Agenda

Tim Müller, Juvan Thavalingam

6. Juni 2025

1 Sync Meeting: 02.06.2025

1.1 Stand der Arbeit

Das Read.me Installation Guide und und wie man Attribute hinzufügen (Schritt für Schritt) wurde erstellt. Die Attribute wurden mehrheitlich bei der Implementation beschreiben, es fehlen aber noch 3 Attribute

1.2 Ziele

- Teile von Resultat fertigstellen! Juvan ✓
- Kapitel Umsetzung: Die letzten vier Implementation von Attributen fertigstellen in Dokumentation beschreiben, Tim ✓
- Kapitel Umsetzung: Datenbank erwähnen! ✓
- Abomodell (dehashed, shodan) in Implementation erwähnen, Tim ✓
- In Implementation das UI kurz vorstellen! Tim ✓
- Abbildungen Titel übernehmen und ins Verzeichnis kopieren! Juvan ✓
- Unterkapitel von Resultat: AktiveTest bei Igs GMbh und Kurt-Bachmann Ag! Juvan & Tim! ✓
- Diskussion fertigstellen! Juvan ✓
- Danksagung schreiben! Tim ✓
- Codeabgabe ergänzen auf GitHub und Server! In Dokumentation Hyperlik auf githubrepo Tim ✓
- Projektmanagement schreiben! Tim (agenda...), overleaf, github, wöchentlich, hyperlink in agenda! ✓
- Einleitung! Juvan ✓
- Abstract und Zusammenfassung erstellen! Juvan ✓
- E-Mail von Stefan umsetzen! Juvan ✓
- Formatierung ändern! ✓
- Rechtschreibung ($ß- > ss$, Anführungszeichen, Grammatik), Juvan & Tim ✓
- Übergänge besser gestalten! ✓
- Quellenverzeichnis kontrollieren! ✓
- Abkürzungsverzeichnis ergänzen! ✓
- Einheitliche Erwähnung von Firmen, (Resultate, Danksagung) ✓

1.3 Offene Fragen

- Welcher Zitierstil bei Quellen? ieee?
- Soa-Eintrag ohne ipv4 und ipv6?
- Code auf Github oder ZIP-Datei?
- Resultat und Diskussion zusammenfügen statt Diskussion und Ausblick? Ausblick sind dann neue Features und in Diskussion sind Fehler, die man korrigieren muss!

  Dikussion: Was hat die Gesellschaft von dieser Arbeit? VOrteil und Nachteil? Wichtiger Schritt zum digitalen Twin. Nicht perfekt weil anfälligkeit bei gewissen Attributen? Kmus funktioniert?

2 Sync Meeting: 26.05.2025

2.1 Stand der Arbeit

Insgesamt haben wir 384 Firmen gescannt. Die Diagramme wurden schöner gestaltet, damit alle Zahlen und Texte lesbar sind. Zudem wurde sichergestellt, dass alle Texte, wie Titel, Y-Label und X-Label, korrekt sind. Es wurde mit dem Kapitel Resultate angefangen. Bei fünf Attributen wurden die Ergebnisse anhand der Grafiken beschrieben.
Zudem haben wir ein Diagramm ergänzt, das anzeigt, in wie vielen Firmen CVEs gefunden wurden. Bei Dehashed haben wir herausgefunden, dass es zwar teurer ist, einzeln nach E-Mail-Adressen zu suchen als nach der Domain, man aber genauere Ergebnisse findet. Nach geleakten Passwörtern im Zusammenhang mit Telefonnummern zu suchen, finde ich zwar spannend, aber angesichts der Kosten lohnt es sich unserer Meinung nach nicht.
Somit haben wir den praktischen Teil abgeschlossen und widmen uns nun vollständig der schriftlichen Dokumentation.

Wir haben auch ein Präsentationstemplate in Overleaf gefunden und werden dieses ergänzen.

2.2 Ziele

- Kapitel Umsetzung: Die Implementation von Attributen fertigstellen
- Kapitel Resultate fertigstellen
- Kapitel Diskussion und Ausblick fertigstellen
- Abomodell in Arbeit erwähnen
- Read.me Installation Guide und Attribute hinzufügen (Schritt für Schritt)

2.3 Offene Fragen

3 Sync Meeting: 19.05.2025

3.1 Stand der Arbeit

Wir haben eine Liste mit ca. 400 Firmenurl um Winterterthur herrum Wir haben die gefundenen Daten in unser Testsystem integriert. Bis zum jetzigen Zeitpunkt haben wir bereits mindestens 100 Firmen gescannt.

Anschliessend haben wir verschiedene Diagramme basierend auf diesen 100 Firmen erstellt. Dabei haben wir zwei Diagrammtypen verwendet:

- **Erster Diagrammtyp:** Dieser zeigt in einem Säulendiagramm, wie viele Firmen bereits geleakte E-Mail-Adressen haben und wie viele nicht. Dasselbe wurde auch für Telefonnummern dargestellt.
- **Zweiter Diagrammtyp:** Hier wird für jedes Attribut ein eigenes Säulendiagramm erstellt. Die Säulen zeigen unter anderem, bei wie vielen Firmen kein Wert zu diesem Attribut gefunden wurde oder bei wie vielen mehr als 10'000 Einträge vorhanden waren.

Daraus konnten wir unter anderem erkennen, dass bei Shodan nicht bei allen Firmen Informationen gefunden wurden. Das lässt darauf schliessen, dass besonders bei bestimmten Firmen keine öffentlich zugänglichen Daten vorhanden sind oder die Daten veraltet sein könnten.

Ausserdem haben wir bei der Firma **Kurt Bachmann AG** bezüglich der Richtigkeit des Scans nachgefragt. Dabei konnten verschiedene Erkenntnisse gewonnen werden:

- Shodan erkannte die richtige Firewall, verband diese jedoch mit einer alten Domain (mail.kurt-bachmann-ag.ch). Diese existiert seit über einem Jahr nicht mehr. Das zeigt, dass die Informationen bei Shodan teilweise veraltet sind.
- Unser Domainscan fand hingegen nur die Hauptdomain, was darauf schliessen lässt, dass dieser aktueller ist als die Daten von Shodan.
- Die auf der Webseite veröffentlichten Telefonnummern wurden alle korrekt erkannt. Weitere Telefonnummern werden ausschliesslich mündlich oder über E-Mail-Signaturen weitergegeben.
- Nicht alle E-Mail-Adressen wurden gefunden. Dies kann daran liegen, dass die Wayback Machine diese nie gescannt hat oder sie sich nicht unter den letzten 50 gespeicherten URLs befinden (aus Optimierungsgründen).
- Die restlichen Attribute wurden korrekt erfasst.

3.2 Ziele

- Kapitel Umsetzung: Die Implementation von Attributen fertigstellen
- Kapitel Resultate und Testing schreiben
- Wenn Resultate und Testing fertig gestellt wurden, dann mit Diskussion beginnen
- Abomodell in Arbeit erwähnen
- Read.me Installation Guide und Attribute hinzufügen (Schritt für Schritt)

3.3 Offene Fragen

4 Sync Meeting: 12.05.2025

4.1 Stand der Arbeit

Login funktioniert über API und das Passwort ist nicht mehr hardcoded. Wenn im Scan nichts gefunden wurde, dann wird das dementsprechend angezeigt. Der Bug mit Zertifikate wurde gefixt und die Ergebnisse werden somit nicht mehr doppelt angezeigt. Zusätzlich haben wir ein Script, was automatisch scannt und in JSON exportiert. Webseite läuft auf dem Server ist aber bis jetzt nur von localhost erreichbar. Wir haben für die Resultate eine Webseite gefunden mit 600 Urls. Das Kapitel Architektur wurde mit einer Vereinfachten Architektur ergänzt.

4.2 Ziele

- Kapitel Umsetzung: Die Implementation von Attributen fertigstellen
- Resultate und Testing beginnen
- Wenn Resultate und Testing fertig gestellt wurden, dann mit Diskussion beginnen
- Abomodell in Arbeit erwähnen
- Read.me Installation Guide und Attribute hinzufügen (Schritt für Schritt)

4.3 Offene Fragen

5 Sync Meeting: 05.05.2025

5.1 Stand der Arbeit

Alle gefundenen Bugs wurden behoben und alle Attribute wurden vollständig überarbeitet mit passenden Spalten. Anzahl Seitenzahlen bei den Buttons (zurück und Weiter) wurden eingefügt. Beschreibung von den Attributen wurden verbessert. Export JSON wurde visuell verbessert. Die Kapitel Motivation und Ausblick wurden fertig gestellt. Die Grafik Pluginsystem wurde neu erstellt und die Dokumentation eingefügt. Kapitel Einleitung ist auch fertig.

5.2 Ziele

- Kapitel Umsetzung: Die Implementation von Attributen fertigstellen
- Resultate und Testing beginnen
- Wenn Resultate und Testing fertig gestellt wurden, dann mit Diskussion beginnen
- Architektur neues Abstraktionslevel
- Bugs fixen (Zertifikate, nicht doppelt vorkommen, wegen alias)
- Klar anzeigen, welche Attribute nicht gefunden wurden
- Abomodell in Arbeit erwähnen
- Read.me Installation Guide und Attribute hinzufügen (Schritt für Schritt)

5.3 Offene Fragen

- Wie sollen wir die Vorlesungsfolien in der Quelle referenzieren?
- Sollen wir für Dehashed bezahlen?

6 Sync Meeting: 28.04.2025

Spalten kontrollieren, Digitaler Twin und Datenbank Liste schicken

6.0.1 Test

6.1 Stand der Arbeit

Wir sind dabei, die gefunden Bugs zu fixen und die Dokumentation weiter zu ergänzen. Das Kapitel Architektur wurde ergänzt aber ist verbesserungswürdig. Subdomains, Endpunkte, Zertifikate und Dienste sind jetzt vollständig implementiert und im Frontend schön designed.

6.2 Ziele

- Implementation von Attributen ergänzen
- Diskussion und Ausblick
- Bugs von Telefonnummer, E-Mail und DNS-Records fixen

6.3 Offene Fragen

- Dehashed wird momentan auf die Version 2 gebracht mit vielen Änderungen. Unter anderem muss man jetzt nicht nur Credits kaufen sondern eine Subscription.
- Aktivitätsdiagramm erwünscht?

Jan Kressebruch! Liste von Attributen Kündigungsfristen dehashed!

7 Sync Meeting: 14.04.2025

7.1 Stand der Arbeit

Es wurde an der Dokumentation weiter gearbeitet. Die Einleitung wurde zum grossen Teil fertiggestellt. Das Kapitel Theoretische Grundlagen" wurde neu geschrieben. Der Fokus lag hier auf Docker und Flutter. Das Kapitel 3 Vorgehen wurde auch angefangen. Der Fokus lag hier auf das Backend. Das Plugin-System wurde beschrieben. Später werden die einzelnen Attribute beschrieben.

Im Frontend wurde der Bug gefixt, dass Einträge nicht mehrmals gespeichert werden. Neu werden die Beschreibungen der Attribute angezeigt und die Attribute Telefonnummer und Shodan für Services und CVE hinzugefügt. Dehashed wurde im Backend leicht vorbereitet für die Zukunft.

7.2 Ziele

- Mit Dokumentation weiterarbeiten
  - Attribute näher beschreiben
  - Frontend beginnen
- Domain an A und AAAA Rekord anhängen
- PTR Record Bug Fix
- Json Hierarchie bearbeiten (Description und Verschachtlung)
- Error- und Infohandling
- Suchfunktion ergänzen

7.3 Offene Fragen

8 Sync Meeting: 07.04.2025

8.1 Stand der Arbeit

Es wurde eine separate Filterfunktion für die Attribute implementiert. Durch Selektion der Checkbox können die Attribute ein- und ausgeblendet werden. Darüber hinaus wurde das Problem der übermässigen Grösse der Tabellen (Zertifikate) behoben. Die Tabellen sind nun auf eine fixe Grösse von 10 begrenzt. Darüber hinaus wurde die Implementierung von zwei Buttons mit den Bezeichnungen Nextünd "Back"vorgenommen. Diese ermöglichen die Anzeige der nächsten bzw. letzten zehn Ergebnisse der Tabelle. Diese Änderungen führten zu einer signifikanten Verbesserung der Performance.
Darüber hinaus wurden Endpoints sowie E-Mails im Frontend ergänzt. Die Erfassung dieser Elemente durch die Waybackmaschine erfolgt zuverlässig. Im Falle der E-Mail-Funktion wird mithilfe von Dehashed überprüft, ob eine Leakage bereits stattgefunden hat.

8.2 Ziele

- Fehlende Attribute der bestehenden Pythonscripts in Frontend ergänzen.
- Description in Frontend bei Attribute anzeigen
- Mit Dokumentation beginnen.
- Json Hierarchie bearbeiten (Description und Verschachtlung)
- Error- und Infohandling
- Suchfunktion ergänzen

8.3 Offene Fragen

9 Sync Meeting: 31.03.2025

9.1 Stand der Arbeit

Wir haben das Backend und Frontend überarbeitet, sodass sie vollständig modular aufgebaut sind. Im Frontend muss nun lediglich eine Zeile ergänzt werden, wenn ein neues Attribut hinzugefügt wird. Diese Vorgehensweise soll beibehalten werden. Im Backend müssen jeweils eine Datei 'attribut_plugin.py' und 'attribut_lookup.py' ergänzt werden.
Zur Überprüfung der Modularität wurde das Attribut 'Zertifikat' hinzugefügt, dies hat erfolgreich funktioniert. Eine erste Version der Fehler- und Informationsanzeige sowie der Tabellenanzeige wurde implementiert.
Die Shodanscripts wurden verbessert. Das Problem war, das ich früher in Shodan nach IP-Adressen gesucht habe. Jetzt benutze ich 'hostname:domain' und es funktioniert so wie erwartet.
Ein einfaches Login-System wurde erstellt, das den Zugriff durch Unbefugte einschränkt.

9.2 Ziele

- Filterfunktion
- Suchfunktion
- Frontend verschönern! (besonders die Tabelle)
- Error- und Infohandling
- Description von Backend der Attribute im Frontend anzeigen
- Webseite auf Server hosten
- Restliche Attribute, welche schon in Pythonformat vorhanden sind, in Webseite integrieren.
- Json Hierarchie bearbeiten (Description und Verschachtlung)
- Certificate Tabellengrösse begrenzen (Performence)

9.3 Offene Fragen

- issuspicious weglassen!

10 Sync Meeting: 24.03.2025

10.1 Stand der Arbeit

Wir verfügen nun über folgende Attribute in einem Python-Skript:

- Alle DNS-Records
- Zertifikate
- Endpunkte einer Domain
- E-Mails und Telefonnummern, die in diesen Endpunkten gefunden werden können
- Subdomains
- Geografische Informationen einer IP
- Betriebssystem
- Services und ihre Ports sowie dazugehörige CVEs

Unser Programm ist mittlerweile eine Webseite, die jedoch derzeit noch lokal läuft. Bisher ist es möglich, einzelne DNS-Rekordtypen sowie Subdomains zu scannen, im Textformat anzuzeigen und im JSON-Format zu exportieren. Wird ein Attribut gescannt, wird es automatisch in die Datenbank übernommen. Zum Starten von Frontend, Backend und Datenbank genügt der Befehl: `docker-compose up --build`.

10.2 Ziele

- Die Webseite modularer gestalten.
- Die Attribute, die derzeit nur im Python-Format existieren, aber noch nicht an die Webseite angebunden sind, sollen integriert werden.
- Die SSH-Keys sollen beide an Thomas gesendet werden. ✓
- Sobald Thomas Zeit hat, uns Zugriff zu gewähren, soll die Webseite auf dem Server installiert werden.
- Json Hierarchie bearbeiten (description und Verschachtlung)
- Suchfunktion Gui
- Filterfunktion
- Zugriffschutz (Login)✓
- Dokumentation
- Description Attribut mit Maus
- refactoring attribute names into config file
- Zeitintervall suchen
- automatisiert speichern?

10.3 Offene Fragen

11 Sync Meeting: 17.03.2025

11.1 Stand der Arbeit

Die Attribute „DNS Records“, „Endpunkte einer Webseite“, „Subdomains“ und „öffentliche E-Mails und Telefonnummern“ wurden in seinen Grundzügen ergänzt. Ein einfaches Frontend mit Flutter wurde aufgesetzt und funktioniert. Das Frontend wurde erfolgreich mit der Datenbank verbunden.

11.2 Ziele

- Flutter mit Datenbank verbinden ✓
- Flutter ein neues Fenster mit Ergebnissen erweitern.
- Funktionierender Ablauf: URL Eingabe und Erhalt der Ergebnisse mit Speicherung in der Datenbank.
- Attribute verschönern und verfeinern.

11.3 Offene Fragen

Folgende API's wollen wir benützen:

- Kostenlos aber mit Anfrage:
  - Zefix / Handelsregister API
- Kostenpflichtig mit Konto
  - Shoaden API
    - 69 Dollar / Monat
    - Scan up to 5,120 IPs per month
  - Dehashed
    - 15 Dollar / Monat
    - Unlimited Assed Search

12 Sync Meeting: 03.03.2025

12.1 Stand der Arbeit

Wir haben alle unsere Ziele erreicht. Die relevanten Scripts von Docker wurden auf Github hochgeladen. Wir können jetzt Ip Adressen anhand von Domainnamen erhalten. Das Blockdiagramm und Frontend Design wurden vorzeitig fertiggestellt, müssen aber sicherlich in naher Zukunft anhand der neu erworbenen Kenntnisse weiter bearbeitet werden. Dokumentation wurde ergänzt.

12.2 Ziele

- Offene Ports & Dienste & OS
- Docker weiter konfigurieren
- Attribute „Offene Ports & Dienste & OS"dokumentieren
- Kapitel Docker weiter bearbeiten
- Ein Container reicht?
- Datenbankzugriff?
- Wichtige API heraussuchen, welche kostenpflichtig sind sowie sowie eine Email benötigen
- Heraussuchen was offiziell OSINT Konform ist.

12.3 Offene Fragen

- Feedback zu Blockdiagramm und Frontend Design Prototyp
- Sollten wir ein Container erstellen, der alle anderen Container startet und abarbeitet?

13 Sync Meeting: 24.02.2025

13.1 Stand der Arbeit

Wir haben die Attribute festgelegt und die Schwierigkeit ihrer Umsetzung bewertet. Zudem wurde ein detaillierter Terminplan für die gesamte Bachelorarbeit erstellt. Jede Woche ist für die Implementierung eines Attributs vorgesehen, parallel dazu wird in der schriftlichen Arbeit die jeweilige Umsetzung dokumentiert. Die letzten Wochen der Bachelorarbeit sind hauptsächlich dem Testing und der Fehlerbehebung gewidmet. Drei Related Works wurden gefunden und werden in dieser Woche analysiert.

Attribute:

- Subdomains & IP Adressen
- Offene Ports & Dienste & OS
- Öffentliche E-Mails & Telefonnummern von Mitarbeitenden & der Firma
- Endpunkte einer Webseite (api/admin/test)
- Bekannte Schwachstellen in genutzter Software & „neue Attribute“ in Vorgehen ergänzen
- TLS-/SSL-Versionen & Zertifikate
- E-Mail-Spoofing (SPF/DKIM/DMARC)
- Content Management System
- Darknet-Präsenz

13.2 Ziele

- Blockdiagramm erstellen ✓
- Docker-Projekt erstellen ✓
- Frontend Design Prototyp ✓
- IP Adressen umsetzen ✓
- Ausgangslage dokumentieren (Tim)
- Attribut „Attribute suchen“ in Vorgehen dokumentieren (Tim)
- ”Docker Erstellung”dokumentieren (Juvan)
- ÏP Adressen”dokumentieren (Juvan)

13.3 Offene Fragen

- Sollen wir dieses Projekt mit dem Gedanken entwickeln, dass diese Software im Bereich von Projekt Cyren veröffentlicht wird? Wenn ja, stellen sich neue Sicherheitsbedenken. Diese sind wie folgt:
  - Dürfen wir Accounts erstellen um bestimmte Tools (Hunter.io) zu verwenden? In einem öffentlichen Projekt würde dies nicht gehen.

14 Sync Meeting: 17.02.2025

14.1 Stand der Arbeit

Wir wollen eine Gui-Applikation mit Flutter und Python erstellen. Diese hat Zugriff auf Docker-Container. Der Docker-Container kann mit Linux Commands und Python Scripts Open Source Intelligence (OSINT) durchführen.

Techniken:

- Port Scanning
- Service Enumeration
- Banner Grabbing
- Web Spidering
- Search Engine Queries
- WHOIS Datenbank
- DNS Records
- Social Media Analyse
- Web Archiv Analyse

Informationen:

- Verwendete Technologien und Infrastruktur
- Anzahl Mitarbeitende
- Mitarbeiter Informationen (Telefonnummer, E-Mail, Jobfunktion, etc.)
- Branche der Firma
- Leaks

14.2 Ziele

- Terminplan erstellen ✓
- Systemarchitektur erstellen (Klassendiagramm, Sequenzdiagramm)
- Ein Prototyp mit Flutter und Docker erstellen.
- Grobstruktur der Bachelorarbeit festlegen. ✓
- Kapitel Theoretische Grundlagen anfangen.

14.3 Offene Fragen

- Ist die Idee mit Flutter und Docker gut?
- Ist das Sequenzdiagramm gut?
- Was können wir noch bei Techniken und Informationen ergänzen?
- Wie viele Seiten muss die Arbeit haben?