-Mails IT-Forensisch zu untersuchen, haben wir verschiedene Werkzeuge verwendet, die im folgenden vorgestellt werden.

3.2.2.1 Wireshark

Wireshark [12] ist eine 1998 erschienene Open-Source-Software die Nutzern dabei hilft, Netzwerkprotokolle zu analysieren, in dem es Netzwerkverkehr filtert, erfasst und speichert. Die Pakete die abgefangen werden, können dabei von verschiedensten Netzwerkprotokollen sein, z.B. TCP, DNS, HTML [13].

3.2.2.2 Mitmproxy

MITM bedeutet Man in the middle. mitmproxy [14] wird über die Kommandozeile aufgerufen und speichert HTTP und HTTPS Verkehr. Es protokolliert selbst verschlüsselten Verkehr zwischen Browser und Server. Entwickler benutzen es gern, zum Debuggen und schauen sich dafür die aufgezeichneten Protokolle an. Darüber hinaus wird es verwendet um MITM Attacken zu verhindern. Jedoch greift mitmproxy beim aufzeichnen der Daten selbst in den Datenverkehr ein, was für einige Projekte zum Nachteil werden könnte [15].

3.2.2.3 Thunderbird

Thunderbird [16] ist ein Open Source E-Mail-Client, der lokal auf dem Computer arbeitet. Es stellt keine eigenen E-Mail Adressen bereit, jedoch ist es möglich seine E-Mail mit Thunderbird zu verbinden und dann werden Nachrichten im Programm empfangen, gesendet und verwaltet [17].

3.2.2.4 Evolution

Evolution [18] ist ein Open Source E-Mail-Client für Linux. Neben dem verwalten von E-Mails verfügt es auch über Kalender und Adressbücher. Durch Verschlüsselung ist es möglich, die Sicherheit von Daten zu erhöhen [19].

3.3 Methodik der Datenakquise [Bernhard]

In Abbildung A.1 ist unsere entwickelte Methodik zur Datenakquise, d.h. zur Aufzeichnung der Datenströme (Netzwerkdatenstrom, Arbeitsspeicher, Massenspeicher) abgebildet. Die Umsetzung der Methodik wird genauer in Punkt 4.5 erläutert.

1) **Versuchsaufbau**: Da die gesamte Untersuchung in einer virtualisierten Systemlandschaft durchgeführt werden soll, benötigen wir eine virtuelle Maschine zum Aufsetzen der Infrastruktur. In unserem Fall haben wir uns für Oracle VirtualBox [20] entschieden, da diese Software bereits auf dem von uns verwendeten Testerstick der Universität vorhanden war. Als Webbrowser benutzen wir Firefox ESR (Extended Support Release) [21].
   Unsere erste Aufgabenstellung erfordert das Analysieren von Daten innerhalb des Netzwerkdatenstroms. Dafür wurden uns zwei verschiedene Ansätze vorgegeben, die mit unterschiedlichen Tools zu realisieren sind. Zum einen benötigen wir Wireshark, um den Netzwerkdatenstrom (TLS-Traffic) aufzuzeichnen und ihn anschließend mit dem bei der Untersuchung generierten SSLKeyLogFile [22] zu entschlüsseln. Das zweite Werkzeug ist mitmproxy, welches ausschließlich HTTP-Traffic mitschreibt. Aufgrund der speziellen Funktionsweise dieses Proxys [23], liegt der Datenverkehr bereits entschlüsselt vor.
   Die zweite Teilaufgabe, welche das Aufsetzen eines Open-Source-E-Mail-Systems fordert, gibt uns als zu verwendende Mail-Clients Thunderbird und Evolution vor.
   Zum späteren Auswerten der gesammelten Daten benötigen wir außerdem noch sqlite3 [24], um die von Firefox und Thunderbird erstellten Datenbanken auslesen zu können.
2) **Vorbereitungen für die Aufzeichnung**: Im ersten Schritt, noch bevor irgendwelche Daten aufgezeichnet werden, müssen alle benötigten Tools zur Datenakquise gestartet werden. Dabei ist zu beachten, dass eine saubere Arbeitsumgebung gewährleistet werden muss, um fehlerhafte Ergebnisse in der Analyse zu vermeiden. Dazu muss die Browser-History geleert werden sowie die Cache-Speicher aller verwendeten Programme gelöscht werden, um Daten von vorangegangenen Untersuchungen zu verwerfen. Das Starten der für die Untersuchung benötigten Tools umfasst zunächst die Mail-Clients (Thunderbird und Evolution) und anschließend, nachdem das E-Mail-Postfach synchronisiert wurde, die Aufzeichnungstools Wireshark und mitmproxy für den Netzwerkdatenstrom. Die Reihenfolge ist insofern wichtig, da wir nicht den von den Mail-Clients generierten Traffic aufzeichnen wollen. Desweiteren ist zu beachten, dass vorher auch die Umgebung korrekt aufgesetzt wurde: Wireshark benötigt später für die Analyse ein „SSLKeyLogFile“, um den mitgeschriebenen TLS-Traffic zu entschlüsseln [22]. Für mitmproxy ist wichtig, dass der Browser-Traffic auch über den Proxy auf „localhost“ bzw. „127.0.0.1“ läuft, damit damit der Datenstrom aufgezeichnet werden kann. Außerdem müssen die von mitmproxy generierten Zertifikate in Firefox als „trusted“ markiert bzw. eingebunden werden [25].
3) **Start der Aufzeichnung**: Anschließend kann mit dem Start eines Timers die Aufzeichnung begonnen werden. Der Timer ist notwendig, damit später eindeutig nachvollzogen werden kann, wann welche Aktion ausgeführt wurde.
4) **Laden der E-Mail**: Sobald alle Tools einsatzbereit sind kann im Mail-Client die zu Untersuchende E-Mail abgerufen werden. Dadurch wird der Content der Mail sowie mögliche Anhänge geladen. Nach dem vollständigen Laden sollte der Zeitpunkt in Wireshark vermerkt werden.
   Diese Vorgehensweise erzeugt einerseits einen Netzwerkdatenstrom durch das Abrufen der E-Mail vom Server, andererseits werden von den Mail-Clients Daten lokal gespeichert (Massen- und Arbeitsspeicher) und können später ausgelesen werden [26].
5) **Durchgehen der Phishing-Mechaniken**: Im nächsten Schritt sollten die eventuell enthaltenen Phishing-Links oder Anhänge im Webbrowser (laufende Instanz des Firefox ESR) geöffnet werden. Insofern die verlinkte Webseite (noch) zur Verfügung steht (könnte bereits offline sein, einen 404-Fehler anzeigen, oder über den Proxy nicht aufrufbar sein (502, Bad Gateway)) [27], können nach dem vollständigen Laden noch synthetische Daten in eventuell vorhandene Formulare oder Textfelder eingegeben werden.
   Dadurch werden wiederum Pakete im Netzwerkdatenstrom erzeugt (z.B. beim Aufrufen der Phishing-Seite oder beim Eingeben synthetischer Daten), sowie Daten im Massenspeicher und Arbeitsspeicher abgelegt.
6) **Beenden der Aufzeichnung**: Wenn alle verfügbaren Inhalte der E-Mail geöffnet bzw. geladen wurden, kann die Untersuchung beendet werden. Bevor irgendein Programm geschlossen werden kann, muss der Arbeitsspeicher der virtuellen Maschine gesichert werden. Erst danach können alle laufenden Tools beendet werden, wobei zu beachten ist, dass keine Daten ungespeichert verworfen werden dürfen.

3.4 Analyse

3.4.1 Untersuchung der E-Mail Header [Felix]

Header einer E-Mail enthält neben Informationen über den Empfänger und Betreff auch Informationen über den Sender und den Weg den die Mail genommen hat (welche Server waren bei der Sendung beteiligt) [28]. Dies ist für unsere Untersuchung sehr interessant. Denn hier kann man recht schnell herausfinden ob die angegeben Daten des Versenders zutreffen und somit ob es sich um eine Phishing/ Spam Mail handelt. Daher handelt es sich bei der Kontrolle des E-Mail Headers auch ein Kriterium welches von großen Dienstanbietern verwendet wird um gefährliche Mail herauszufiltern bzw. potentiell schädliches zu markieren.

Jedoch sind sich dessen auch die Kriminellen bewusst und fälschen die Header um den Filtern zu entgehen. Daher ist die Korrektheit dieser Header immer etwas, was vor der Analyse geklärt werden sollte.
E-Mail Header lassen sich in so ziemlich allen Programmen (so auch Thunderbird/Evolution) und Online E-Mail Portalen einsehen. Die Analyse dieser stellt auch keine Schwierigkeit dar. Wenn man diese nicht per Hand durchschauen will, kann man diese auch in Webseiten (z.B. Message Header Analyzer [29]) einfügen die die Informationen klarer widerspiegelt. Dort kann man dann ablesen ob die E-Mail des Senders dem gleicht was im Header sichtbar ist.
Obwohl dies alles nach einer einfachen Lösung unseres Problems klingt, ist es keine. Denn wie schon angemerkt laufen bereits sehr genaue Filter großer Mailservice-Anbieter über die Header und sortieren gefährliches und manipuliertes aus. Das heißt das was man tatsächlich in das E-Mail Postfach bekommt ist schon durch diese Filter gekommen. Was das für unsere Untersuchungen bedeutet ist, das diese sich kaum von Headern echter E-Mails unterscheiden.

3.4.2 Untersuchung des Netzwerkdatenstroms mit Wireshark [Michelle]

Im Folgenden wird der in Diagramm A.2 gezeigte Leitfaden für die Analyse der Wireshark Aufnahmen beschrieben. Dabei ist zu beachten, dass die Schritte 1-4 für ein einzelnes Datenpaket durchgeführt werden und danach für jedes zu untersuchende Datenpaket wiederholt werden müssen.

3.4.2.1 Vorbereitung der Analyse

Bevor mit der eigentlichen Analyse der Wireshark Aufnahme begonnen wird, nehmen wir noch eine Einstellung in Wireshark vor: die Namensauflösung wird aktiviert (Statistik → Endpunkte → Namensauflösung). Dadurch werden die IP-Adressen in IPv4 als Hostnamen dargestellt. Die Hostnamen, die dadurch sichtbar werden, sind später leichter wiederzuerkennen, sollten sie in den zu untersuchenden Datenströmen (z.B. bei einem HTTP Netzwerkdatenstrom als Tracker) nochmal auftauchen, da diese einen höheren Wiedererkennungswert haben, als die reinen IP-Adressen. Der nächste Schritt ist die Überprüfung der sichtbar gewordenen Hostnamen auf ihre Echtheit mit der Webseite IPinfo [30]. Dabei ist es egal, ob man dort die Hostnamen oder IP-Adressen angibt. Eine besonders hilfreiche Information, die man mit dieser Webseite bekommen kann, und die viel Aufschluss über die Vertrauenswürdigkeit der Domain gibt, ist, ob der Standort der IP-Adresse mit der Endung der Domain übereinstimmt. Endet der Hostname z.B. auf .de, aber es wird angezeigt, dass die IP-Adresse in einem anderen Land lokalisiert ist, ist das ein guter Hinweis darauf, dass es sich hierbei um eine Phishing-Webseite handelt.

3.4.2.2 Die Analyse

1) **Der I/O-Graph**: Es ist wichtig die Analyse in Phasen aufzuteilen. Das Ziel dabei ist zu erkennen, welche Domains in den verschiedenen Phasen der Interaktion mit der Webseite erscheinen. Um nicht wahllos durch die Datenpakete zu klicken, benutzt man den I/O Graph (Statistiken → I/O Graph) als Orientierungshilfe. Dort wird dargestellt, wie viele Pakete pro Sekunde ein- und ausgehen. Die Anstiege im Graphen zeigen die Zeitpunkte zu denen besonders viele Pakete übertragen wurden. Dort kann man erkennen, welche Interaktionen mit der E-Mail zu welchen Netzwerkverbindungen geführt haben. Durch einen Klick auf den Hochpunkt eines Anstiegs, wird das Paket angezeigt, das bei diesem Hochpunkt erschienen ist. Es lohnt sich in jedem Fall auch Datenpakete, die direkt davor oder danach erschienen sind, zu untersuchen.
   Bei der Erstellung der Wireshark Aufnahme war es wichtig einen Timer zu starten, damit man die Zeiten seiner Interaktionen mit der Website genau datieren kann. Denn nun können die Aufzeichnungen zu den Zeiten der Interaktionen mit der Webseite (z.B. wann die Eingabe eigener Daten erfolgte) mit den angegebenen Zeiten in dem Paket verglichen werden, um zu überprüfen, ob ein Anstieg im I/O Graph mit der Eingabe auf der Webseite übereinstimmt. So lässt sich herausfinden, ob bestimmte Interaktionen die Anstiege verursacht haben. Falls diese Anstiege allerdings nicht direkt zugeordnet werden können, sollten sie genauer untersucht werden.
2) **Schlüsselwörter**: Es sollte außerdem bei den Paketen auf enthaltene Protokolle, Methoden und Signale geachtet werden. Dadurch erhält man mehr Informationen, welche Art von Datenaustausch wann stattfand. Eine gute Orientierungsmöglichkeit sind dabei diese Schlüsselwörter:
   - „POST“ [31] bedeutet, dass eigene Informationen eingegeben wurden, und in einem Paket mit diesem Schlüsselwort wiederzufinden sein sollten.
   - „GET“ [31] bedeutet, dass dort Informationen angefordert wurden. Diese Methode ist weniger sicher, da hier eingegebene Daten in der URL lesbar sind. Daten wie z.B. Passwörter dürften in Datenpaketen mit diesem Schlüsselwort nicht zu finden sein.
   - „Handshake“ [32] signalisiert die gegenseitige Bestätigung der Teilnehmer über den Erhalt der Nachrichten.
3) **Streams**: Innerhalb eines Datenpakets können sich Streams der unterschiedlichen Protokolle angeschaut werden [33]. (Rechtsklick auf das Paket → Folgen → xx Stream). Dabei wird ein Filter aktiviert, der alle Datenpakete des gleichen Protokolls anzeigt. Das sich öffnende Fenster enthält die Daten des Streams. In diesen kann man unter anderem nach den Hostnamen der IP-Adressen Ausschau halten. Sollten URLs lesbar sein, lohnt es sich meist diese durch eine einfache Google Suche zu überprüfen. So können z.B. Tracker ausfindig gemacht werden. Außerdem können in den Streams auch eingegebene Passwörter oder E-Mail Adressen erscheinen.
   Es lohnt sich, sich durch alle verfügbaren Streams der Protokolle zu klicken, da die Informationen mitunter anders sind oder anders dargestellt werden (z.B. weniger oder mehr verschlüsselt). Wenn man damit fertig ist, sollte darauf geachtet werden, den Filter, der am Anfang aktiviert wurde, wieder zu entfernen. Dadurch werden wieder alle Datenpakete angezeigt.

4) **Inspektion des Payloads**: Der letzte Schritt ist die Überprüfung des Payloads der Datenpakete. Dafür werden die Header der Datenpakete überprüft. Hier lässt sich herauslesen, welche Cookies übermittelt werden und wer der Referrer für diese ist. Es ist auch zu sehen, welche Scripte und Dateien (CSS- bzw. JavaScript Dateien) übermittelt werden.

3.4.2.3 Eine alternative Herangehensweise

Sollten sich bei dem systematischen Durchgehen durch die Datenpakete keine aufschlussreichen Informationen ergeben haben, besteht noch die Möglichkeit, manuell nach den Eingaben, die man während der Wireshark-Aufzeichnung auf der Phishing Webseite getätigt hat, zu suchen. Zuerst benutzt man das Kommandozeilentool „Strings“. Damit werden alle Zeichenketten aus der Wireshark Aufnahme extrahiert. Die Eingabe im Terminal dafür lautet: „`strings -a nameoffile.pcapng`“. Das „`-a`“ stellt dabei sicher, dass die komplette Eingabedatei gescannt wird [34].
Um die Ausgabe etwas einzuschränken, besteht die Möglichkeit zu begrenzen, welche Länge die ausgegebenen Strings mindestens haben sollen. Der Befehl „`strings -n 8 nameoffile.pcapng`“ würde dabei nur Zeichenketten der Länge acht oder größer ausgeben [34].
Bisher wurden die Strings nur auf dem Terminal ausgegeben. Zur weiteren Untersuchung werden diese in eine Textdatei umgewandelt. Es genügt „`> name.txt`“ an die bisherigen Befehle anzuhängen. Die gespeicherten Zeichenketten können nun mit „grep“ untersucht werden. Grep ist ein weiteres Kommandozeilentool, welches dem Benutzer ermöglicht, bestimmte Buchstabenkombinationen aus einer Textdatei auszulesen [35]. Mit dem Befehl „`cat name.txt | grep testwort >> nametestwort.txt`“ werden alle Strings aus der Datei „name.txt“, die „testwort“ enthalten, in eine neue Datei „nametestwort.txt“ ausgelagert. Dies kann beliebig oft mit unterschiedlichen Zeichenfolgen wiederholt werden. Allerdings sollte dabei darauf geachtet werden, dass „>>“ statt „>“ benutzt wird, damit die neuen Ergebnisse nicht die Vorherigen überschreiben, sondern nur am Ende der Datei hinzugefügt werden. Je nachdem, welche Daten auf der Phishing-Webseite eingegeben wurden, kann so spezifisch nach dem Vornamen, Nachnamen, einer E-Mail Adresse oder einer Telefonnummer gesucht werden.

3.4.3 Untersuchung des Netzwerkdatenstroms mit mitmproxy [Michelle]

Im Folgenden sind ein paar kurze Eckpunkte für die Analyse mit mitmproxy beschrieben. Dabei ist anzumerken, dass unsere Analysen vorrangig mit Wireshark stattfanden, und mitmproxy eher als Ergänzung genutzt wurde. Da mitmproxy explizit nur HTTP und HTTPS Netzwerkverkehr aufzeichnet, ist es nicht nötig Filter für Netzwerkprotokolle zu aktivieren. Dies ist unter anderem einer der Gründe, warum es uns hauptsächlich als Ergänzung zu Wireshark diente. Da auch einige andere Netzwerkprotokolle wichtig für die Untersuchung sind, die bei mitmproxy nicht zur Untersuchung zur Verfügung stehen, kann es passieren, dass wichtige Informationen nicht angezeigt werden, die Hinweise auf einen Phishing-Angriff geben könnten.
Wie in Abschnitt 3.4.2.2 in Schritt 2 beschrieben, gibt es Schlüsselwörter, die einem Hinweise darauf geben, welche Art von Datenaustausch in dem jeweiligen Netzwerkpaket stattgefunden hat. Auf die gleichen Schlüsselwörter kann auch in der Analyse mit mitmproxy geachtet werden.
Die Hostnamen der Verbindungen müssen nicht, wie in Wireshark, erst manuell aktiviert werden, sondern werden gleich angezeigt, wenn eine Verbindung dazu aufgebaut wird. Auch hier ist es von Vorteil, die URLs, die bei einer Verbindung mit einer Webseite aufgetaucht sind, genauer zu beobachten und ggf. mit IPinfo zu überprüfen.

3.4.4 Vergleich der Vor- und Nachteile von Wireshark (SSLKeyLogFile) und mitmproxy [Abdalla]

3.4.4.1 Wireshark

Man benutzt Wireshark, um die Struktur des Datenverkehrs zu sehen bzw. zu analysieren. Auch wenn man Probleme im Netzwerk suchen und beheben möchte, kann man das alles mit Wireshark durchführen.
Während der Analyse sind uns folgende Eigenschaften aufgefallen:

- bietet verschiedene Funktionalitäten zur Paketaufzeichnung und -analyse (z.B. I/O-Graph)
- nutzerfreundliches grafisches Interface (GUI) und Werkzeuge zur Datenfilterung bzw. -analyse
- Live- und Offline-Untersuchung
- kann verschiedene interessante Protokolle wie z.B. SMTP (Simple Mail Transfer Protocol), TCP (Transmission Control Protocol), TLS (Transport Layer Security) aufzeichnen

3.4.4.2 mitmproxy

- Man-in-the-Middle-Prinzip: Man-in-the-Middle ist eine Methode Netzwerkdatenstöme mitzulesen. Dabei gibt man dem Rechner vor der Server zu sein, mit dem eine Verbindung geschaffen werden soll. Dadurch lassen sich Daten des Servers und des PC einsehen [23]. Darüber hinaus können Daten zwischen zwei Kommunikationspartnern weitergeleitet und kontrolliert (ändern, löschen oder hinzufügen) werden, sodass geglaubt wird, dass die Kommunikation über eine private und sichere Verbindung abläuft [36].
- Dabei ist mitmproxy, wie der Name schon sagt, als Man-in-the-Middle Proxy geschaltet, und stellt damit einen aktiven Eingriff in den Datenstrom dar. Denn im Gegensatz zu Wireshark, welches nur passiv auf den Netzwerkverkehr schaut, befindet sich Mitmproxy direkt im Verkehr. Bei der Analyse der Phishing-Mails bzw. Webseiten kann man nur HTTPS-Anfragen abfangen. Außerdem erhält man bei der Benutzung von mitmproxy die Zertifikate, die zum Mitlesen des Datenverkehrs notwendig sind. Die aufgezeichneten Daten und Informationen können danach wiedergegeben werden. Dabei liegen zu jeder Verbindung zusätzliche Angaben vor. Dies erlaubt eine genaue Analyse des Netzwerkverkehrs.

Gemäß der Quelle [37] sind die folgenden Funktionalitäten von mitmproxy genannt, die wir auch durch unsere Analyse mit mitmproxy gemerkt haben:

- Aufnahme und Wiedergabe von HTTP-Traffic
- HTTP- und HTTPS-Anfragen (Requests) und -antworten (Responses) abfangen und spontan ändern
- hat eine webbasierte Schnittstelle (mitmweb)
- ermöglicht interaktive Untersuchung und Änderung des HTTP-Datenverkehrs

3.4.5 Untersuchung des Arbeitsspeichers [Felix]

In dem Arbeitsspeicher kann man sowohl aktive Cookies erkennen als auch ob zusätzliche Programme im Hintergrund laufen die von der Mail/Website aufgesetzt bzw gestartet wurden. Also zum Beispiel ob sich Malware im System befindet und welche Prozesse diese gestartet hat. In der Theorie kann man auch den Netzwerkverkehr auslesen (ref hier). Oder zumindest die Verbindungen die im Moment des Dumps vorlagen.
Diese Punkte lassen Untersuchungen am Arbeitsspeicher recht attraktiv aussehen. Dennoch ließen sich aufgrund zahlreicher Probleme wenige nutzbare Daten aus dem Arbeitsspeicher gewinnen. Eines der Probleme ist die große Menge an irrelevanten Daten diesem. Obwohl eine rauscharme Umgebung verwendet wurde, ist der Anteil an Daten die von Interesse sind recht gering. Das heißt man diese wenigen Teile herausfiltern, was aufgrund der Datenmenge entweder sehr zeitaufwendig oder ungenau ist. Zum anderen sind Informationen in anderen Programmen leichter auslesbar und zu analysieren. So verwenden wir Beispielsweise Mitmproxy und Wireshark um den Netzwerkdatenstrom auszulesen. Ähnlich verhält es sich mit den Cookies.
Eine Analyse des Arbeitsspeichers wird über eine Funktion der Virtual Machine ermöglicht, dem “dump core”. Hierbei wird der gesamte Arbeitsspeicher im Moment des Aufrufs in ein Dump File exportiert. Dieser kann dann weiter analysiert werden durch verschiedenste Programme, wir haben uns für “Strings” [38] entschieden. Danach ist es einfach ein Suchspiel, bei dem der resultierende Text per Hand auf interessante Daten untersucht werden. Dabei hilft es sich Notizen zu der Aufzeichnung anzusehen und gezielt nach Stichwörtern oder ähnlichem zu suchen.

3.4.6 Untersuchung des Massenspeichers von Firefox, Thunderbird und Evolution [Felix]

Für die Untersuchung des Massenspeichers wird direkt in der Virtual Machine gearbeitet. Denn hier werden alle betroffenen Ordner und Dateien kopiert und getrennt abgespeichert. Die für uns wichtigen Aspekte sind zum einen die Speicherung der Cookies, der Mails (nur Evolution und Thunderbird) und die zugehörigen Metadaten (Header).

3.4.6.1 Cookies

Die Untersuchung ob und welche Cookies sich im System abgelegt haben geschieht bei Thunderbird und Firefox nahezu identisch. Bei beiden befinden sich alle Cookies in SQLite Datenbanken, die mit entsprechenden Tools geöffnet werden können. Tatsächlich sind aber alle Daten die sich in den Firefox bzw. Thunderbird Cookie-Datenbanken befinden auch über Firefox und Thunderbird selbst einsehbar. Trotz langer Suche haben wir für Evolution keinen Ort der Datenspeicherung bezüglich Cookies auffinden können. Daher war eine Untersuchung der Cookies bei Evolution nicht möglich. Dies ist allerdings kaum ein Problem, da sich in allen anderen Punkten die Daten von Thunderbird und Evolution sehr ähnlich sind. Daraus folgt dass höchstwahrscheinlich auch keine zusätzliche Informationsgewinnung über Thunderbird in Evolution möglich ist.
Wichtig ist auch zu beachten, das sich nur Cookies in Datenbanken befinden, die noch nicht abgelaufen sind. Denn wenn Cookies abgelaufen sind werden sie gelöscht. Außerdem existieren Beispielsweise Cookies welche nur während sie Aktiv sind abgespeichert werden. Diese sind demnach nicht einsehbar in den Datenbanken, insofern die Datenbanken nicht während der Aktivität des Cookies gesichert werden.

3.4.6.2 Mail und Metadaten

In Thunderbird befinden sich Mails und die zugehörigen Daten wieder in einer SQLite Datenbank. Zusätzlich existiert auch eine Klartext-Datei, in der sich alle Mails mit deren respektiven Header befinden. Da die Arbeit in Textdateien aber schwieriger als mit Datenbanken ist, haben wir uns auf die schon bekannten SQLite Datenbanken beschränkt. Von diesen kann man die Header-Daten auslesen und den Inhalt der Mail selbst. Auch in Evolution werden E-Mails lokal gespeichert. Dabei legt Evolution für jede Mail ein Verzeichnis an in dem sich mindestens eine Textdatei befindet. Diese enthält alle Informationen über die E-Mail (Header, Absender, Empfänger, . . .). Wie aber in dem vorgehendem Abschnitt schon beschrieben, die Arbeit mit Textdateien nicht praktikabel.
Sehr früh in unserer Arbeit hat sich die Frage ergeben, welchen vorteilhaften Eigenschaften die Untersuchung des Massenspeichers für unsere Fragestellung hat. Dadurch das alle Daten die sich in den Datenbanken und Textdateien befinden auch über die E-Mail Clients und Firefox einsehbar sind, haben wir die Untersuchungen zurückgestellt. Das heißt sie wurden nur auf Auffälligkeiten wie Unterschiede zu den Anzeigen in den Programmen untersucht.

Tabelle 1: Datentypen

Datenart	Lokalisiert
Raw Data (DT1)	im Arbeitsspeicher-Dump
Details über Daten (DT3)	E-Mail Header und mit Wireshark aufgenommen
Communication Protocol Data (DT5)	TCP (Wirshark) und HTTP/HTTPS (mitmproxy)
Process Data (DT6)	Cookies (Massen- und Arbeitsspeicher und Wireshark)

4 Implementierung [Gina]

Im folgenden Abschnitt gehen wir auf die Datentypen ein, die wir während unserer Untersuchungen gefunden haben. Darüber hinaus werden Phishing-Mails kategorisiert. Wir beschreiben, wie wir ein E-Mail System mit synthetischen Daten eingerichtet haben und wie eine rauscharme Umgebung aufgebaut wurde. Als nächstes schreiben wir über die Umsetzung der Datenakquise und wir vergleichen mitmproxy und Wireshark. Danach werden die E-Mail Header analysiert. Es folgen weitere Analysen von Netzwerkdatenströmen der verschiedenen Arten von Phishing-Mails und eines Newsletters mit Hilfe von Wireshark und mitmproxy. Anschließend werden sowohl der Arbeitsspeicher, als auch der Massenspeichers von Firefox, Thunderbird und Evolution untersucht. Am Ende folgt ein kurzer Vergleich der Phishing-Mails und des Newsletters.

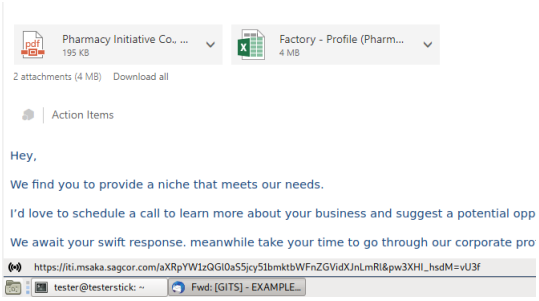
4.1 Data Types [Gina]

Raw Data haben wir im Arbeitsspeicher gefunden.
Details about Data haben wir im E-Mail Header gefunden und die Zeiten während der Netzwerkpakete in Wireshark aufgenommen wurden.
Communication Protocol Data (DT5) konnten wir mit Hilfe von Wireshark und mitmproxy ausfindig machen. Dazu gehören zum Beispiel TCP streams, die wir mit Wirehsark gefunden haben und HTTP/HTTPS Daten, die wir mit mitmproxy finden konnten.
Process Data sind zum Beispiel Cookies, die wir im Massen- und Arbeitsspeicher und mit Wireshark betrachtet haben.
In Tabelle 1 sind die für unsere Untersuchungen relevanten Datenarten nach Kiltz [39] gelistet.

4.2 Kategorisierung von Phishing-Mails [Vincent]

Im Zuge der Untersuchungen ist es natürlich wichtig, geeignete E-Mails zu finden, für welche passende Analysen durchgeführt werden können. Dementsprechend gilt es, die Phishing-Mails in verschiedene Kategorien zu gliedern.
Da die Untersuchung von Phishing-Mails in Verbindung mit E-Mail-Clients wie Thunderbird ein besonderer Schwerpunkt dieses Projekts ist, sind vorrangig E-Mails interessant, die keinen leeren „body“ haben. Solche E-Mails, die HTML-Elemente wie Bilder enthalten , können Tracker enthalten [40], welche auch theoretisch für den E-Mail-Client sichtbar sind - und somit Bedeutung für unsere zentrale Fragestellung besitzen.
Ein Großteil der von uns untersuchten Phishing-Mails entsprach dieser Kategorie. Jedoch können diese E-Mails erneut untergliedert werden und zwar in zwei Kategorien: „Open Redirect“ [41] E-Mails, welche bekannte Domänen wie Google nutzen, um Nutzer auf die eigentlichen gefährlichen Webseiten weiterzuleiten und E-Mails, welche „URL Obfuscation“ [42] nutzen, um den Nutzer überhaupt nicht sehen zu lassen, welche Links er besucht. Beispiele dazu sind jeweils in Abbildung 3a und Abbildung 3b zu sehen.
Als letzte Kategorie gibt es noch E-Mails, die zwar einen leeren „body“ besitzen, jedoch eine HTML-Datei im Anhang besitzen. Solche E-Mails müssen keine Links oder Bilder, welche Tracker übermitteln könnten, besitzen. Stattdessen beginnt der Angriff, sobald der Nutzer die HTML-Datei im Anhang öffnet. Wie in Abbildung 4 zu sehen ist, lässt der Verfasser der E-Mail den Empfänger glauben, dass es sich bei der Datei lediglich um eine Zahlungsinformation hält. In Wirklichkeit öffnet die Datei jedoch die Interaktion mit einer anderen Webseite.

(a) Beispiel für eine Open-Redirect E-Mail.



(b) In dieser E-Mail ist eine Bilddatei enthalten, welche den Nutzer glauben lässt, eine pdf- oder xlsx-Datei zu öffnen. Tatsächlich öffnet ein Klick auf das Bild jedoch einen Link zu einer Webseite.

Abbildung 3: Kategorisierung von Phishing-Mails

Tabelle 2 bildet abschließend einen kurzen Überblick über die Anzahl vorhandener E-Mails und in welche Kategorie diese zuzuordnen sind.

Tabelle 2: Anzahl erfasster und analysierter Phishing-Mails mit entsprechender Kategorie.

Anzahl	Kategorie
Open Redirect	2
Url Obfuscation	3
HTML Attachement	1

Abbildung 4: Beispiel für eine E-Mail, die eine HTML-Datei im Anhang enthält. Konventionelle Virenschutzprogramme wie Microsoft Defender werden davor warnen, diese Datei herunterzuladen.

4.3 Einrichten eines E-Mail-Systems mit synthetischen Daten [Bernhard]

Damit wir möglichst schnell ein breites Spektrum von Phishing-Mails erhalten, haben wir zu Beginn dieser Projektarbeit insgesamt sechs E-Mail-Adressen bei verschiedenen Mail-Providern [43] erstellt. Später hat sich aufgrund der sehr geringen Anzahl von erhaltenen Phishing-Mails ergeben, dass die Weiterleitung aller für unseren Zweck interessanter Mails auf eine E-Mail Adresse sinnvoll ist. Das heißt, dass wir alle Aufzeichnungen auf ein und demselben E-Mail-Account durchgeführt haben.
Während unseren ersten Untersuchungen hat sich ergeben, dass Thunderbird im Vergleich zu Evolution sehr viel besser für die Mass-Storage-Analyse geeignet ist. Beispielsweise speichert Evolution die Anwendungsdaten in mehreren unterschiedlichen Verzeichnissen verteilt ab [44]. Zusätzlich sind die Daten der E-Mails in einer HashMap auf dem Dateisystem organisiert, was das gezielte Auslesen der Daten erschwert (mehr dazu in Abschnitt 4.10.3). Aus diesem Grund beschränken wir uns bei der Analyse auf Thunderbird. Für die Verbindung von Thunderbird mit unserer Phishing-Mailadresse haben wir die in Abbildung 5 zu sehenden Einstellungen verwendet.

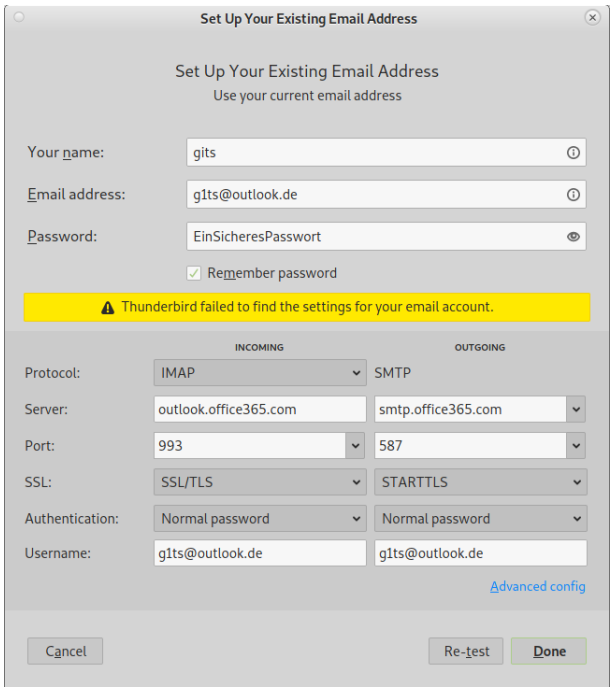


Abbildung 5: Einstellungen in Thunderbird zum Verbinden mit dem von uns für die Untersuchung genutzten E-Mail-Postfach.

4.4 Aufbau einer rauscharmen Umgebung [Bernhard]

Der theoretische Aufbau der Systemlandschaft wurde bereits in Punkt 3.2.1 beschrieben. Um diese geforderten Eigenschaften umsetzen zu können, haben wir als Basis den gestellten Testerstick der Universität für datenforensische Untersuchungen mit einer Debian-Linux-Distribution (11.3.0-amd64) [45] verwendet, um darauf aufbauend eine virtuelle Maschine (VM) mit Hilfe von Oracle VirtualBox aufzusetzen (ebenfalls Debian-Linux Live-System). VirtualBox ist dabei bereits auf dem Testerstick vorhanden.
Innerhalb der VM werden nun die Tools zur Datenakquise eingesetzt (Firefox ESR, Wireshark, mitmproxy, Thunderbird und Evolution). Das Setup innerhalb der virtuellen Maschine wird von unserem Script zur Aufzeichnungsmethodik (Anhang C.1) weitestgehend übernommen, wodurch die benötigten Tools automatisch auf der VM installiert werden.

4.5 Umsetzung der Datenakquise [Bernhard]

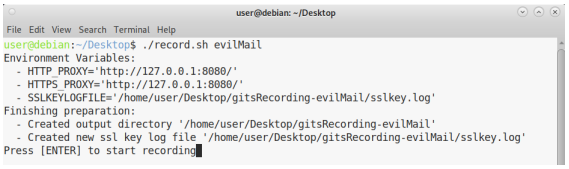
Um eine vergleichende Auswertung der Untersuchungsergebnisse zu ermöglichen ist es sinnvoll, alle Aufzeichnungen in einem einheitlichen Format mit gleicher Struktur systematisch durchzuführen. Deshalb haben wir uns entschlossen, den wesentlichen Teil der Aufzeichnungsmethodik (Abbildung A.1) in der virtuellen Umgebung als Bash-Script (Anhang C.1) zu implementieren.
Teile der Untersuchung können allerdings nicht automatisiert umgesetzt werden (z.B. Speichern der Wireshark-Aufzeichnung) oder finden nicht in der virtuellen Umgebung statt (z.B. RAM-Dump der VM) und können deshalb nicht über das Script gesteuert werden.
Das Script benötigt genau einen Parameter zur späteren eindeutigen Zuordnung des Untersuchungsgegenstands zu

den Aufzeichnungsergebnissen bei der Analyse (z.B. Betreff der zu untersuchenden E-Mail oder Name der Phishing-Seite).

```
##################################################
# Config                                         #
##################################################

LINUX_HOME="/home/${USER}"

#Temporary output directory (will be wiped prior recording)
OUTPUT_DIRECTORY="${LINUX_HOME}/Schreibtisch/gitsRecording-${PHISHING_MAIL_IDENTIFIER}"

#Location of metadata
METADATA_LOCATION="${OUTPUT_DIRECTORY}/metadata.txt"

#Location of mitmproxy executable
MITMPROXY_EXECUTABLE="${LINUX_HOME}/opt/mitmproxy"

#Where to store keys and mitmproxy data (also save wireshark .pcapng file here)
SSLKEYLOGFILE_LOCATION="${OUTPUT_DIRECTORY}/sslkey.log"
MITMPROXY_OUTPUT="${OUTPUT_DIRECTORY}/mitmproxy.log"

#Where to find firefox sql stuff
FIREFOX_DBS="${LINUX_HOME}/.mozilla/firefox/9fwqaeub.tester"

#Where to store firefox sql stuff
FIREFOX_DBS_OUT="${OUTPUT_DIRECTORY}/firefox"

#Where to find thunderbird sql stuff
THUNDERBIRD_DBS="${LINUX_HOME}/.thunderbird/t6qhkkv7.default-default"

#Where to store thunderbird sql stuff
THUNDERBIRD_DBS_OUT="${OUTPUT_DIRECTORY}/thunderbird"

#Where to find evolution sql stuff
EVOLUTION_DBS="${LINUX_HOME}/.cache/evolution"

#Where to store evolution sql stuff
EVOLUTION_DBS_OUT="${OUTPUT_DIRECTORY}/evolution"
```

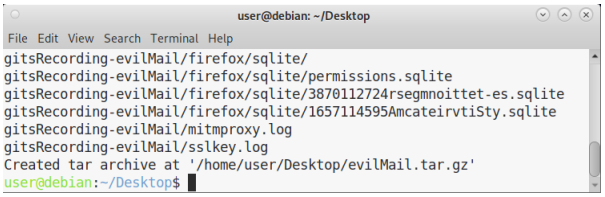
Abbildung 8: Konfigurationsbereich im Script. Einige Pfade wie z.B. das Thunderbird-Profil sind auf jedem System einzigartig und müssen dementsprechend angepasst werden.

1) **Prüfung der Systemumgebung**: Für eine reibungslose Aufzeichnung prüft das Script vor dem Start der Tools zunächst, ob die Systemumgebung korrekt aufgesetzt ist. Dies umfasst u.a. die Installation von mitmproxy, Wireshark, Thunderbird und Evolution, sowie eine Prüfung des Vorhandenseins des Firefox-Profils und der Thunderbird-Daten. Unter Umständen wird fehlende Software sofort installiert bzw. auf fehlende Daten hingewiesen. Eventuell müssen Verzeichnispfade im Config-Bereich des Scripts angepasst werden (Abbildung 8).
   Nach erfolgreicher Überprüfung werden die für die Aufzeichnung und anschließende Analyse benötigten Umgebungsvariablen gesetzt (SSLKEYLOGFILE [22], HTTP_PROXY, HTTPS_PROXY [25]) und ein Ausgabeverzeichnis erstellt.
   Das Script wartet nun auf den Nutzer, um mit der Aufzeichnung zu beginnen (Abbildung 6a).
2) **Aufzeichnung**: Nach Aufzeichnungsbeginn wird der aktuelle Timestamp gespeichert und alle benötigten Programme gestartet (Firefox ESR, Wireshark, mitmproxy, Thunderbird, Evolution). Der Zeitpunkt der Untersuchung ist insofern wichtig, da nach abgeschlossener Aufnahme nur die Anwendungsdaten betrachtet werden sollen, die sich während der Untersuchung geändert haben. Der Nutzer muss anschließend die Wireshark-Aufzeichnung manuell starten und kann danach seine Untersuchungen an den Phishing-Mails im Mail-Client bzw. an den Phishing-Links im Webbrowser durchführen. Die detaillierte Vorgehensweise dazu wurde bereits im Abschnitt 3.3 erklärt.
   Nach Abschluss der Untersuchungen in der virtuellen Umgebung, wenn alle Tools noch geöffnet sind, sollte ein Abbild des Arbeitsspeichers der virtuellen Maschine für spätere Analysen erstellt werden. Natürlich muss dies außerhalb der virtuellen Umgebung über VirtualBox mit folgendem Befehl aus Listing 1 passieren.

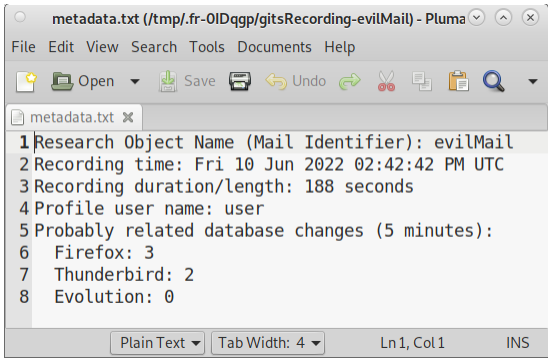


(a) Start des Scripts mit Beispielparameter und erfolgreicher Prüfung der Umgebung.

(b) Script wartet, bis alle von ihm gestartete Programme beendet wurden.



(c) Script hat SQL-Datenbanken aufbereitet und alle Informationen archiviert.



(d) Automatisch vom Script erfasste Metadaten.

Abbildung 6: Ablauf einer Aufzeichnung mit Script

```
vboxmanage debugvm "<MachineName>"
    dumpvmcore --filename
    example_dump
```

Listing 1: Dieser Befehl erstellt einen Dump des Arbeitsspeichers der laufenden VM als .elf-Datei.

Das Script wartet nun, bis alle gestarteten Tools ordnungsgemäß beendet wurden (Abbildung 6b). Mitmproxy speichert seine Aufzeichnungen automatisch ins Ausgabeverzeichnis, die Wireshark-Aufzeichnung muss vor dem Beenden allerdings manuell gespeichert werden.

3) **Speichern der gesammelten Daten**: Nach Beendigung aller Programme wird erneut der aktuelle Timestamp verwendet, um die Dauer der Aufzeichnung zu bestimmen. Mit dieser Information werden nun alle Datenbanken von Thunderbird und Firefox ESR in das Ausgabeverzeichnis kopiert, die sich in der Zeitspanne der Aufzeichnung geändert haben. Anschließend werden die Tabellen dieser direkt mit sqlite3 [24] in Textdateien umgewandelt, um die spätere Analyse zu vereinfachen. Vor der abschließenden Archivierung der Untersuchungsergebnisse als .tar.gz (Abbildung 6c) werden zunächst noch einige Metadaten wie Aufzeichnungszeitpunkt, -dauer oder der Name des Nutzerprofils gespeichert (Abbildung 6d).

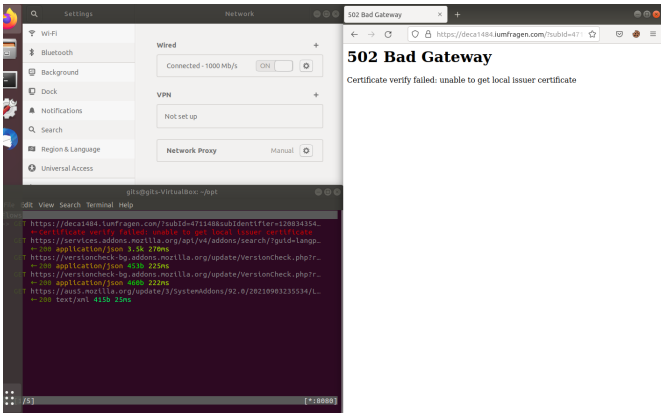
4.6 Vergleich mitmproxy/Wireshark [Abdalla]

4.6.1 Wireshark

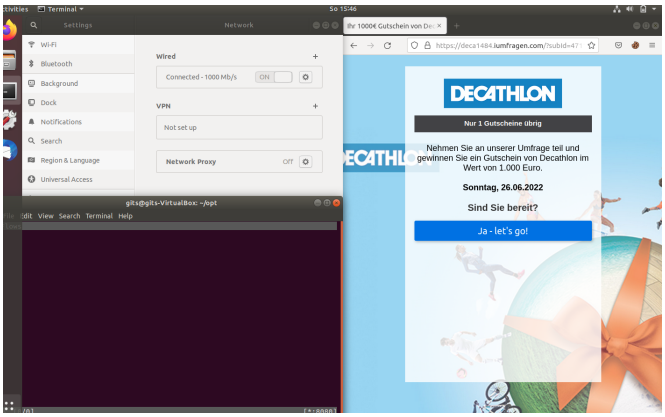
Da wir verdächtige und bösartige Phishing-Mails überprüfen, sowie den Verkehrsstrom auf unserem System untersuchen, ist es für uns nützlich und sinnvoll Wireshark (wie im Abschnitt 3.4.4) zu nutzen. Außerdem werden verschiedene Arten unserer gefundenen Phishing E-Mails durch Wireshark untersucht.
Abbildung B.3 zeigt, wie man die erstellte SSLKEYLOGFILE aus der gespeicherte Aufzeichnung auswählen kann, um die Pakete zu entschlüsseln. Ab diesem Punkt werden dann die SSLKEYLOGFILEs mit Wireshark geöffnet und alle Kommunikationen nach dem Öffnen von Phishing E-Mails, die in unserem Netzwerk ausführlich untersucht.

4.6.2 mitmproxy

Um den Netzwerkverkehr aufzuzeichnen, lassen wir mitmproxy auf unseres automatisches Proxy in Netzwerkeinstellungen auf "localhost" bzw. Port "8080" laufen. Außerdem muss das vom mitmproxy erstellte Zertifikat in Firefox installiert werden [25].
Jedoch ergab sich während unserer Arbeit mit Mitmproxy das Problem, das sich einige Webseiten nicht öffnen ließen. Dies liegt an dem Zertifikat, welches Mitmproxy zur Verfügung stellt. Es wird nicht anerkannt von diesen Webseiten, bzw es kann nicht verifiziert werden. Dieses Problem hat sich auch nicht beheben lassen, sodass einige Aufzeichnungen und Analysen ohne die Daten von Mitmproxy durchgeführt wurden (Abbildung 7a und 7b).



(a) Phishing-Webseite nicht funktional mit mitmproxy aktiv.



(b) Phishing-Webseite funktioniert, wenn mitmproxy nicht Teil des Datenstroms ist.

Abbildung 7: Funktionalität einer Phishing-Seite mit und ohne mitmproxy

4.7 Analyse der E-Mail Header [Felix]

Wie in Punkt 3.4.1 beschrieben kann man die Analyse mittels Hilfsmitteln lösen, die einem beim Verständnis helfen. Wir haben Header mithilfe von dem Message Header Analyzer überprüft. Dabei wurden die Header Dateien in Thunderbird geöffnet, der gesamte Text kopiert und auf dieser Website eingefügt. Die Anzahl der E-Mails, die wir hier untersuchen konnten war wesentlich größer, als die in der Website-Analyse, da E-Mail Header erstellt werden während des Versendens. Damit sind sie unabhängig von dem Status der dahinterliegenden Website. Dies wiederum heißt das auch die Untersuchung von Mails, deren dahinterstehende Phishing-Webseiten offline sind, sinnvoll ist. Wie bereits erwähnt betrachten die meisten Mailprovider die Header und sortieren E-Mails anhand dieser in z.B. dem Spam-Ordner ein. Ein Beleg dafür findet sich in 9 Hierin kann man deutlich erkennen, dass beispielsweise Outlook Header analysiert und dann in Bezug auf verschiedene Eigenschaften eine Bewertung vergibt. Hier hat diese Mail eine PCL (phishing confidence level) Bewertung von 2 bekommen, was bedeutet dass es sich wahrscheinlich um keine Phishing-Mail handelt [46]. In diesem Fall ist die Bewertung allerdings falsch, was auch illustriert, dass diese Filter nicht alles korrekt identifizieren. Ohne zu wissen ob ein Server, über den die Mail geleitet/entsendet wurde, gibt es in den Headern keine klare Anzeichen dafür, dass es sich um eine Phishing-Mail handelt. Dies war zu erwarten, da diese Mails den Filter der Mailprovider überwunden haben, was nicht der Fall gewesen wäre, hätte es klare Indizien für gefährliche Inhalte oder betrügerische Absichten gegeben.

49	X-MS-Exchange-Organization-PCL	2
50	X-MS-Exchange-Organization-SCL	9
51	X-MS-Exchange-CrossTenant-OriginalArrivalTime	17 May 2022 15:59:40.8441 (UTC)
52	X-MS-Exchange-CrossTenant-Network-Message-Id	49987994-fa4b-4689-1aa8-08da381e408b
53	X-MS-Exchange-CrossTenant-Id	84df9e7f-e9f6-40af-b435-aaaaaaaaaaaa
54	X-MS-Exchange-CrossTenant-RMS-PersistedConsumerOrg	00000000-0000-0000-0000-000000000000
55	X-MS-Exchange-CrossTenant-rms-persistedconsumerorg	00000000-0000-0000-0000-000000000000
56	X-MS-Exchange-CrossTenant-AuthSource	DB8EUR05FT023.eop-eur05.prod.protection.outlook.com
57	X-MS-Exchange-CrossTenant-AuthAs	Anonymous

Abbildung 9: Ausschnitt aus Header Analyse

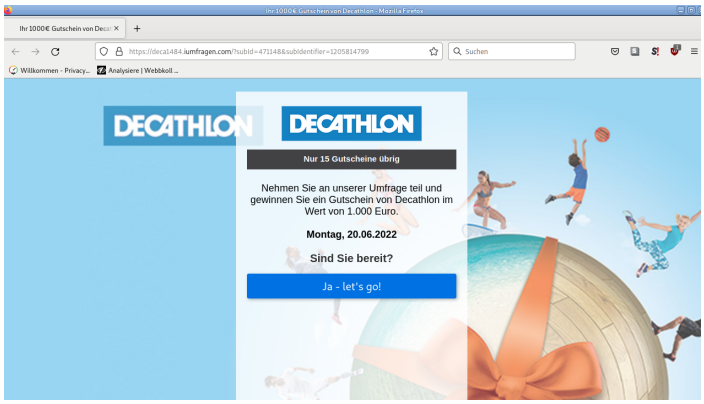
4.8 Analyse des Netzwerkdatenstroms mit Wireshark und mitmproxy [Vincent]

Im folgenden Abschnitt sollen nun konkret die Untersuchungen, wie sie in 3.4.2 bzw. 3.4.3 konzeptionell beschrieben wurden, anhand einiger Beispiele erklärt werden. Dabei ist anzumerken, dass die Analyse fast ausschließlich mit Wireshark und nicht mit mitmproxy durchgeführt wurde, auch wenn von beiden Anwendungen entsprechende Aufzeichnungsdaten erstellt wurden. Ebenso gilt, dass die IP-Adressen, welche im Zuge der Untersuchungen gefunden wurden, mit Hilfe der Webseite „ipinfo“ [30] untersucht wurden.

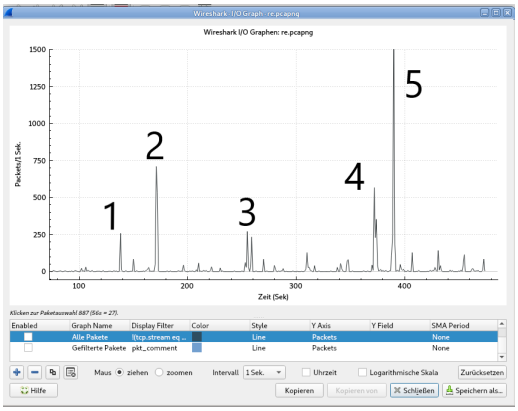
4.8.1 Beispiel für Url-Obfuscation [Vincent]

Bei der folgenden E-Mail handelt sich um eine Umfrage, die einen Gutschein als Belohnung verspricht, wie in Abbildung 10a zu sehen ist. Interessant ist hier der beigefügte Link „https://storage.googleapis.com/cezayin/maurisio.html#lokWX.aspx?fgb7YnccNR39cx81ncdcKPcKc4mdTgvD9cbbb4V“. Es handelt sich hierbei also um eine HTML-Datei, welche über die Google Cloud gespeichert ist und danach eine neue Webseite öffnet, sobald sie geöffnet wird.
Auf der sich nun geöffneten Webseite, welche in Abbildung 10b zu sehen ist, wird der Nutzer durch eine kurze Umfrage geführt, an deren Ende er sich einen Gutschein aussuchen kann, der anfangs versprochen wurde.
Ein Blick in den I/O Graphen, wie er in Abbildung 10c zu sehen ist, gibt einen guten Überblick darüber, welche Aktionen zu welchen Kommunikationen geführt haben.
Nach dem Öffnen der Seite bildet sich nach kurzer Zeit unser erster Anstieg, sobald wir bestätigen, uns an einer Umfrage zu beteiligen. Zum Beispiel existieren Verbindungen zu einer israelischen Telekommunikationsgesellschaft [47] und einem kanadischen Hosting-Unternehmen [48], wobei beide Verbindungen dazu bestehen, Bilddateien zu übermitteln. Ebenso existieren Verbindungen zu „zonestor.com“ und „deca1484iumfragen.com“, welches dem Name der Webseite an sich entspricht. Beide diese Adressen lassen sich auf eine spanische Domäne zurückverfolgen [49] und übermitteln Cookies, Scripte und CSS-Dateien.
Der zweite Anstieg korreliert mit dem Übergang zu einer weiterführenden Seite, in welcher der Nutzer Daten eingibt. Eine neue Verbindung mit der Adresse „dec-voucher.gewinn24-7.de“ öffnet sich, welche der vorherigen spanischen IP entspricht [49]. Ebenso öffnen sich mehrere Verbindungen, die zu Google führen [50], u.a. „gstatic.com“ und „fonts.googleapis.com“. Die spanische Adresse übermittelt dabei erneut einige Cookies, Scripte und CSS-Dateien. Interessant ist hier, dass „referrer“ für die Cookies die vorher erwähnte, spanische IP [49] ist.
Der dritte Anstieg entspricht der Eingabe von Namen, E-Mail und anderen personenbezogenen Daten. Hier existieren keine neuen Verbindungen, sondern es bestehen die gleichen Verbindungen wie im vorherigen Anstieg. In Abbildung 10d ist zu sehen, wie die Daten an die Seite über HTTP übermittelt werden.
Als nächstes folgt die Eingabe des Wohnortes, welche dem vierten Anstieg entspricht. Diese Eingabe erfolgte insgesamt dreimal, da die Seite die ersten beiden Eingaben nicht akzeptierte. Mit korrekter Eingabe öffnet sich eine neue Verbindung zu der Adresse „pixel.pixelweiche.de“, welche der IP Adresse von Amazon entspricht [51]. Über diese Verbindung wird ein Script übermittelt, welches auf einem Server von Amazon gespeichert ist. Ebenso werden die Dateneingaben wieder an die spanische Adresse übermittelt.
Der letzte Anstieg folgt dem Öffnen einer neuen Seite, nachdem der Nutzer alle Angaben erfolgreich eingegeben hat. Es öffnet sich eine neue Verbindung zur Seite „media.deutsche-familienversicherung.de“ [52], welche der Adresse eines deutschen Unternehmens für

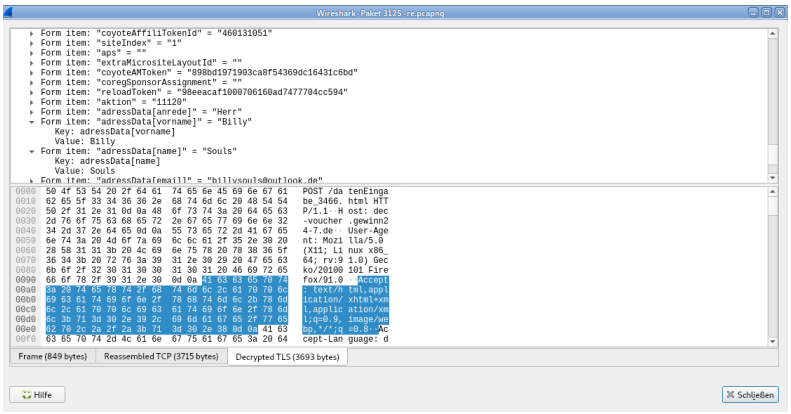
(a) Sobald der Nutzer auf den Button klickt, wird eine HTML-Datei geöffnet, welches den Nutzer zur eigentlichen Webseite weiterleitet.



(b) Die erste Seite, die der Nutzer nach Öffnen des Links sieht.



(c) Der I/O Graph, welcher von Wireshark erstellt wurde. Die Zahlen wurden manuell selbst eingefügt, um einen besseren Überblick zu gewähren.



(d) Ein Ausschnitt aus der Übermittlung der personenbezogenen Daten an die Phishing-Webseite.

Abbildung 10: Analyse der Decathlon-Mail

Wassertechnologie entspricht [53]. Von dieser Seite werden lediglich einige Bilddateien übermittelt.
Im Hinblick auf die Fragestellung, inwiefern Dienstanbieter Phishing-Attacken wahrnehmen können, ist diese E-Mail in einigen Punkten interessant.
Zuerst besteht der Fakt, dass die HTML-Datei, die den Nutzer auf die Phishing-Webseite führt, auf dem Cloud-Service von Google selbst gespeichert wird. Google selbst sollte also durchaus in der Lage sein, wahrzunehmen, welche Daten von Nutzern auf der Cloud gespeichert werden und inwiefern diese missbraucht werden könnten.
Ebenso existiert im Verlaufe der Verbindung ein Datenaustausch eines Scripts über einen Server von Amazon. Auch hier ist anzunehmen, dass Amazon in der Lage ist, einzusehen, welche Daten auf den eigenen Servern vorliegen bzw. verarbeitet werden.
Schlussendlich ist es auch fraglich, wieso Verbindungen zu „fonts.googleapis.com“ und „gstatic.com“ bestehen. Erneut besteht eine berechtigte Annahme, dass Google die Fähigkeit besitzt, wahrzunehmen, welche dubiosen Seiten Dienste von Google beanspruchen. In all jenen genannten Punkten besteht also eine berechtigte Annahme, dass die jeweiligen Dienstanbieter Kenntnisse über die Phishing-Angriffe besitzen.

4.8.2 Beispiel für HTML-Attachment [Michelle]

Bei der im Folgenden zu analysierenden E-Mail handelt es sich um eine Phishing E-Mail mit einem HTML-Attachment. Das bedeutet, dass es keinen Link zu einer Phishing-Webseite im E-Mail Körper gibt, da Spamfilter diese meist leicht aussortieren. Stattdessen wird eine HTML-Datei als Anhang an die E-Mail beigefügt [54], wie in Abbildung 11 zu sehen ist. Hier wird dem Empfänger vorgemacht, dass im Anhang eine Zahlungsbestätigung zu finden sei. Da die E-Mail keinen Hinweis darauf gibt, welche Zahlung gemeint ist, wird man leicht dazu verleitet den HTML-Anhang zu öffnen.

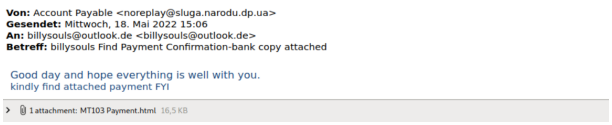


Abbildung 11: E-Mail mit HTML-Attachment

Bei einem ersten Blick auf die Hostnamen der IP-Adressen fällt auf, dass es viel Kommunikation mit den Adressen „i.ibb.co“ und „theoysteracademy.org“ gibt. Dabei fällt bei „i.ibb.co“ auf, einer Domain, die ihren Sitz in

Kolumbien haben sollte, dass diese beim Wechsel zwischen dem Hostnamen und der IP-Adresse zwei unterschiedliche IP-Adressen aufweist. Verfolgt man die Erste (51.210.3.236) über IPinfo [55], findet man heraus, dass diese zur Firma OVH SAS [56] führt. Dies ist eine französische Firma, die sich auf Hosting spezialisiert hat. „theoysteracademy.org“ ist auf IPinfo unter der Firma „websitewelcome.com“ eingetragen [57]. Gibt man diese in eine Suchmaschine ein, findet man relativ schnell heraus, dass es sich dabei um eine Phishing-Webseite handelt [58].
Schauen wir uns nun die Datenpakete der Wireshark-Aufzeichnung genauer an. Zur besseren Übersicht ist in Abbildung 12 der I/O-Graph dargestellt, der während der Aufzeichnung entstanden ist und uns einen Überblick darüber verschafft, welche Interaktion mit der Webseite zu welchem Anstieg der Datenpakete geführt hat. Zur leichteren Erkennung wurden alle Anstiege nummeriert.

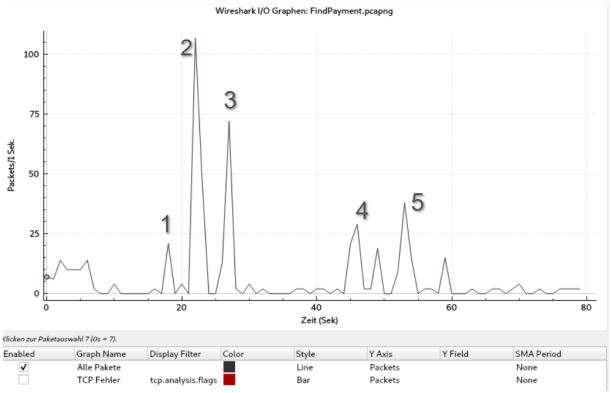


Abbildung 12: Der I/O Graph in Wireshark, nummeriert für einen besseren Überblick

Klickt man nun auf den HTML-Anhang, wird man zu der Phishing-Webseite weitergeleitet. Das ist im I/O-Graph der Anstieg 1. In diesem Fall führt der Link, wie in Abbildung 13 zu sehen, zu einer Webseite, die vorgibt, OneDrive zu sein. Hier ist bereits die E-Mail Adresse des Absenders eingetragen. Dies ist keine echte OneDrive Seite, sondern wird nur als solche ausgegeben, um den Nutzer dazu zu bringen, dort seine Daten einzugeben. Wir erkennen die Fälschung daran, dass bei dem echten Anmeldefenster für OneDrive oben links das Microsoft Logo zu sehen ist, und, nicht wie hier, das OneDrive Logo mit dem OneDrive Schriftzug.
Seit dem Öffnen des Anhangs gibt es Kommunikationen mit Amazon über TCP, sowie Cloudflare, max-cdn.bootstrapcdn.com, i.ibb.co und Microsoft.

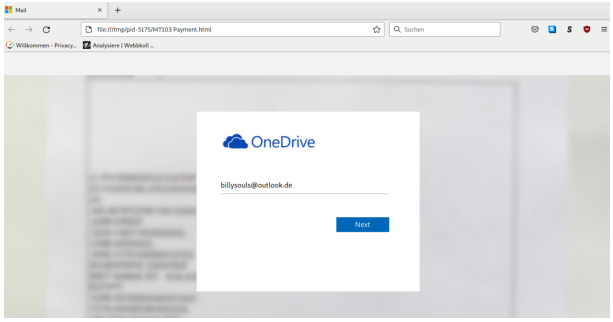


Abbildung 13: Wird der HTML-Anhang geöffnet, öffnet sich die gefälschte OneDrive Seite

Bei Cloudflare (IP: 104.17.24.14) findet man GET Methoden für „popper.min.js“ und „popper.js“, „max-cdn.bootstrapcdn.com“ lädt eine „bootstrap.min.js“.

Der größte Anstieg, der im Graph zu sehen ist, kann zu keiner Interaktion mit der Phishing Seite zugeordnet werden. Hier werden vor allem sehr viele Pakete mit Google Fonts übertragen (fonts.googleapis.com). Hier findet man jquery.min.js mit einer GET Methode. Es gibt außerdem Kommunikation über eine uns bekannte Adresse, „i.ibb.co“. In ihren Datenpaketen lassen sich GET Methoden für .png- und .jpg-Dateien finden. Die Microsoft Kommunikation ist dazu da, um Bilder zu hosten.
In Anstieg 3 wurde mit der vorgeschlagenen E-Mail Adresse auf „Next“ geklickt, ohne, dass die Eingabe eines Passworts erfolgte. Hier findet man wieder Google APIs, in dem Fall „ajax.googleapis.com“. Außerdem lässt sich „pki-goog.l.google.com“ in den Paketen finden. Das ist Google's Public Key Infrastructure und soll die Kommunikation im Internet durch Verschlüsselung und Authentifizierung sicherer machen. [59]
Im 4. Anstieg wird zum ersten Mal ein falsches Passwort eingegeben („eindollespasswort“) und auf „Next“ geklickt. Die Verbindung führt zu „theoysteracademy.org“, von der wir oben schon herausgefunden haben, dass hinter ihr eine Phishing Seite steckt. Hier erkennt man im HTTP Netzwerkdatenstrom eine POST Methode. Es ist nicht viel entschlüsselt, aber die E-Mail und das Passwort, das wir eingegeben haben sind erkennbar, wie in Abbildung 14 zu sehen ist. Das bedeutet, dass sie woanders hingeschickt werden.

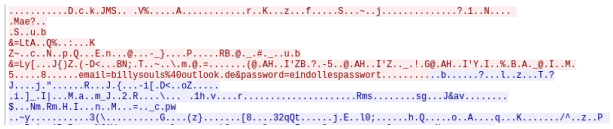


Abbildung 14: Im Stream erkennbar: E-Mail und Passwort

Der 5. Anstieg ist die erneute Eingabe des falschen Passworts. Hier passiert das gleiche wie auch im vorherigen Anstieg.
Direkt nach der 2. Eingabe des falschen Passworts, gibt es eine Verbindung zu „creativeagencysecrets.com“ (IP: 192.124.249.69). Diese wird benutzt um eine Fehlermeldung als PNG zu posten, wie in Abbildung 16 zu sehen ist. Der angezeigte 404 Error ist keine echte Fehlermeldung, sondern, wie schon erwähnt, nur ein Screenshot von einem

404 Error. Sobald die Seite also die Daten des Nutzers bekommen hat, wird vorgetäuscht, dass sie nicht mehr funktioniert.

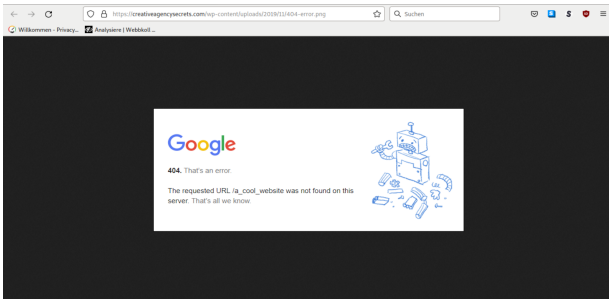
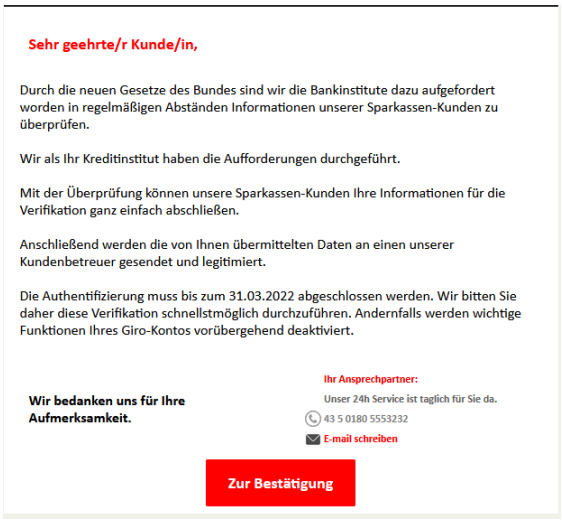


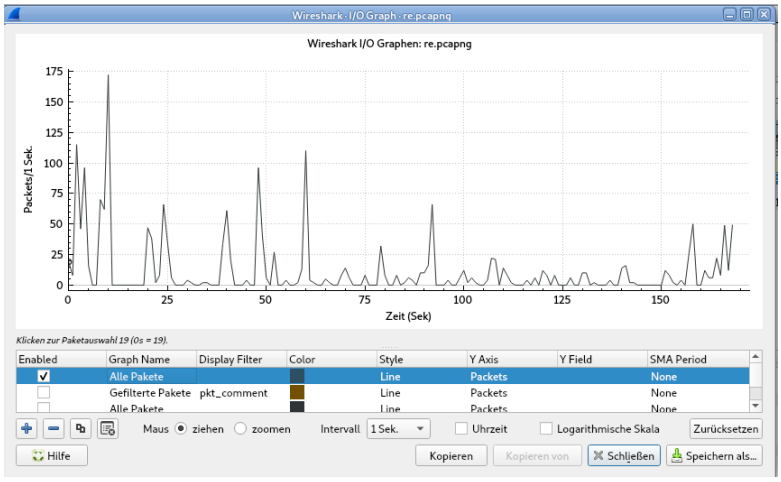
Abbildung 16: Nach zweimaliger falscher Eingabe des Passworts bekommt man einen vermeintlichen 404 Error

Als Fazit sehen wir, dass in den Datenpaketen dieser Phishing E-Mail einige namhafte Dienstanbieter, wie z.B. Amazon, Microsoft und Google, wiederzufinden sind.

Da wir die Dinge nur aus Sicht des Nutzers sehen, können wir daraus schließen, dass große Dienstanbieter mindestens die gleichen Sachen sehen können.
Zum einen wird eine Domain („theoysteracademy.org“), die als Phishing Seite bekannt ist, auf OVH SAS gehostet. Diese Firma ist nach eigenen Angaben weltweit auf Platz 3 im Bereich Hosting. Eine so großen Firma sollte die Ressourcen und Möglichkeiten besitzen, nachvollziehen zu können, was mit ihren Domains passiert und wozu diese benutzt werden.
Die bereitgestellten APIs von Google, die in die Webseite mit eingebunden waren, werden von Google selbst gehostet. Auch hier sollte theoretisch die Möglichkeit für Google bestehen, verfolgen zu können, in welche Art von E-Mails ihre APIs eingebunden sind.

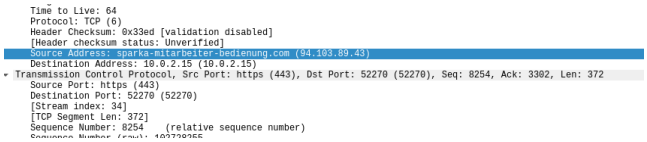


(a) Wenn der Benutzer auf den Button (Zur Bestätigung) klickt, wird von Google an Sparkasse-Webseite weitergeleitet.

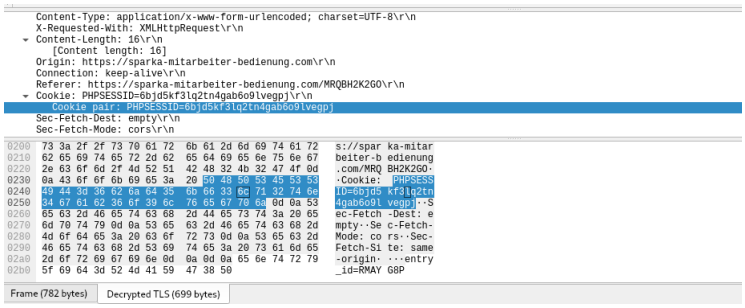


(b) Der von Wireshark erstellten I/O Graph, um einen besseren Überblick zu haben.

(c) Die erste Seite, die der Nutzer nach dem Öffnen des Links geöffnet.



(d) Ein Ausschnitt aus der Übermitllung der Source Address von der Webseite.



(e) Ein Ausschnitt aus der Übermitllung in Wireshark für die benutzte Cookies an die Webseite.

Abbildung 15: Analyse einer Open-Redirect-Mail

4.8.3 Beispiel für Open-Redirect [Abdalla]

Hier geht es um eine Phishing-Mail vom Typ „Open-Redirect“. Der Betrüger gibt sich in dieser Mail als Sparkasse (Bank) aus. Dabei fordert die Bank ihre Kunden auf ihre Daten zu aktualisieren bzw. zu verifizieren, da dies gesetzlich vorgeschrieben ist. Wer diese Daten in bestimmter Zeit nicht aktualisiert, dem wird sein Giro-Konto gesperrt, wie in Abbildung 15a zu sehen ist.
Sobald wir auf den Button aus Abbildung 15a klicken, wird eine Webseite geöffnet („https://www.google.mw/url?usg=AFQjCNGWiU5U0fbyorW_PGz9rAABR9I4Qg&sntz=1&sa=D&q=https://spk-itsicherheit2022.com/LQDJU9JB2W#vXeVDtz1N4A“), welche in Abbildung 15c zu sehen ist. Klickt man hier auf „weiterleiten“, öffnet sich die Phishing-Seite.
Mit dem I/O Graph in Abbildung 15b bekommt man eine gute Übersicht darüber, welche Aktionen zu welchen Kommunikationen geführt haben.
Nachdem man den Link geöffnet hat, wurde man auf die Phishing-Webseite weitergeleitet. Diese Webseite gab vor die Sparkasse Aachen zu sein. Auf dieser mussten anschließend Logindaten eingegeben werden. Nach hinzufügen der synthetischen Daten erkennt die Webseite, dass es sich um falsche Informationen handelt. Anschließend geht die Phishing-Webseite in eine Warteschleife. Danach wird die Webseite unempfänglich für weitere Eingaben und Aktionen.
Aus der Source-Address (Wireshark) in Abbildung 15d kann man eine IP Adresse erkennen. Das Eingeben dieser in eine IP-Untersuchungswebseite bringt uns zu einer russischen Kommunikationsgesellschaft [60], welche eingegebene Daten übermitteln. Es existiert auch eine neue Verbindung zu der Adresse “ocsp.quovadisglobal.com”, welche der IP Adresse von Amazon entspricht [61].
Abbildung 15e zeigt das cookie-pair, welche auf dem Browser ausgeführt wurden und in der Verbindung mit der Webseite stehen 4.8.3.
Als Fazit kann man sagen, dass Open-Redirects zur Entwendung sensibler Daten und ggf. zu finanziellen Verlusten des Opfers führen. Außerdem werden durch diese Weiterleitung(en) mehr Verbindungen zu anderen Websites augebaut, welche wiederum Tracker besitzen können.

4.8.4 Beispiel für einen normalen Newsletter [Gina]

Im Folgenden wird eine Newsletter E-Mail vom Spiegel analysiert, um die Unterschiede zwischen Phishing und nicht Phishing E-Mails herauszufinden und um sie im Anschluss zu vergleichen.
Auch hier werfen wir einen Blick auf die Datenpakete, die mit Hilfe von Wireshark aufgenommen wurden. Zum besseren Überblick ist in Abbildung 17 ein nummerierter I/O- Graph dargestellt, der uns anzeigt, welche Interaktion zu welchen Kommunikationen geführt haben. Mit einem Klick auf den Graphen können wir dann zu den jeweiligen Paketen navigieren. Die E-Mail beinhaltete einen Link zum Spiegel Newsletter. Das weiterleiten zum Newsletter resultiert in Anstieg 1 des IO-Graphen. Man landet auf einer Seite, die eine Übersicht an Beiträgen anzeigt. Die weiteren Anstiege entstanden durch das klicken auf die einzelnen Beiträge.

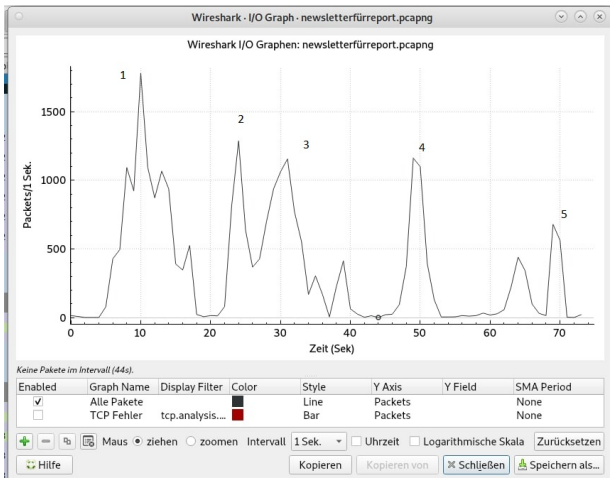


Abbildung 17: nummerierter I/O Graph aus Wireshark

Beim untersuchen der IP-Adressen ist fast nichts besonderes aufgefallen. Eine der Adressen mit der meisten Kommunikation ist: a1661.dscd.akamai.net (23.213.161.15). Laut IP.info gehört die IP-Adresse zu Akamai Technologies, Inc. Und hat ihren Standort in Frankfurt am Main [62]. Akamai sorgt dafür, dass eine schnellere Internetverbindung vorhanden ist und somit Beiträge schneller geladen werden [63]. Eine weitere Adresse mit viel Kommunikation ist ft2.map.fastly.net (151.101.114.209). Sie gehört zu Fastly, Inc und hat ebenfalls ihren Standort in Frankfurt am Main [64]. Fastly, Inc ist auf Echtzeit-Content Delivery Network (CDN) spezialisiert [65]. Es gab etwas vermehrt Kontakt mit ads.viralize.tv (35.186.236.232) die laut IP.info zu Google LLC gehört und sich in Kalifornien in den USA befindet [66]. Wenn man jedoch ads.viralize.tv im Internet sucht, findet man schnell heraus das Virazlize ein unternehemn ist, das darauf spezialisiert ist, bezahlte Werbung zu schalten [67].

4.9 Analyse des Arbeitsspeichers [Vincent]

Die Analyse des Arbeitsspeichers erfolgte, wie in 3.4.5 beschrieben wurde. Die .elf-Dateien, welche mit dem in Listing 1 dargestellten Befehl erstellt wurden, wurden mittels dem Befehl, wie er in Listing 2 zu sehen ist, auf alle Zeichenkette einer Mindestlänge von Zehn in lesbare „.txt-Dateien“ überführt.
Daraufhin wurden diese .txt-Dateien auf Besonderheiten untersucht. Dies umfasst beispielsweise Namen von Cookies, URLs oder Eingaben im Zusammenhang mit der Interaktion mit der Webseite.
Allerdings erwies sich diese Vorgehensweise als wenig erfolgsversprechend. Zwar gab es Treffer in Hinblick auf die gesuchten Elemente, allerdings fehlte hier der Kontext dafür, inwiefern diese Daten in welchem Zusammenhang im Arbeitsspeicher lagen. Abbildung 18 zeigt ein Beispiel für ein „Cookie Pair“, welches in der Verbindung mit der Webseite aus 4.8.1 übermittelt wurde. Jedoch zeigt sich hier, dass so jegliches Verständnis dafür fehlt, wie diese Datei nun im Arbeitsspeicher vorliegt.
In Hinblick auf diese Problematik, so wie die riesige Menge an Daten, welche bei der Analyse des Arbeitsspeichers vorliegt, konnten hier keine nützlichen Informationen gewonnen werden.

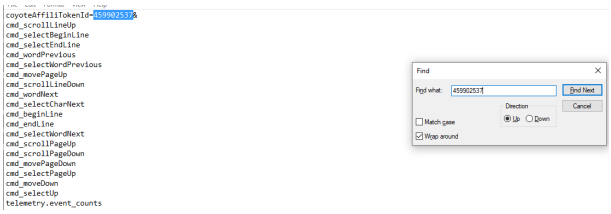


Abbildung 18: Ein Cookie pair, welches in Aufzeichnungen über den Arbeitsspeicher gefunden wurde.

```
strings -n 10 example_dump.elf>"example_dump.txt"
```

Listing 2: Der Befehl, welcher eine .elf-Datei in eine .txt-Datei überführt.

4.10 Analyse des Massenspeichers von Firefox, Thunderbird und Evolution [Vincent]

Der Massenspeicher in Hinblick auf Firefox, Thunderbird und Evolution wurde entsprechend der Methodik untersucht. Hier wurden ausschließlich „sqlite“- und „.db“-Dateien untersucht, welche zu den entsprechenden Anwendungen existieren. Im Folgenden wird beschrieben, was die drei Anwendungen jeweils an interessanten Informationen speichern.

4.10.1 Firefox [Bernhard]

Firefox speichert Anwendungsdaten nach verschiedenen Browser-Profilen sortiert ab. Das benutzte Profil kann bei jedem Nutzer einen anderen Namen haben, dementsprechend kann kein absoluter Pfad zum Speicherort angegeben werden: `$HOME/.mozilla/firefox/<profile>.$USER` [68]
Innerhalb des Verzeichnisses befinden sich neben einigen Konfigurationsdateien mehrere SQL-Datenbanken (Abbildung B.4a), die Informationen zu besuchten Webseiten (History) und Cookies enthalten.

Bei unserer Analyse wurden folgende Daten berücksichtigt:

1) **Cookie-Daten** („cookies.sqlite“): In der Tabelle der Cookies können wichtige Informationen wie z.B. der Name, der Wert, der betreffende Host, sowie Zeitstempel vom Erstellen, letzten Zugriff und Ablaufzeitpunkt entnommen werden (Abbildung B.4b).
2) **Formulardaten** („formhistory.sqlite“): Die in Formularfelder auf der Phishing-Seite eingegebenen (synthetischen) Daten werden vom Browser in dieser Tabelle gespeichert. Diese Daten eröffnen die Möglichkeit des genauen Nachvollziehens, welche Daten wann an den Phisher gesendet wurden (Abbildung B.4c).
3) **Besuchte Seiten** („places.sqlite“): Die Daten über die besuchten Webseiten liefern Auskunft darüber, wann welche Webseite wie oft geladen wurde (Abbildung B.4d).

4.10.2 Thunderbird [Bernhard]

Thunderbird speichert seine Daten ähnlich wie Firefox in mehreren Profilen, allerdings in einem anderen Verzeichnis: `$HOME/.thunderbird/<profile>.default-default` [69]

Folgende Daten haben wir während unserer Untersuchung betrachtet:

1) **Abgerufene Webseiten** („places.sqlite“): In dieser Tabelle werden die durch eventuell enthaltene Links nachgeladene Seiten gespeichert. Wie in Abbildung B.5 zu sehen ist, wurde bei dem Beispiel aus der untersuchten E-Mail aus Abschnitt 4.8.2 ein HTML-Dokument von Google nachgeladen.
2) **Generelle E-Mail-Daten** („global-messages-db.sqlite“): Diese Datenbank umfasst alle Nachrichten im E-Mail-Postfach, die lokal gespeichert wurden. Dabei handelt es sich um Daten wie z.B. Kontakte, Betreffe der Mails, vorkommende MIME-Types [70] sowie Inhalte der Nachrichten. Da alle diese Daten auch im Mail-Client ersichtlich sind, hat die Analyse dieser Datenbank einen eher geringen Wert für unsere Untersuchung.
3) **Cookies** („cookies.sqlite“): Die meisten von uns untersuchten E-Mails setzen keine Cookies. Dies ist daran zu erkennen, dass durch keine E-Mail die Datenbank modifiziert wurde. Bei Einzelfällen hat sich die Cookie-Datenbank während der Untersuchung zwar geändert, allerdings konnten keine Cookies in der Datenbank während der Analyse identifiziert werden, da sie leer war.

4.10.3 Evolution [Vincent]

Evolution speichert Informationen über E-Mails auf zwei verschiedene Weisen. In der Datenbank „folders.db“, welche unter `$HOME/.local/share/evolution` [44] zu finden ist, werden Informationen über das Postfach an sich gespeichert, wie in Abbildung B.6a zu sehen ist. Dieser Datenbank sind u.a. der Sender, der Betreff der E-Mail und andere Metainformationen zu den Nachrichten, welche in den verschiedenen Ordnern des Postfachs vorliegen, zu entnehmen.
Ebenso speichert Evolution Header- und Body von E-Mails. Diese lassen sich unter `$HOME/.cache/evolution` [44] finden. Wie in Abbildung B.6b zu sehen ist, existieren in entsprechenden Ordnern, welcher der Struktur des Postfachs des Nutzers entsprechen, einzelne Dateien, welche Header und Body der entsprechenden E-Mails enthalten, wie Abbildung B.6c zu entnehmen ist.
Von den drei untersuchten Anwendungen speichert Evolution am wenigsten Informationen. Ebenso sind die Informationen, die gespeichert werden, relativ uninteressant, da hier höchstens Informationen aus den Header-Dateien, welche gespeichert werden, gewonnen werden könnten. Jedoch können diese nachweislich verfälscht werden [7], wodurch sie als Betrachtungsgegenstand unattraktiver werden. Eine Untersuchung dieser Header ist dennoch im Abschnitt 3.4.1 nachzuverfolgen.

4.11 Vergleich [Gina]

Insgesamt gab es bei Phishing-Mails deutlich mehr Auffälligkeiten, als bei dem Newsletter, den wir untersucht haben. Es gab auf den Phishingseiten deutlich mehr IP-Adressen mit den kommuniziert wurde, viele davon Tracker. Es gab auch beispielsweise Verbindungen zu dritten Parteien aus dem Ausland. Eine war angeblich die deutsche Rentenversicherung, die IP befand sich jedoch in Spanien.

Darüber hinaus gab es auf den Phishingseiten Aufforderungen etwas einzugeben (Bankdaten, Name etc). Das war beim Newsletter nicht der Fall, denn da war es nur möglich zwischen den Beiträgen zu wählen oder ggf. auf eine Dritte Seite weitergeleitet zu werden, was jedoch schon vor dem Klick erkennbar war. Die Phishingseiten hingegen verliefen sich nach einiger Zeit. Sobald man seine Daten angegeben hat um zum Beispiel einen Gutschein zu gewinnen, dann ging es auf der Seite nicht weiter.

5 Evaluierung

5.1 Vergleich Mitmproxy und Wireshark [Abdalla]

Die Untersuchung von Phishing E-Mails wurde durch Wireshark bzw. mitmproxy durchgeführt. Beide Tools haben zahlreiche Eigenschaften bzw. Vorteile. Aber bei mitmproxy-Tool ist wie im Abschnitt 4.6.2 erläutert, können durch dieses Tool nicht alle Phishing Webseiten geöffnet sein. Deswegen haben wir die Untesuchungen ausführlich durch Wireshark durchgeführt , da es effektiver und einfach zu benutzen ist.

5.2 Phishing-Quellen Testset [Michelle]

Im Rahmen der Untersuchung haben wir sechs Phishing E-Mails aufgezeichnet und analysiert, wie in Abschnitt 4.2 in Tabelle 2 beschrieben ist. Diese lassen sich in die Kategorien URL-Obfuscation, HTML-Attachment und Open-Redirect einteilen.
Dabei ist anzumerken, dass wir nicht nur sechs Phishing E-Mails erhalten haben. Allerdings waren einige Links zu Phishing Webseiten in den E-Mails nur für einen so kurzen Zeitraum funktionsfähig, dass es nicht mehr möglich war Aufzeichnungen von diesen zu machen.
Da uns nur eine so geringe Menge funktionierender Phishing E-Mails zur Verfügung stand, war es uns nicht möglich aussagekräftige Testsets zusammenzustellen.
Hätten wir mehr Phishing E-Mails bekommen, hätten wir sie nach den drei Phishing E-Mail Arten in Testsets eingeteilt. Es wäre außerdem sinnvoll gewesen zwei Vergleich-Testsets zu erstellen. Eins, was Spam E-Mails enthält und eins, welches normale E-Mails enthält.

5.2.1 Hauptproblem

Das Hauptproblem waren die fehlenden Phishing E-Mails, die wir für die Untersuchung brauchen. Es war erheblich schwerer gezielt Opfer von Phishing Angriffen zu werden, als gedacht. Mit den Strategien, die wir benutzt haben sind wir eher Ziel von Spam als von Phishing E-Mails geworden. Um keinen persönlichen Schaden zu nehmen wurde eine neue E-Mail Adresse erstellt, die nur dazu da ist Ziel von Phishing E-Mails zu werden.
Vorab haben wir uns online informiert, welche E-Mail Anbieter als am „schlechtesten“ gelten, was Spamfilter und Spamabwehr (und in dem Zuge auch Phishing Abwehr) betrifft. Der Gedanke dahinter war, dass wir so schneller mehr Phishing E-Mails bekommen können. Leider gab es bei einigen dieser Anbieter schon bei der Erstellung einer neuen E-Mail-Adresse das Problem, dass sie eine Telefonnummer o.Ä. zur Authentifizierung verlangen. Diese Anbieter fielen also raus, da wir dazu keine Testdaten benutzen konnten und es einfach nicht im Rahmen unserer Möglichkeiten war, uns eine Telefonnummer nur für dieses Problem zu besorgen. Wir konnten aber auch nicht riskieren eigene Telefonnummern zu benutzen, denn im schlimmsten Fall hätten diese bei der Aufzeichnung der Phishing-Mails den Betrügern in die Hände fallen können.
Anfangs hat jeder von uns eine eigene E-Mail Adresse erstellt, um Phishing E-Mails zu bekommen. Später haben wir uns auf eine E-Mail Adresse beschränkt und unsere Strategien, möglichst Ziel von Betrügern zu werden, darauf konzentriert.
Da die E-Mail Adressen alle neu erstellt wurden, war das erste Ziel diese im Internet zu verbreiten, weil es so wahrscheinlicher ist, Ziel eines Angriffs zu werden. Das geschah v.a. durch die Anmeldung bei Newslettern, dem Posten der Adresse in Foren und dem Anmelden bei verschiedenen Dienstleistern, wie z.B. Amazon. Wir haben außerdem das Mugu Guestbook [71] benutzt. Dies ist eine Internetseite, auf der man seine E-Mail Adresse angeben kann und verschiedene E-Mails von Betrügern zugeschickt bekommt, bei denen es um Betrugsversuche mit dem Thema Finanzen geht.
Leider waren diese E-Mails immer Spam E-Mails und keine Phishing E-Mails. Zur Untersuchung haben sie sich nicht geeignet, da es meist E-Mails waren, die keine eingebetteten Links oder HTML-Anhänge enthielten. Man sollte z.B. mit seinen persönlichen Daten, inklusive Kontodaten, auf die empfangene E-Mail antworten.
Eine große Hilfe waren Freunde, Bekannte und Familienmitglieder, die Phishing E-Mails, welche an ihre privaten Adressen gesendet wurden, an uns weitergeleitet haben. Der Großteil unserer untersuchten E-Mails wurde uns von Dritten zugeschickt.
Am Ende konnten wir trotzdem keinen großen Pool an Phishing E-Mails ansammeln. Uns war es zwar möglich mindestens eine Phishing E-Mail jeder Art (URL-Obfuscation, HTML-Attachment, Open-Redirect) zu untersuchen, allerdings spiegelt das bei weitem nicht das volle Spektrum aller Phishing-Angriffe wider.

5.3 Was ein großer Dienstanbieter von Phishing-Angriffen wahrnehmen kann [Felix]

Diese Frage muss für jede Art von Diensteanbieter selbst geklärt werden, da nicht jede alle Daten einsehen kann. Zusätzlich gibt es auch innerhalb verschiedener Diensteanbieter einer Klasse (z.B. Mail Provider) Unterschiede.

5.3.1 Mail-Dienstanbieter

Mail Anbieter stellen die erste Verteidigungslinie gegen Phishing-Attacken via Mail dar. Dabei stehen ihnen zahlreiche Hilfsmittel zur verfügung um diese zu erkennen und entfernen.

5.3.1.1 Header

Alle größeren Mailanbieter untersuchen Mailheader auf verdächtige Inhalte und entfernen diese Mail falls festgestellt wird, dass es sich um eine gefährliche Mail handelt. Dadurch erhielten wir nur E-Mails die mindestens einen Filter überwunden haben. Unserer Untersuchungen haben uns

zu dem Schluss geführt, dass eine weitere Herausfilterung der Phishing E-Mails nahezu unmöglich ist. Denn wären diese leicht als schädlich identifizierbar, wären sie herausgefiltert worden und wir hätten die E-Mail nicht bekommen. Wenn es rein um den Header geht, lassen sich legitime und Phishing Mail kaum voneinander unterscheiden.

5.3.1.2 Mail-Body

Auch hier findet bereits bei den meisten Mailanbietern eine Filterung statt. Dabei geht es zu einen Teil um die Filterung nach Stichwörtern, um gezielter Werbung schalten zu können und zum anderen, um die Erkennung gefährlicher Mail [72]. Dabei wird der Wortschatz und die Grammatik überprüft und gegebenenfalls als schädliche Mail identifiziert. Weitere Erkennungsmerkmale existieren jedoch nicht, wenn es sich um E-Mails ohne eingebettete HTML-Seite oder Bilder handelt. Falls dies doch der Fall sein sollte, können sich neben Links auch Tracker in diesen enthalten.

5.3.1.3 Anhang

Genauso wie der Mail Body wird auch der Anhang auf gefährliche Dateien überprüft. Dies ist hier insbesondere notwendig, da sich im Anhang Malware befinden kann. Die Betrachtung des Anhangs stellt jedoch nur einen kleinen Teil unserer Untersuchungen dar, da Malware nicht Teil dieser Arbeit sind. Dennoch haben wir E-Mails bekommen in deren Anhang sich eine Weiterleitung auf eine Phishing Webseite befand. Diese zu überprüfen fällt wiederum in das Feld der Linkverfolgung.

5.3.1.4 Linkverfolgung

Ob eine Linkverfolgung stattfindet und in welchem Umfang ist für jeden Mailanbieter individuell zu beantworten. Jedoch kann man sagen, dass eine genaue Analyse aufgrund der extrem hohen Anzahl an Mail die täglich versendet werden nahezu unmöglich ist (300 Mrd Mail pro Tag). Bei einer Verfolgung dieser Links, könnte der Mailprovider dasselbe wahrnehmen wie der Internetanbieter (beschrieben in 5.3.2).

5.3.1.5 Zusammenfassung

Insgesamt lässt sich also für Mailanbieter festhalten, dass sie bereits einen guten Schutz vor Phishing-Mails bieten und ohne eine Linkverfolgung eine weitere Herausfilterung aus lokalen Daten nicht möglich ist. Ob die Linkverfolgung eine sinnvolle Investition von seitens der Mailprovider ist, lässt sich von unserem Standpunkt aus nicht feststellen.

5.3.2 Internet-Dienstleister

Im Gegensatz zum Mail-Dienstanbieter stehen den Internetanbietern mehr Daten, um bösartige Webseiten zu erkennen, zur Verfügung. Hierbei muss jedoch unterschieden werden in den Internetanbieter des Opfers und dem auf welcher die Phishing-Website sich befindet. Wichtig ist auch zu beachten, ob die Website verschlüsselt ist oder nicht.

5.3.2.1 Internetanbieter des Opfers

Egal ob die Phishing Website verschlüsselt ist oder nicht muss diese zuerst geladen werden. Dabei versuchen viele Kriminelle zu verbergen, dass es sich um eine gefälschte Website handelt. Dies geschieht beispielsweise über Redirects. Solche Methoden, die die eigentliche Webadresse nicht direkt preisgeben, können definitiv vom Internetanbieter und auch dem Browser festgestellt werden (z.B. 4.8.1).

Für verschlüsselte Phishing Webseiten sind es wenige Daten die durch den Internetanbieter erfasst werden können. So können nur die Domain der Website, das verwendete Gerät/ Browser und der ungefähre Standort ermittelt werden. Das heißt von Seiten des Opfers aus kommen keine Daten die zur Identifizierung von Phishing-Angriffen sinnvoll wären.

Für den Fall dass die Website unverschlüsselt ist, können wesentlich mehr Daten gefunden werden. So kann der Internetanbieter die genaue Adresse und Zeiten der Verbindung herauslesen, welche Eingaben auf der Website vollzogen und welche Dateien hoch bzw. heruntergeladen werden. Hierbei kann ein ziemlich genaues Bild über die besuchte Website geschaffen werden und die Identifizierung von Phishing-Webseiten ist möglich. Ähnlich zu den Daten die den Link betreffen kann wieder nicht mit absoluter Gewissheit ausgesagt werden, ob eine Website für betrügerische Aktionen verwendet wird oder nicht.

5.3.2.2 Host der Phishing-Website

Für Diensteanbieter die Webseiten hosten, gibt es keine Regeln was sie abspeichern müssen [73]. D.h. welche Daten mitgeschnitten werden hängt vom Host ab. Im allgemeinen Fall werden jedoch sehr viele Daten erhoben und abgespeichert. Diese Daten umfassen zum Beispiel den Namen des Betreibers, Adresse, Kreditkarteninformationen usw. Von unseren Ergebnissen kann man allerdings nicht implizieren, was dieser Anbieter mitschneidet. Wenn er jedoch den Verkehr auf der Seite festhält, kann der Host alle Verbindungen und Informationen zu diesen einsehen. Im Falle, dass dem Serveranbieter alle Daten und sämtliche Zahlungsvorgänge vorliegen sollten, ist eine Identifikation der Phishing-Websiten durchaus möglich. Allerdings steht hier der Datenschutz in Frage.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung der Untersuchungsergebnisse [Gina]

In unserer Arbeit haben wir feststellen müssen wie schwierig die Erkennung eines Phishing-Angriffs anhand fester Daten ist. Wie bereits erwähnt ist die Filterung nach gefährlicher Mail, die von großen Dienstanbietern bisher durchgeführt wird, sehr erfolgreich. Ohne Analyse der eingebetteten Links können wir momentan eine weitere Erkennung von Phishing-Mail, die nicht durch Nutzer oder menschliche Überwachung geschieht ausschließen. Dies lässt als einzige Möglichkeit die Untersuchung der Phishing Webseiten. Das heißt für Mailprovider eine Verfolgung der Links, welche in den E-Mails eingebunden sind. Ob diese allerdings eine praktisch umsetzbare Strategie ist, können wir jedoch nicht mit Gewissheit sagen. Für Internetanbieter verhält es sich ähnlich, wie bei den Mailanbietern. Das bedeutet eine Untersuchung des Netzwerkverkehrs und den aufgebauten Verbindungen erlaubt es unter Umständen einen Phishing-Angriff zu erkennen. Dennoch ist wieder eine faktenbasierte Aussage über die Authentizität einer Website schwierig.
Abschließend lässt sich feststellen, dass trotz zahlreicher Bemühungen Phishing-Attacken zu unterbinden, immer noch die Wachsamkeit der Nutzer einen wichtigen Teil des Selbstschutzes darstellt. Dies wird sich ohne Hinzufügen von zusätzlichen Protokollen und genaueren Methoden der Analyse auch nicht ändern.

6.2 Ausblick für zukünftige Arbeiten [Abdalla]

Es gibt viele Verbesserungen, deren Umsetzung den Rahmen dieser Arbeit sprengen würde und so bleibt noch Raum für weitere Arbeiten. In diesem Abschnitt werden empfohlene Verbesserungsvorschläge und ihre Auswirkungen auf Phishing-Angriffe beschrieben.

6.2.1 Was könnten Dienstanbieter tun, um Phishing-Angriffe zu verhindern, wenn sie davon wissen?

Für die Dienstanbieter und die Organisationen ist es wichtig, Sicherheitsmaßnahmen durchzuführen, um Phishing-Angriffe zu verhindern. Die folgenden Maßnahmen könnten beispielsweise eingeführt werden:

- Die E-Mails von ihren Domänen schwer zu fälschen, indem sie Anti-Angriff-Kontrollen wie z.B. DMARC [74] und DKIM [74] einsetzen.
- Alle eingehende Phishing E-Mails filtern und blockieren, bevor sie ihre Benutzer erreicht.
- Direktes Erkennen von ungewöhnlichen Anfragen wie z.B. Zahlungen, Anmeldungen, die getätigt werden.
- Eine Umgebung für die Benutzer schaffen, damit sie Phishing-Mails direkt melden können.

Alle Dienstanbieter werden irgendwann mit einem Sicherheitsvorfall konfrontiert, daher müssen sie in der Lage sein, diese so schnell wie möglich zu erkennen und darauf zu reagieren, und sicherstellen, dass die Benutzer im Voraus wissen, wie sie einen Vorfall/Phishing-Angriff melden können.

6.2.2 Mehr Fokus auf nützliche Tools in diesem Zusammenhang für die Anaylse von Pishing-Mails

Mit Volatility [75] lässt sich die Untersuchung von Ram (Speicherforensik) besser analysieren. Volatility ist ein Open-Source Tool und hilft den Benutzern bei der Anaylse von Ram, ausführliche Informationen zu bekommen. Einige Eigenschaften von Volatility, die der zukünftigen Arbeit hilft sind:

- Ausführliche Analyse des Laufzeitzustands eines Systems anhand der gefundenen Daten in RAM.
- Sensible Daten/Informationen, E-Mail Header, Passwörter und verschlüsselten Daten in unverschlüsselter Form können erfasst werden.
- Mit bestimmten Plugins können alle Dateien-Formaten identifiziert werden.

Mit Volatility wird dann die Veränderung an Daten im System besser erkannt. Außerdem kann man zahlreiche Informationen erkennen was im Speicher passiert, wenn ein Link von einer Phishing-Mail geöffnet wird. Beziehungsweise ob sich Malware oder Cookies im Speicher befinden.

6.2.3 Automatisierte Erkennung von Phishing E-Mails [Michelle]

Ein möglicher Ausblick für die Zukunft, zum automatisierten Erkennen von Phishing E-Mails, wäre der Aufbau eines eigenen Servers, der synthetische Phishing E-Mails verschickt. Die Idee dahinter ist, dass man kleine Veränderungen an den zugeschickten Phishing E-Mails vornimmt. Die damit neu erstellten E-Mails werden an die eigene E-Mail Adresse verschickt, um zu sehen, ob sie immer noch vom Spamfilter gefiltert werden. Damit könnte herausgefunden werden, welche Kriterien wichtig für eine automatisierte Erkennung von Phishing E-Mails wären. Zum einen müsste man dazu bereits ein Testset mit mehreren Phishing Mails jeder Art haben, um mehr von diesen erstellen zu können. Zum anderen bräuchte auch die Erstellung der „falschen“ Phishing E-Mails eine bestimmte Zeit.

Literatur

[1] "Phishing e-mails – passwortdiebstahl durch phishing"," Jun. 2022. [Online]. Available: https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Cyber-Sicherheitslage/Methoden-der-Cyber-Kriminalitaet/Spam-Phishing-Co/Passwortdiebstahl-durch-Phishing/passwortdiebstahl-durch-phishing_node.html

[2] "Was sie bei einem phishing-angriff riskieren," Jul. 2022. [Online]. Available: https://de.norton.com/norton-blog/2016/10/was_sie_bei_einemph.html

[3] "how gmail helps users avoid email scams," Jul. 2022. [Online]. Available: https://cloud.google.com/blog/products/workspace/how-gmail-helps-users-avoid-email-scams

[4] "Vorsicht, phishing! das wichtigste in fünf punkten," Jul. 2022. [Online]. Available: https://www.triumph-adler.com/ta-de-de/talking-future/it-sicherheit/phishing

[5] "83 prozent aller unternehmen opfer von phishing-attacken," Jul. 2022. [Online]. Available: https://itwelt.at/news/83-prozent-aller-unternehmen-opfer-von-phishing-attacken/

[6] "Was ist phishing und wie funktioniert das prinzip?" Jul. 2022. [Online]. Available: http://www.was-ist-malware.de/virenschutz/phishing/

[7] K. Morovati, "Detection of phishing emails with email forensic analysis and machine learning techniques," *International Journal of Cyber-Security and Digital Forensics*, vol. 8, pp. 98–107, 01 2019.

[8] "Alle wichtigen informationen aus rechtlicher sicht zum phishing!" Jun. 2022. [Online]. Available: https://www.recht-freundlich.de/phishing-online-banking/alles-wichtige-phishing

[9] "Bsi - leitfaden it-forensik," pp. 81–82, Jun. 2022. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Leitfaden_IT-Forensik.pdf?__blob=publicationFile&v=1

[10] "How to get scam emails - quora," Jun. 2022. [Online]. Available: https://www.quora.com/How-can-I-get-scam-emails

[11] "Ein tool für getting phishingsmails," Jun. 2022. [Online]. Available: https://temp-mail.org/de/

[12] "Wireshark · go deep." Jun. 2022. [Online]. Available: https://www.wireshark.org/

[13] "Verwendung von wireshark: Umfassendes tutorial + tipps," Jun. 2022. [Online]. Available: https://www.varonis.com/de/blog/verwendung-von-wireshark

[14] "mitmproxy - an interactive https proxy," Jun. 2022. [Online]. Available: https://mitmproxy.org/

[15] "How to man in the middle https using mitmproxy," Jul. 2022. [Online]. Available: https://earthly.dev/blog/mitmproxy/

[16] "Thunderbird — e-mails leicht gemacht. — thunderbird," Jun. 2022. [Online]. Available: https://www.thunderbird.net/de/

[17] "Häufig gestellte fragen zu thunderbird," Jul. 2022. [Online]. Available: https://support.mozilla.org/de/kb/haeufig-gestellte-fragen-zu-thunderbird

[18] "Gnome/evolution: Read-only mirror of https://gitlab.gnome.org/gnome/evolution," Jun. 2022. [Online]. Available: https://github.com/GNOME/evolution

[19] "14 evolution: E-mails und kalender," Jul. 2022. [Online]. Available: https://documentation.suse.com/de-de/sles/15-GA/html/SLES-all/cha-gnome-evolution.html

[20] "Oracle vm virtualbox," Jun. 2022. [Online]. Available: https://www.virtualbox.org/

[21] "Versionszyklus von firefox esr (extended support release)," Jun. 2022. [Online]. Available: https://support.mozilla.org/de/kb/versionszyklus-von-firefox-esr

[22] "Decrypting ssl/tls traffic with wireshark [updated 2021] - infosec resources," Jun. 2022. [Online]. Available: https://resources.infosecinstitute.com/topic/decrypting-ssl-tls-traffic-with-wireshark/

[23] "How mitmproxy works," Jun. 2022. [Online]. Available: https://docs.mitmproxy.org/stable/concepts-howmitmproxyworks/

[24] "Sqlite home page," Jun. 2022. [Online]. Available: https://www.sqlite.org/index.html

[25] "Getting started," Jun. 2022. [Online]. Available: https://docs.mitmproxy.org/stable/overview-getting-started/

[26] "Imap-synchronisierung," Jun. 2022. [Online]. Available: https://support.mozilla.org/de/kb/imap-synchronisierung

[27] "Study: Most phishing pages are abandoned or disappear in a matter of days," Jun. 2022. [Online]. Available: https://www.techrepublic.com/article/study-most-phishing-pages-are-abandoned-or-disappear-in-a-matter-of-days/

[28] "What is an email header?" Jul. 2022. [Online]. Available: https://www.gmass.co/blog/email-header/#whatis

[29] "Message header analyzer insert the message header you would like to analyze," Jul. 2022. [Online]. Available: https://mha.azurewebsites.net

[30] "Ipinfo," Jul. 2022. [Online]. Available: https://ipinfo.io/

[31] "Http request methods," Jul. 2022. [Online]. Available: https://www.w3schools.com/tags/ref_httpmethods.asp

[32] "Handshake protocol," Jul. 2022. [Online]. Available: https://www.ip-insider.de/was-ist-ein-handshake-a-794054/

[33] "Following protocol streams," Jul. 2022. [Online]. Available: https://www.wireshark.org/docs/wsug_html_chunked/ChAdvFollowStreamSection.html

[34] "How to use strings command in ubuntu," Jul. 2022. [Online]. Available: https://linuxhint.com/strings-ubuntu-command-use/

[35] "What is the grep command?" Jul. 2022. [Online]. Available: https://phoenixnap.com/kb/grep-command-linux-unix-examples#:~:text=Grep%20is%20a%20Linux%20%2F%20Unix,searching%20through%20large%20log%20files.

[36] "Mitm-angriff," Jun. 2022. [Online]. Available: https://www.forensicfocus.com/articles/man-in-the-middle-attack-forensics/

[37] "Vorteile von mitmproxy," Jun. 2022. [Online]. Available: https://docs.mitmproxy.org/stable/#features

[38] "Strings2 github," May 2022. [Online]. Available: https://github.com/glmcdona/strings2

[39] S. Kiltz, "Data-centric examination approach (dcea) for a qualitative determination of error, loss and uncertainty in digital and digitised forensics," Ph.D. dissertation, Otto-von-Guericke-Universitaet Magdeburg, Fakultaet fuer Informatik, 2020. [Online]. Available: http://dx.doi.org/10.25673/34647

[40] "Email trackers are watching your inbox. here's how to block them." Jun. 2022. [Online]. Available: https://www.huffpost.com/entry/email-trackers-how-to-block-see-why_l_5c95335fe4b01ebeef0f0e05

[41] "Cwe-601: Url redirection to untrusted site ('open redirect')," Jun. 2022. [Online]. Available: https://cwe.mitre.org/data/definitions/601.html

[42] "Url obfuscation: Kein link ohne risiko," Jun. 2022. [Online]. Available: https://www.it-administrator.de/themen/sicherheit/fachartikel/123090.html

[43] "Top 9 email services without phone verification [2022]," Jun. 2022. [Online]. Available: https://www.emailsverified.com/free-email-without-verification/

[44] "Data storage locations," Jun. 2022. [Online]. Available: https://help.gnome.org/users/evolution/stable/data-storage.html.en

[45] "Debian 11 aktualisiert: 11.3 veröffentlicht," Jun. 2022. [Online]. Available: https://www.debian.org/News/2022/20220326

[46] "Antispam stamps," Jul. 2022. [Online]. Available: https://docs.microsoft.com/en-us/exchange/antispam-and-antimalware/antispam-protection/antispam-stamps?view=exchserver-2019

[47] "82.81.102.128/25," Jun. 2022. [Online]. Available: https://ipinfo.io/AS8551/82.81.96.0/20-82.81.102.128/25

[48] "172.245.1.37 server-mania, ca," Jun. 2022. [Online]. Available: https://urlscan.io/ip/172.245.1.37

[49] "As197518 rackmarkt sl," Jun. 2022. [Online]. Available: https://ipinfo.io/AS197518/188.95.248.0/21-188.95.252.0/25

[50] "142.250.186.170," Jun. 2022. [Online]. Available: https://ipinfo.io/142.250.186.170

[51] "35.158.79.89," Jun. 2022. [Online]. Available: https://ipinfo.io/35.158.79.89

[52] "82.98.110.193," Jun. 2022. [Online]. Available: https://ipinfo.io/82.98.110.193

[53] "Supratec membrane gmbh," Jun. 2022. [Online]. Available: https://supratec-membrane.de/en/

[54] "Html attachments in phishing e-mails," Jul. 2022. [Online]. Available: https://securelist.com/html-attachments-in-phishing-e-mails/106481/

[55] "Ipinfo," Jul. 2022. [Online]. Available: https://ipinfo.io/51.210.3.236

[56] "Ovh," Jul. 2022. [Online]. Available: https://de.wikipedia.org/wiki/OVH

[57] "Ipinfo," Jul. 2022. [Online]. Available: https://ipinfo.io/192.185.195.169

[58] "Is websitewelcome.com safe?" Jul. 2022. [Online]. Available: https://www.mywot.com/en/scorecard/websitewelcome.com
[59] "Google trust services," Jul. 2022. [Online]. Available: pki.goog
[60] "94.103.89.43," Jun. 2022. [Online]. Available: https://ipinfo.io/94.103.89.43
[61] "18.194.140.191," Jun. 2022. [Online]. Available: https://ipinfo.io/18.194.140.191
[62] "Ip address details," Jul. 2022. [Online]. Available: https://ipinfo.io/23.213.161.15
[63] "Was ist akamai? einfach erklärt," Jul. 2022. [Online]. Available: https://praxistipps.chip.de/was-ist-akamai-einfach-erklaert_40879
[64] "Ip address details," Jul. 2022. [Online]. Available: https://ipinfo.io/151.101.114.209
[65] "Fastly, inc. (a2ph9t)," Jul. 2022. [Online]. Available: https://de.marketscreener.com/kurs/aktie/FASTLY-INC-58136331/unternehmen/
[66] "Ip address details," Jul. 2022. [Online]. Available: https://ipinfo.io/35.186.236.232
[67] "Earn more money with your website," Jul. 2022. [Online]. Available: https://viralize.com/
[68] "What folder in ubuntu holds my history, bookmarks, and extensions?" Jun. 2022. [Online]. Available: https://support.mozilla.org/en-US/questions/1267236
[69] "Profiles - where thunderbird stores your messages and other user data," Jun. 2022. [Online]. Available: https://support.mozilla.org/en-US/kb/profiles-where-thunderbird-stores-user-data
[70] "Common mime types - http — mdn," Jun. 2022. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types
[71] "Mugu guestbook," Jul. 2022. [Online]. Available: https://www.muguguestbook.com/
[72] "Which email providers are scanning your emails?" Jul. 2022. [Online]. Available: https://www.maketecheasier.com/which-email-providers-scanning-emails/
[73] A. B. Todd Shipley, *Investigating Internet Crimes*. Syngress, 2013.
[74] "Domain-based message authentication, reporting, and conformance (dmarc)," Jul. 2022. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7489
[75] "Ramanalyse," Jul. 2022. [Online]. Available: https://www.volatilityfoundation.org/

Anhang A

A.1 Konzept und Versuchsaufbau: Methodikdiagramm der Datenakquise (zu 3.3)

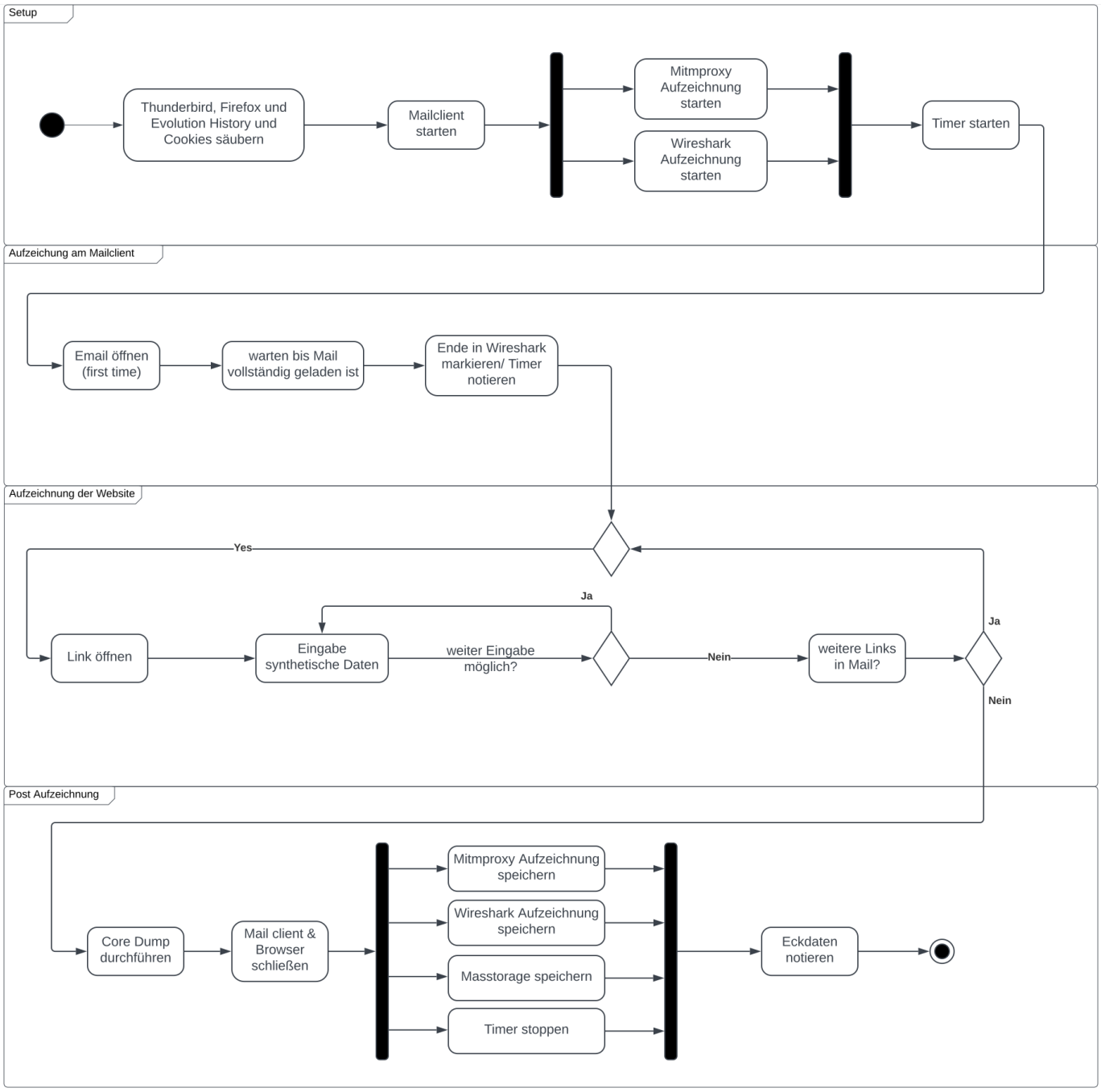


Abbildung A.1: Methodikdiagramm für die Datenakquise.

A.2 Konzept und Versuchsaufbau: Methodikdiagramm der Analyse mit Wireshark (zu 3.4.2)

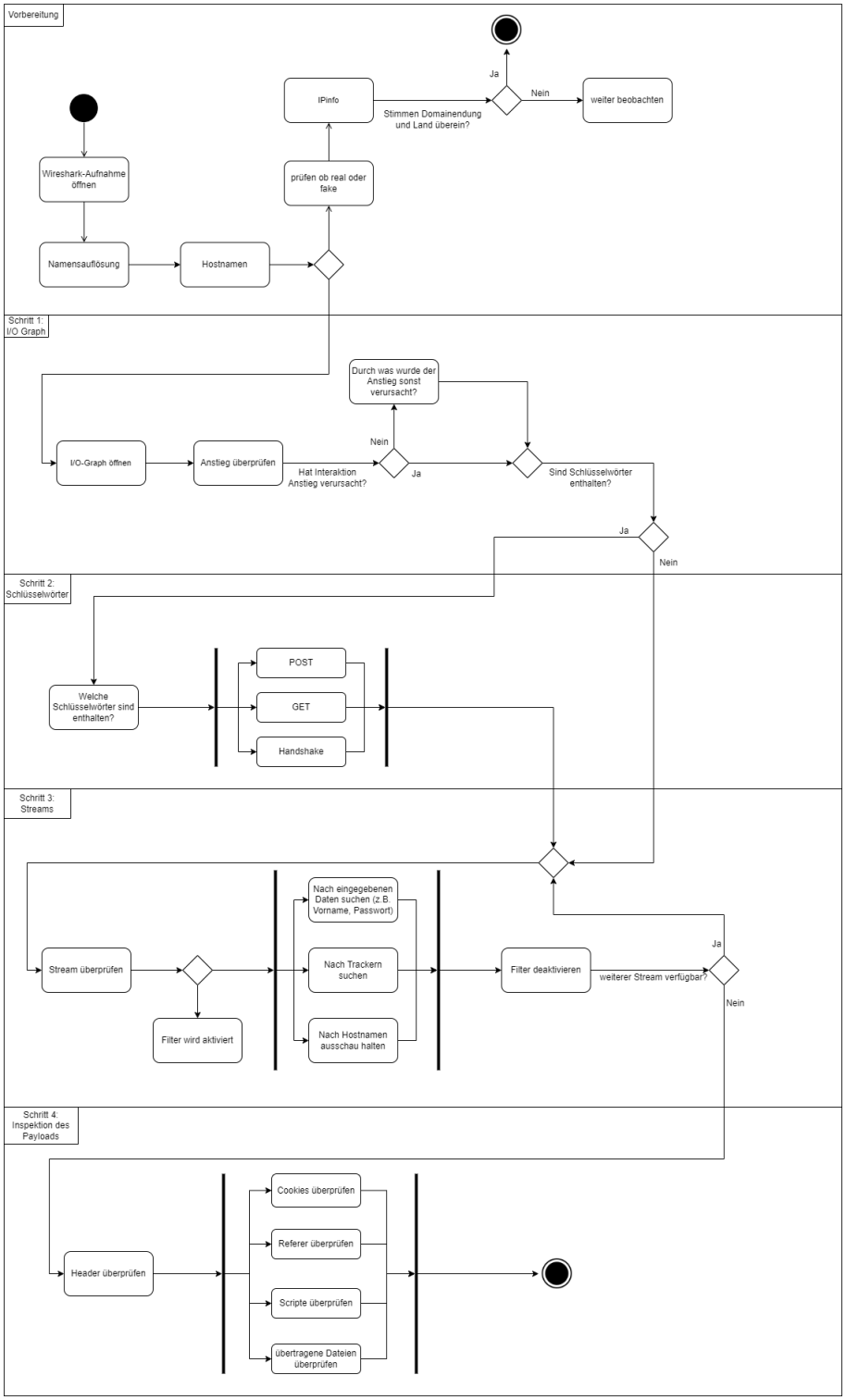


Abbildung A.2: Methodikdiagramm für die Analyse mit Wireshark.

Anhang B

B.1 Implementierung: Vergleich mitmproxy/Wireshark (zu 4.6.1)

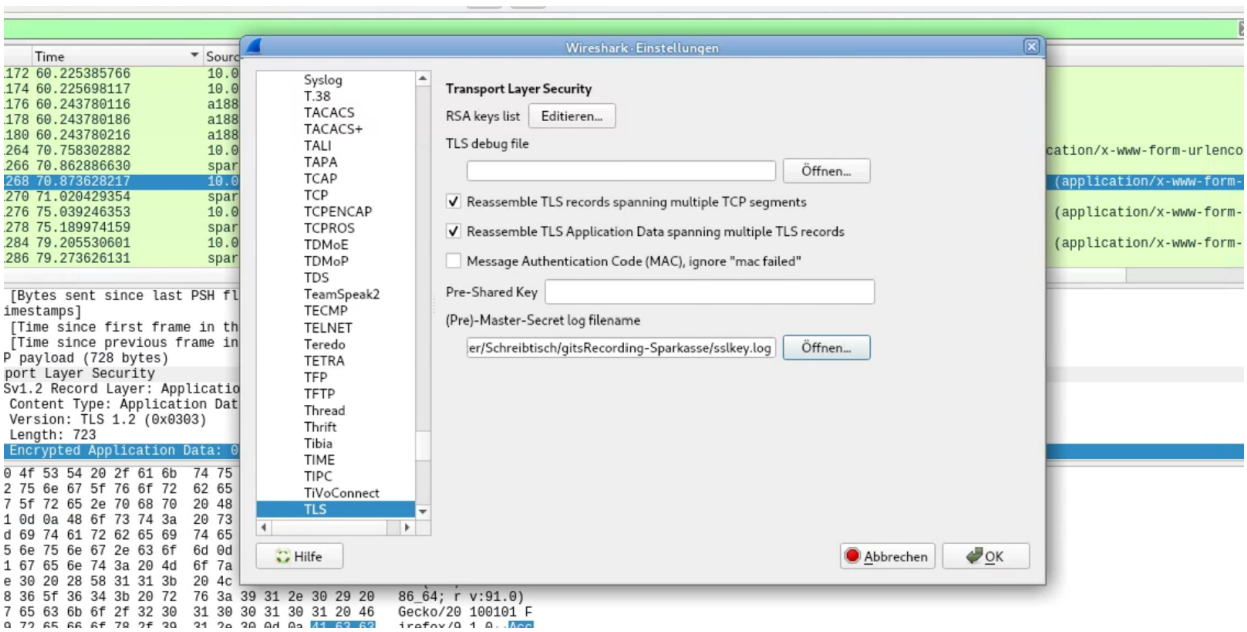
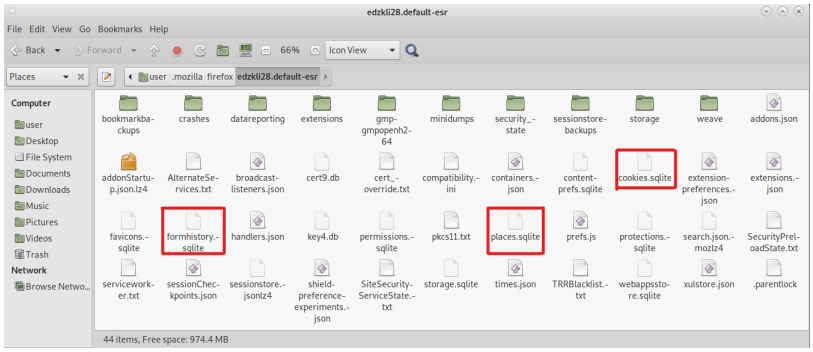


Abbildung B.3: Ein Ausschnit aus Wireshark für das Entschlüsseln der Pakete

B.2 Implementierung: Daten von Firefox (zu 4.10.1)



(a) Relevante Dateien im Firefox-Profil.

moz_cookies (10 rows)

```
SELECT * FROM 'moz_cookies' LIMIT 0,30
```

Execute

id	originAttributes	name	value	host	path	expiry	lastAccessed	creatio
1		uid21318	1202639069-20220611151...	zonestor.com	/	1654982200	1654978599553943	165497
2		coyoteTrackingCookie_2220	460131051	gewinn-ometer.com	/	1657570722	1654978722864940	165497
3		coyoteSimpleTrackingCookie	460131051	gewinn-ometer.com	/	1657570722	1654978722865216	165497
5		_GRECAPTCHA	09AiTtJzfKX16dSrtuICTHH...	www.google.com	/recaptcha	1670530806	1654978805896254	165497
8		coyoteAffiliTokenId3466	460131051	dec-voucher.gewinn24-7.de	/	1654993341	1654978941406069	165497
9		CAT	1654978940.7458	dec-voucher.gewinn24-7.de	/	1654982541	1654978941406523	165497
10		b7f11a32-e22d-4991-9145-...	sOWISC06exXMuge6aa8U_...	.cenents-meason.com	/	1655065342	1654978942088219	165497
11		cc-v4	qzk62jZJjh%2BYTGk6BFqg...	.cenents-meason.com	/	1686514942	1654978942093896	165497
12		cm_mmc	coadv	www.deutsche-familienversi...	/	1660162944	1654978944092427	165497
13		marketing_utm	%7B%22utm_source%22%...	www.deutsche-familienversi...	/	1660162944	1654978944092885	165497

(b) Cookies in der Firefox-Datenbank, Beispiel von der in Abschnitt 4.8.1 untersuchten E-Mail.

moz_formhistory (34 rows)

```
SELECT * FROM 'moz_formhistory' LIMIT 20,13
```

Execute

id	fieldname	value	timesUsed	firstUsed	lastUsed	guid
21	adressData[vorname]	billy	1	1654864187482000	1654864187482000	PgSg67oLTkWm6vka
22	adressData[name]	souls	1	1654864187482000	1654864187482000	ouwX3exOQDqcLLuR
23	adressData[email]	billysouls@outlook.de	2	1654864187482000	1654978805610000	7h5uuqOsTKGTu+IC
24	adressData[plz]	38120	2	1654864236767000	1654978899486000	zXMtvYUNSq6LiRfU
25	adressData[hausnr]	2	1	1654864236767000	1654864236767000	kRiOlyFwRSCYAR15
26	adressData[telefon1]	0391733333	2	1654864236767000	1654978862158000	E3qTBZh1SLapGZEh
27	adressData[vorname]	Billy	1	1654978805610000	1654978805610000	nOXh7kBkQ5C8SBfP
28	adressData[name]	Souls	1	1654978805610000	1654978805610000	1lQcY5dnQninQPQp
29	adressData[plz]	38510	1	1654978862158000	1654978862158000	uORdalsiQ3OWOgH7
30	adressData[ort]	Marleine	1	1654978862158000	1654978862158000	3l2lvueSRDeSh552
31	adressData[strasse]	Straße	1	1654978862158000	1654978862158000	QbgbBvc1SLSoHFbc
32	adressData[hausnr]	1	1	1654978862158000	1654978862158000	K1YR8WsGQnGwcjAL
33	adressData[strasse]	Mittelweg	1	1654978923260000	1654978923260000	/JKoDUs1SxWI+/Sa

(c) Eingegebene Formulardaten in der Firefox-Datenbank, Beispiel von der in Abschnitt 4.8.1 untersuchten E-Mail.

moz_places (314 rows)

```
SELECT * FROM 'moz_places' LIMIT 270,1
```

Execute

id	url	title	rev_host	visit_count	hidden	typed	frecency	last_visit_date	guid
303	https://deca1484.iumfragen...	Ihr 1000€ Gutschein von D...	moc.negarfmui.4841aced.	5	0	0	500	1654865900235787	Pxh8B_R-EZI

(d) Besuchte Seiten in der Firefox-Datenbank, Beispiel von der in Abschnitt 4.8.1 untersuchten E-Mail.

Abbildung B.4: Inhalt relevanter Firefox-Datenbanken

B.3 Implementierung: Daten von Thunderbird (zu 4.10.2)

moz_places (45 rows)

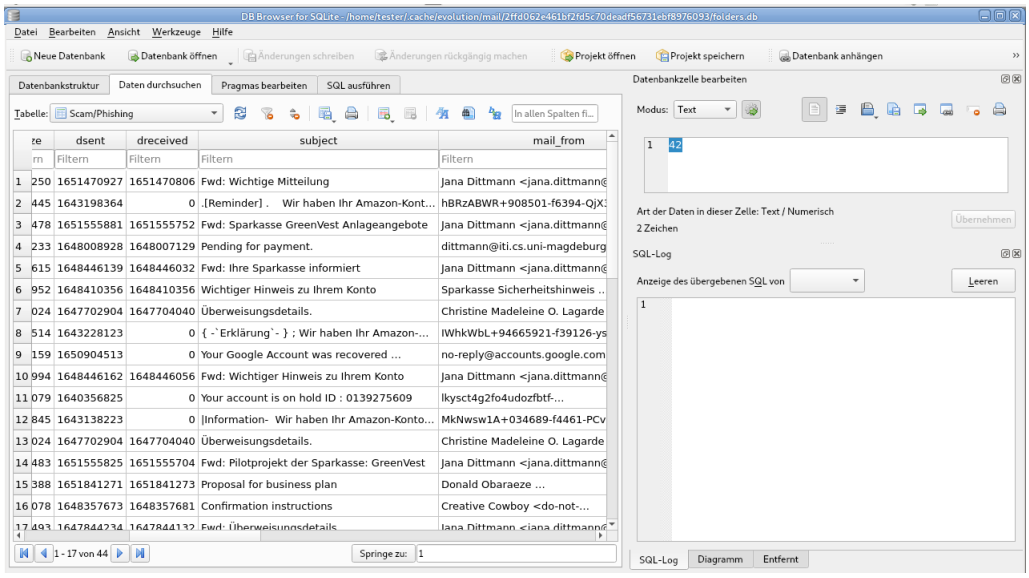
```
SELECT * FROM 'moz_places' LIMIT 33,2
```

Execute

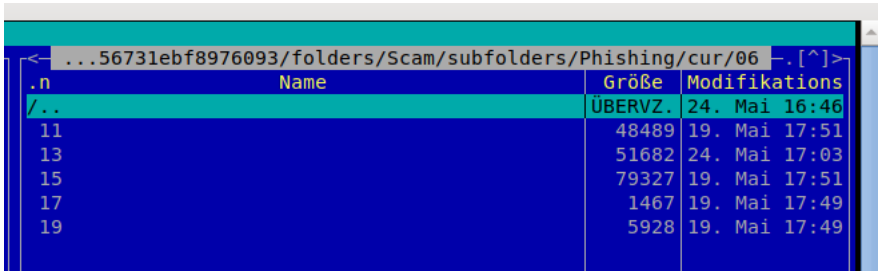
id	url	title	rev_host	visit_count	hidden	typed	frecency	last_visit_date	guid	foreign_count	url_h
34	https://storage.googleapis.c...	null	moc.sipaelgoog.egarots.	4	0	0	400	1654978593837000	pgggF6CrWBbQ	0	47359
35	https://storage.googleapis.c...	null	moc.sipaelgoog.egarots.	2	0	0	200	1654865378405000	vOslb_3_nZyc	0	47359

Abbildung B.5: Hinweis auf nachgeladene HTML-Datei, Beispiel von der in Abschnitt 4.8.2 untersuchten E-Mail.

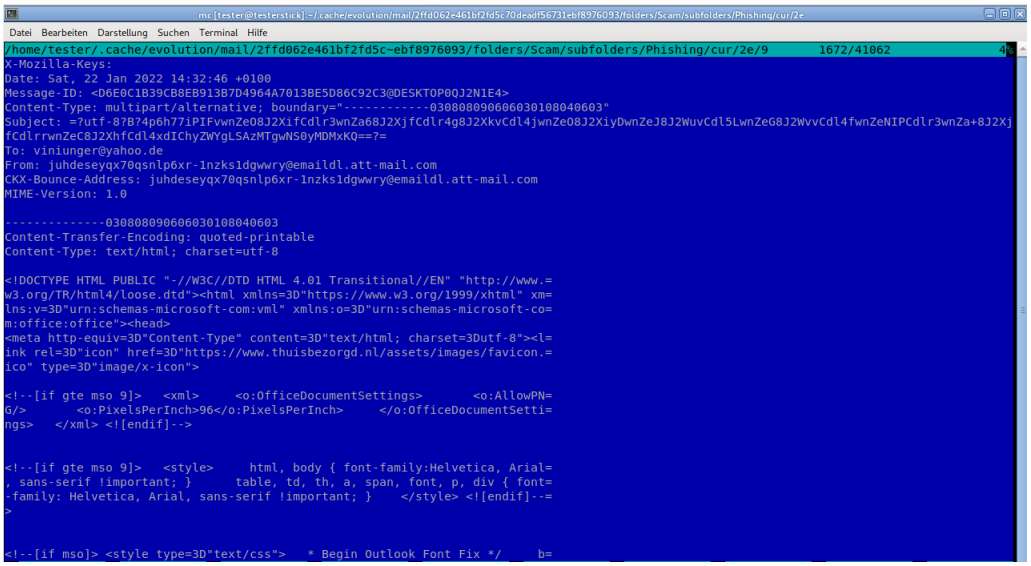
B.4 Implementierung: Daten von Evolution (zu 4.10.3)



(a) Diese Datenbank umfasst Informationen zu den Daten, welche zum Postfach des Nutzers vorliegen.



(b) Die verschiedenen Dateien, wie sie hier numerisch aufgelistet sind, enthalten entsprechend Abbildung B.6c Informationen zu den verschiedededenen E-Mails des Nutzers.



(c) Hier zu sehen: Header und Body einer E-Mail.

Abbildung B.6: Datenspeicherung in Evolution

Anhang C

C.1 Implementierung: Bash-Script zur Datenakquise (zu 4.5)

```bash
#!/bin/bash
#Version 6

#Argument supplied?
if [ $# -eq 0 ]; then
    echo "Error: Argument missing, Syntax: ./record.sh <phishingMailIdentifier>"
    echo "Aborted."
    exit
fi

PHISHING_MAIL_IDENTIFIER=$1

##################################################
# Config                                         #
##################################################

LINUX_HOME="/home/${USER}"

#Temporary output directory (will be wiped prior recording)
OUTPUT_DIRECTORY="${LINUX_HOME}/Schreibtisch/gitsRecording-${PHISHING_MAIL_IDENTIFIER}"

#Location of metadata
METADATA_LOCATION="${OUTPUT_DIRECTORY}/metadata.txt"

#Location of mitmproxy executable
MITMPROXY_EXECUTABLE="${LINUX_HOME}/opt/mitmproxy"

#Where to store keys and mitmproxy data (also save wireshark .pcapng file here)
SSLKEYLOGFILE_LOCATION="${OUTPUT_DIRECTORY}/sslkey.log"
MITMPROXY_OUTPUT="${OUTPUT_DIRECTORY}/mitmproxy.log"

#Where to find firefox sql stuff
FIREFOX_DBS="${LINUX_HOME}/.mozilla/firefox/9fwqaeub.tester"

#Where to store firefox sql stuff
FIREFOX_DBS_OUT="${OUTPUT_DIRECTORY}/firefox"

#Where to find thunderbird sql stuff
THUNDERBIRD_DBS="${LINUX_HOME}/.thunderbird/t6qhkkv7.default-default"

#Where to store thunderbird sql stuff
THUNDERBIRD_DBS_OUT="${OUTPUT_DIRECTORY}/thunderbird"

#Where to find evolution sql stuff
EVOLUTION_DBS="${LINUX_HOME}/.cache/evolution"

#Where to store evolution sql stuff
EVOLUTION_DBS_OUT="${OUTPUT_DIRECTORY}/evolution"

##################################################
# Check requirements                             #
##################################################

#Is mitmproxy available?
if [ ! -f $MITMPROXY_EXECUTABLE ]; then
    echo "Warning: mitmproxy not installed"
 read -p "Press [ENTER] to install it now (or Crtl+C to abort)" </dev/tty
 if [ ! -d "${LINUX_HOME}/opt" ]; then
  mkdir "${LINUX_HOME}/opt"
 fi
 echo "Downloading mitmproxy..."
 curl -s -o "${LINUX_HOME}/opt/mitmproxy.tar.gz" https://snapshots.mitmproxy.org/8.1.0/
     mitmproxy-8.1.0-linux.tar.gz
 echo "Extracting mitmproxy..."
 tar -xf "${LINUX_HOME}/opt/mitmproxy.tar.gz" --directory "${LINUX_HOME}/opt"
fi
```

```
if [ ! -f $MITMPROXY_EXECUTABLE ]; then
    echo "Error: Could not find mitmproxy-executable at '${MITMPROXY_EXECUTABLE}'"
    echo "Aborted."
    exit
fi

#Check if required commands installed
function checkCommandExistence() {
    cmd=$1

    if ! command -v $cmd &> /dev/null
    then
     echo "Warning: ${cmd} not installed"
     read -p "Press [ENTER] to install it now (or Crtl+C to abort)" </dev/tty
     sudo apt-get update
     sudo apt-get upgrade
     sudo apt-get install $cmd
    fi
    if ! command -v $cmd &> /dev/null
    then
        echo "Error: ${cmd} not installed!"
        echo "Aborted."
        exit
    fi
}

checkCommandExistence "wireshark"
checkCommandExistence "thunderbird"
checkCommandExistence "evolution"
checkCommandExistence "sqlite3"

#Is firefox profile available?
if [ ! -d "$FIREFOX_DBS" ]; then
    echo "Error: Could not find firefox profile at '${FIREFOX_DBS}'"
    echo "Aborted."
    exit
fi

#Is thunderbird profile available?
if [ ! -d "$THUNDERBIRD_DBS" ]; then
    echo "Error: Could not find thunderbird profile at '${THUNDERBIRD_DBS}'"
    echo "Aborted."
    exit
fi

#Is evolution data available?
if [ ! -d "$EVOLUTION_DBS" ]; then
    echo "Error: Could not find evolution data at '${EVOLUTION_DBS}'"
    echo "Aborted."
    exit
fi

##################################################
# Preparation                                    #
##################################################

#Re-Set environment variables
unset HTTP_PROXY
unset HTTPS_PROXY
unset SSLKEYLOGFILE
export HTTP_PROXY=http://127.0.0.1:8080/
export HTTPS_PROXY=http://127.0.0.1:8080/
export SSLKEYLOGFILE=$SSLKEYLOGFILE_LOCATION

echo "Environment Variables:"
echo "  - HTTP_PROXY='${HTTP_PROXY}'"
echo "  - HTTPS_PROXY='${HTTPS_PROXY}'"
echo "  - SSLKEYLOGFILE='${SSLKEYLOGFILE}'"

#Delete output directory if exists
```

```
if [ -d "$OUTPUT_DIRECTORY" ]; then
    rm -d -r $OUTPUT_DIRECTORY
fi

echo "Finishing preparation:"

#Create output directory
mkdir $OUTPUT_DIRECTORY
echo "  - Created output directory '${OUTPUT_DIRECTORY}'"

#Create SSLKEYLOGFILE
touch $SSLKEYLOGFILE_LOCATION
echo "  - Created new ssl key log file '${SSLKEYLOGFILE_LOCATION}'"

#Ready to start recording
read -p "Press [ENTER] to start recording" </dev/tty

##################################################
# Recording                                      #
##################################################

echo "Starting mitmproxy, Firefox and Wireshark, this could take a while..."
TIMESTAMP_START=$(date +%s)
xterm -e "${MITMPROXY_EXECUTABLE} -w ${MITMPROXY_OUTPUT}" &
firefox -profile $FIREFOX_DBS &> /dev/null &
wireshark &> /dev/null &
thunderbird &> /dev/null &
evolution &> /dev/null &
{
    echo "IMPORTANT:"
    echo "  - Please set ssl key log file in Wireshark to '${SSLKEYLOGFILE_LOCATION}'"
    echo "  - Please start Wireshark recording manually"
    echo "  - Remember to save Wireshark .pcapng file to '${OUTPUT_DIRECTORY}/wireshark.pcapng'"
    echo "  - Timer started. Now do your research!"
    echo "Waiting for processes to be closed:"
    echo "  - mitmproxy"
    echo "  - Firefox"
    echo "  - Wireshark"
    echo "  - Thunderbird"
    echo "  - Evolution"
} & wait

#All closed
TIMESTAMP_END=$(date +%s)
TIMESPAN=$((TIMESTAMP_END-TIMESTAMP_START))
TIMESPAN_MIN=$(($((TIMESPAN/60))+2))
echo "All recordings stopped, took ${TIMESPAN} seconds (~${TIMESPAN_MIN} minutes)"

##################################################
# Dump recent changed databases                  #
##################################################

#Sql dumping
function textifyDatabase() {
    db_path=$1
    db_out=$2

    db_name=$(basename ${1})
    for l in $(sqlite3 $db_path "SELECT name FROM sqlite_master WHERE type='table'"); do
        sqlite3 $db_path ".schema ${l}" > "${db_out}/txt/${db_name}-${l}.txt"
        sqlite3 $db_path "SELECT * FROM ${l}" >> "${db_out}/txt/${db_name}-${l}.txt"
    done
}

#Loop directory by file extension
function loopDirectoryDatabases() {
    db_dir=$1
    db_out=$2
    timespan=$3
```

```
    mkdir $db_out
    mkdir "${db_out}/sqlite"
    mkdir "${db_out}/txt"

    counter=0
    for i in $(find $db_dir -mmin -$timespan -type f -name "*.sqlite"); do
        echo "  - ${i}"
        textifyDatabase $i $db_out
        fname=$(basename ${i})
        cp $i "${db_out}/sqlite/${fname}"
        counter=$((counter+1))
    done
    for i in $(find $db_dir -mmin -$timespan -type f -name "*.db"); do
        echo "  - ${i}"
        textifyDatabase $i $db_out
        fname=$(basename ${i})
        cp $i "${db_out}/sqlite/${fname}"
        counter=$((counter+1))
    done
    if [ "$counter" -eq "0" ]; then
        echo "  - nothing changed, nothing dumped"
    fi
    return $counter
}


#Store firefox sql databases
echo "Copying/Dumping firefox dbs..."
loopDirectoryDatabases $FIREFOX_DBS $FIREFOX_DBS_OUT $TIMESPAN_MIN
COUNTER_FIREFOX_DBS=$?

#Store thunderbird sql databases
echo "Copying/Dumping thunderbird dbs..."
loopDirectoryDatabases $THUNDERBIRD_DBS $THUNDERBIRD_DBS_OUT $TIMESPAN_MIN
COUNTER_THUNDERBIRD_DBS=$?

#Store evolution sql databases
echo "Copying/Dumping evolution dbs..."
loopDirectoryDatabases $EVOLUTION_DBS $EVOLUTION_DBS_OUT $TIMESPAN_MIN
COUNTER_EVOLUTION_DBS=$?

##################################################
# Metadata & Archive                             #
##################################################

#Write metadata
TIME_NOW=$(date --debug)
touch $METADATA_LOCATION
echo "Research Object Name (Mail Identifier): ${PHISHING_MAIL_IDENTIFIER}" > $METADATA_LOCATION
echo "Recording time: ${TIME_NOW}" >> $METADATA_LOCATION
echo "Recording duration/length: ${TIMESPAN} seconds" >> $METADATA_LOCATION
echo "Profile user name: ${USER}" >> $METADATA_LOCATION
echo "Probably related database changes (${TIMESPAN_MIN} minutes):" >> $METADATA_LOCATION
echo "  Firefox: ${COUNTER_FIREFOX_DBS}" >> $METADATA_LOCATION
echo "  Thunderbird: ${COUNTER_THUNDERBIRD_DBS}" >> $METADATA_LOCATION
echo "  Evolution: ${COUNTER_EVOLUTION_DBS}" >> $METADATA_LOCATION
echo "Metadata written to '${METADATA_LOCATION}'"

#Archive
OUTPUT_DIRECTORY_NAME=$(basename $OUTPUT_DIRECTORY)
ARCHIVE_LOCATION=$(realpath "${OUTPUT_DIRECTORY}/../${PHISHING_MAIL_IDENTIFIER}.tar.gz")
echo "Processing recorded data..."
cd $OUTPUT_DIRECTORY/..
tar -czvf $ARCHIVE_LOCATION $OUTPUT_DIRECTORY_NAME
echo "Created tar archive at '${ARCHIVE_LOCATION}'"

#EOF
```