**Dokumentation**

**Projekt Brute-Force**

*Projektarbeit Matlab Workshop*

Fachhochschule Nordwestschweiz

Studiengang Systemtechnik

Klasse 8Sbb

Dozent: Prof. Dr. Norbert Hofmann

Projektteam: Adrian Gonzalez

Bruno Hürzeler

Brugg-Windisch, Freitag, 25. Mai 2018, V 1.0000.00

# Inhalt

[Inhalt 2](#_Toc514925673)

[Änderungen 3](#_Toc514925674)

[Abbildungen 3](#_Toc514925675)

[Tabellen 3](#_Toc514925676)

[Formeln 3](#_Toc514925677)

[1 Aufgabenstellung 4](#_Toc514925678)

[1.1 Rahmenbedingungen 4](#_Toc514925679)

[1.2 Projektidee 4](#_Toc514925680)

[1.3 Implementierung / Tools 4](#_Toc514925681)

[1.4 Funktionsbeschreibung 5](#_Toc514925682)

[2 Programmaufbau 6](#_Toc514925683)

[3 Bedienung der Software 7](#_Toc514925684)

[3.1 Einstellungen von Matlab 7](#_Toc514925685)

[3.2 Programm starten 8](#_Toc514925686)

[3.3 System evaluieren 9](#_Toc514925687)

[3.4 Daten eintragen und Einstellungen auswählen 10](#_Toc514925688)

[3.5 Brute-Forcing ohne GPU starten 11](#_Toc514925689)

[3.6 Erfolgreiches Brute-Forcing ohne GPU 13](#_Toc514925690)

[3.7 Abbruch des Brute-Forcing 14](#_Toc514925691)

[3.8 Brute-Forcing mit der GPU 15](#_Toc514925692)

[3.9 Generieren eines Hashs 16](#_Toc514925693)

[4 Projektmanagement 20](#_Toc514925694)

[4.1 Soll – Ist Vergleich des Lastenheftes 20](#_Toc514925695)

[4.2 Commits 20](#_Toc514925696)

[4.3 Aufwand pro Funktion 20](#_Toc514925697)

[5 Abnahmetests 21](#_Toc514925698)

[5.1 Abnahmetest der Module in «functions.cpp» 21](#_Toc514925699)

[5.2 Abnahmetest der Systemanforderungen 27](#_Toc514925700)

[5.3 Abnahmetest des Gesamtsystems 30](#_Toc514925701)

[5.4 Geplante Optimierungen 32](#_Toc514925702)

[6 Ehrlichkeitserklärung 33](#_Toc514925703)

## Änderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Index** | **Kapitel Verweis** | **Dok. Rev.** | **Datum / Wer** | **Zweck** |
| 1 | Alle | V00.0000.00 | 17.05.18 / Hub | Erstellung des Dokumentes |
| 2 | Kapitel 2 | V00.0000.01 | 24-05.18 / Hub | Ausarbeitung des Kapitels |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Tabelle 1 Änderungen

## Abbildungen

[Abbildung 1 Programmaufbau 6](file:///C:\DATA\FHNW\8_Semester\matl\matlProject\doc\matl_Dokumentation_Gonzalez_Huerzeler.docx#_Toc514925704)

[Abbildung 2 Matlab Einstellungen 7](file:///C:\DATA\FHNW\8_Semester\matl\matlProject\doc\matl_Dokumentation_Gonzalez_Huerzeler.docx#_Toc514925705)

[Abbildung 3 Programm starten 8](file:///C:\DATA\FHNW\8_Semester\matl\matlProject\doc\matl_Dokumentation_Gonzalez_Huerzeler.docx#_Toc514925706)

[Abbildung 4 System-Evaluierung 9](file:///C:\DATA\FHNW\8_Semester\matl\matlProject\doc\matl_Dokumentation_Gonzalez_Huerzeler.docx#_Toc514925707)

[Abbildung 5 Dateneingabe nach Evaluation 10](#_Toc514925708)

[Abbildung 6 Bereit für Brute-Forcing ohne GPU 11](#_Toc514925709)

[Abbildung 7 Nach Brute-Forcing-Start ohne GPU 12](#_Toc514925710)

[Abbildung 8 Erfolgreiches Brute-Forcing ohne GPU 13](#_Toc514925711)

[Abbildung 9 Abbrechen des Brute-Force-Vorgangs 14](#_Toc514925712)

[Abbildung 10 Aktivieren der GPU 15](#_Toc514925713)

[Abbildung 11 Hash Generator 16](#_Toc514925714)

## Tabellen

[Tabelle 1 Änderungen 3](#_Toc514925715)

[Tabelle 2 Beschreibung der Elemente 6](#_Toc514925716)

[Tabelle 3 Zeitaufwand in Stunden 20](#_Toc514925717)

## Formeln

[Formel 3: Refresh-Rate Display 20](#_Toc514337662)

# Aufgabenstellung

## Rahmenbedingungen

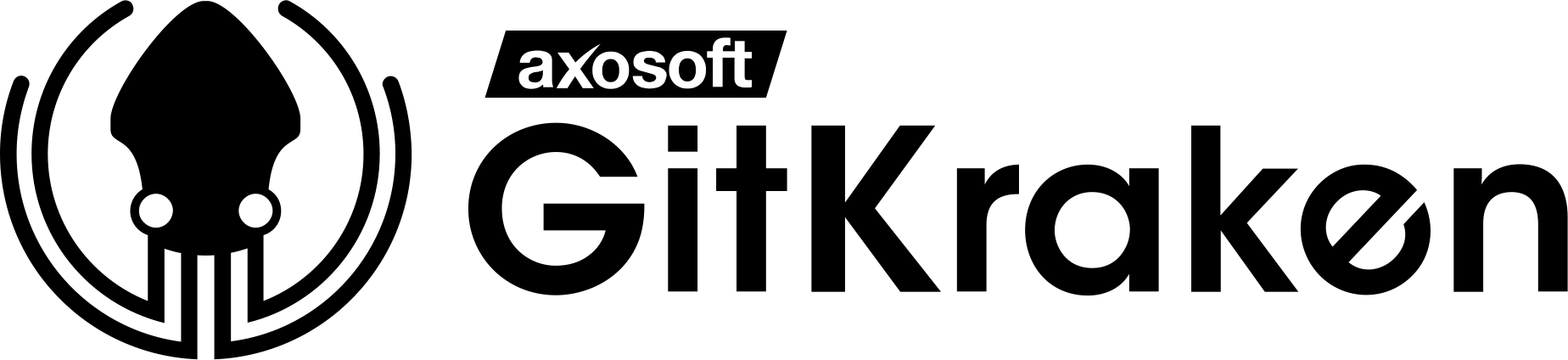
In dem Projekt sollen Daten aus Benutzereingaben, Daten oder Bildern eingelesen, gefiltert oder verändert werden. Die Ergebnisse dieser Operation werden graphisch dargestellt (Plots, Bilder, Video) und die Ergebnisse wieder auf der Festplatte abgespeichert.

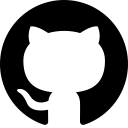
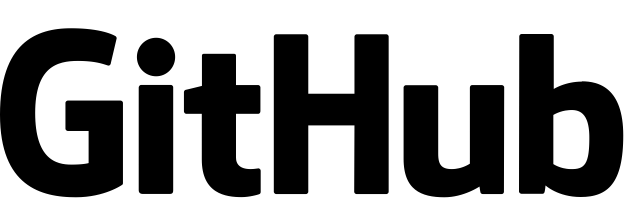
Die Arbeit wird während des Frühlingssemesters durchgeführt. Während der Unterrichtszeit des Frühlingssemesters sollten zusammen mindestens 40 Stunden investiert werden.

## Projektidee

In dieser Projektarbeit wird eine Software entwickelt, welche für das Entschlüsseln von Passwörtern oder Hashes genutzt wird. Die Entschlüsselung soll mit der *Brute-Force-Methode* realisiert werden. Da in der heutigen Zeit der Digitalisierung der Schutz der persön-lichen Daten immer wichtiger wird, kann anhand dieser Software z.B. die Stärke des Pass-wortes bestimmt werden.

## Implementierung / Tools

Die Implementierung erfolgt mittels folgenden Tools:

* MATLAB®, *Version R2017b (9.3.0.713579)*
* GitHub
* GitKraken, *Version 3.6.1*

Das GitHub wird als Versionenkontrolle verwendet, um möglichst effizient im Team arbeiten zu können. Das Projektmanagemet wurde mittels dem integrierten IssueTool «Glo» von GitKraken durchgeführt.

Das GitHub kann unter folgender URL abgerufen und ggf. heruntergeladen / geklont werden:

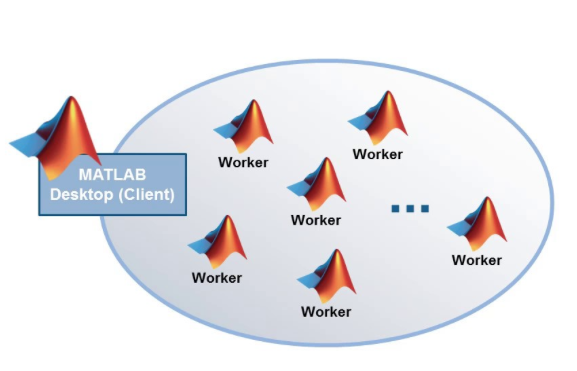
<https://github.com/adrn1990/matl_project.git>

## Funktionsbeschreibung

Die Funktion der Software liegt neben der Interaktion mit dem Benutzer im Wesentlichen im Teil des BruteForce-Vorgangs. Um diesen so effizient wie möglich zu gestalten, wird die MATLAB® Toolbox «Parallel Computing Toolbox» verwendet und dabei zwei Strategien verfolgt. Einerseits werden die Passwörter vom ersten bis zum letztmöglichen Passwort durchprobiert. Andererseits werden Anhand von Zufallszahlen Passwörter generiert, welche danach gehasht werden und mit dem ursprünglichen Hash verglichen werden.

Die Applikation erkennt die vorhandenen Ressourcen und nutzt die Cluster, die dem User zur Verfügung stehen. Anhand des ausgewählten Clusters teilt die Applikation die Arbeit auf die vorhandenen Workern homogen auf. Danach übermittelt der Client den Workern die Arbeit. Sobald ein Worker das Passwort gefunden hat, werden alle anderen Worker unterbrochen und das Passwort wird an die Benutzeroberfläche zurückgegeben und angezeigt.

Im folgendem Bild sind sechs Worker dargestellt. Mit einem Cluster von dieser Grösse würde die Arbeit von Iterationen auf die Worker aufgeteilt werden.



# Programmaufbau



**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**10**

**11**

**12**

**13**

**14**

**15**

**16**

**17**

**18**

**19**

**20**

Abbildung 1 Programmaufbau

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Element** | **Sub-Element** | **Beschreibung** |
| 1 | File | *New run…* | Alle Daten bereinigen |
|  |  | *Save…* | Speichern der Daten aus dem Log-Monitor |
|  |  | *Exit* | Beenden des Programmes |
| 2 | ? | *About…* | Informationen über die Software und Entwickler |
| 3 | Evaluate System |  | Starten der Systemevaluation |
| 4 | Mode |  | Modus Auswahl |
| 5 | Input |  | Eingabefeld für das Passwort oder Hash |
| 6 | Encryption |  | Auswahl des Verschlüsselungs-Algorithmus |
| 7 | Cluster |  | Auswahl des zu verwendenden Clusters |
| 8 | GPU |  | An- und Abwählen der GPU |
| 9 | Start |  | Start Brute-Force |
| 10 | Abort |  | Abbruch eines Brute-Force Vorganges |
| 11 | Status |  | Status- Fortschrittsanzeige in Prozent |
| 12 | Log-Monitor |  | Informationsanzeige |
| 13 | CPU Graph |  | Visualisierung des Gebrauchs der CPU Leistung |
| 14 | GPU Graph |  | Visualisierung des Gebrauchs der GPU Leistung |
| 15 | CPU load |  | Anzeige des Gebrauchs der CPU Leistung |
| 16 | CPU temp |  | Anzeige der CPU Temperatur |
| 17 | GPU load |  | Anzeige des Gebrauchs der GPU Leistung |
| 18 | GPU temp |  | Anzeige der GPU Temperatur |
| 19 | Result |  | Ausgabefeld in welchem das Resultat angezeigt wird |
| 20 | Create a hash |  | Link auf eine Webseite um einen Hash zu generieren |

Tabelle 2 Beschreibung der Elemente

# Bedienung der Software

## Einstellungen von Matlab

Abbildung 2 Matlab Einstellungen

Um den vollen Funktionsumfang der Brute-Force Software nutzen zu können, sollte das MATLAB® aus Administrator ausgeführt werden. Die Abbildung 1 zeigt auf, wie Sie das Programm standardmässig als Administrator ausführen.

MatlabXXXXy.exe ->Rechte Maustaste -> Eigenschaften -> Erweitert -> Hacken setzen

## Programm starten

Abbildung 3 Programm starten

Das Starten der Brute-Force Software erfolgt mittels der im Projekt enthaltenen Anwendung.

Doppelklick auf Brute Force VX.X.exe

## System evaluieren

Abbildung 4 System-Evaluierung

Damit die Brute-Force Software funktionieren kann, muss als erstes das System evaluiert werden. Solange die Systemdaten nicht erfasst wurden, können keine anderen Bedienelemente ausgewählt werden.

## Daten eintragen und Einstellungen auswählen

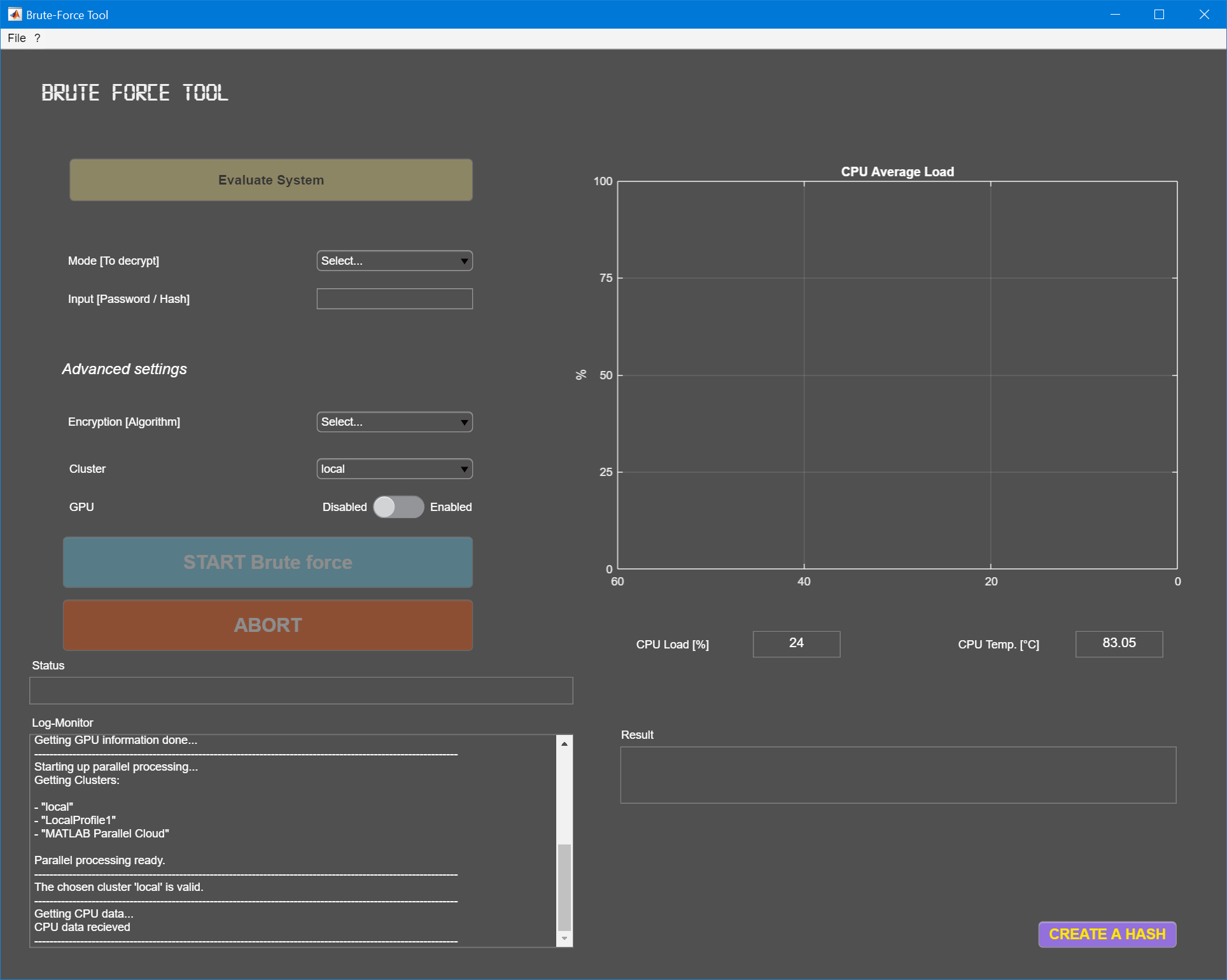


Abbildung 5 Dateneingabe nach Evaluation

Nach erfolgreichem Evaluieren des Systems können die evaluierten Daten aus dem Log-Monitor entnommen werden. Zudem ist es nun möglich, die Daten für das Brute-Forcing einzustellen.

## Brute-Forcing ohne GPU starten

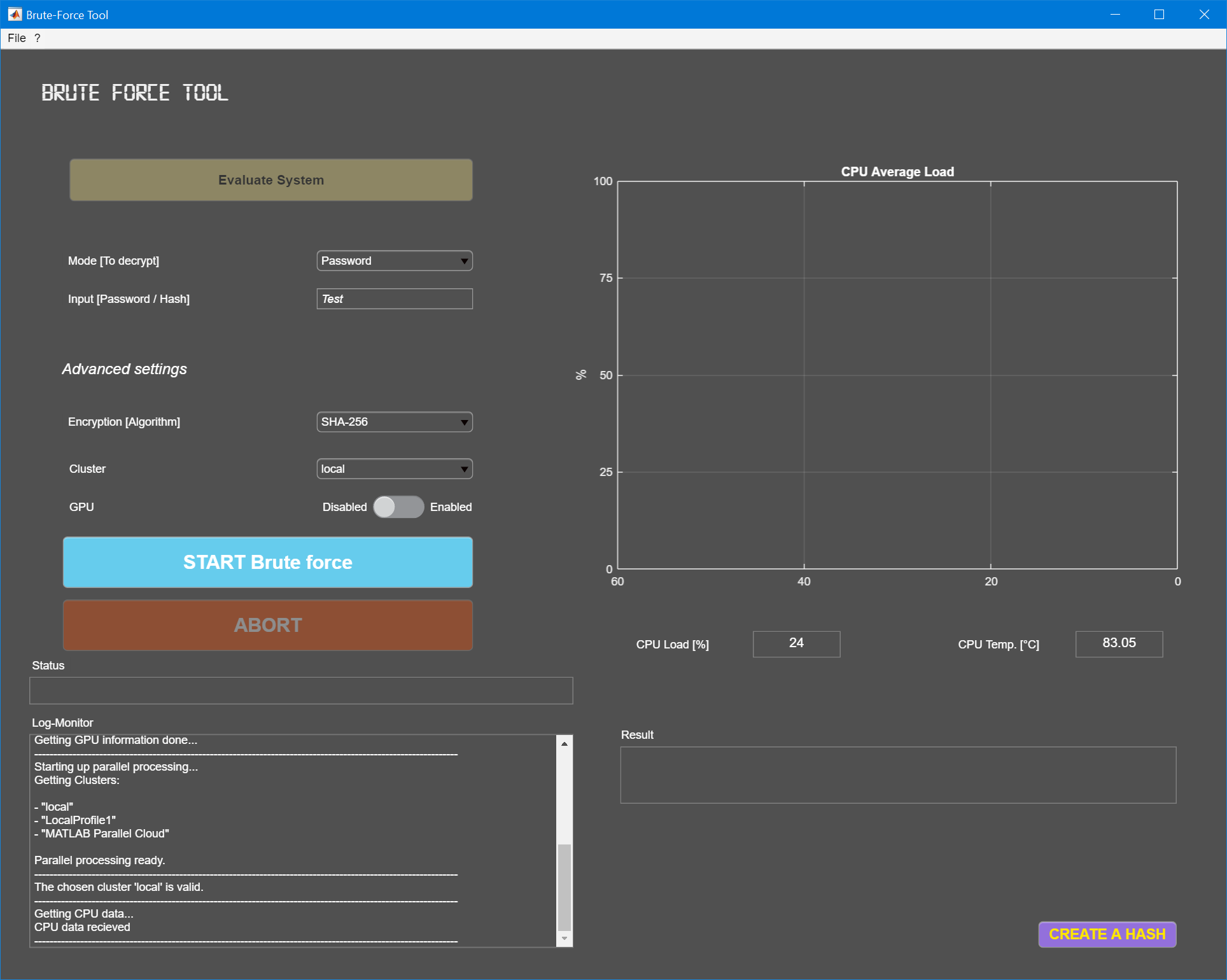


Abbildung 6 Bereit für Brute-Forcing ohne GPU

Sobald alle erforderlichen Daten, welche für das Brute-Forcing benötigt werden, eingestellt sind wird das Start-Bedienelement aktiv und ein Brute-Forcing kann gestartet werden.

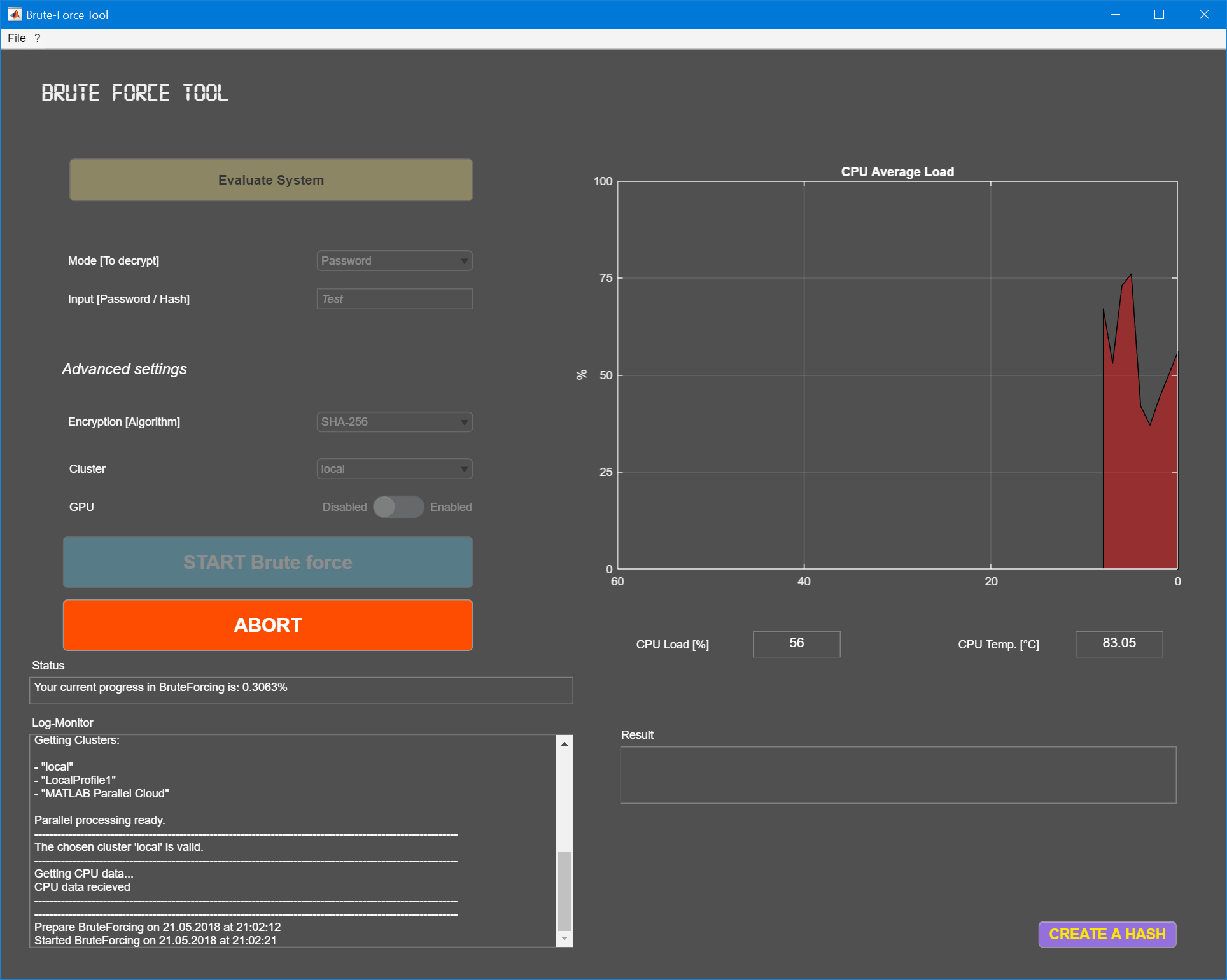


Abbildung 7 Nach Brute-Forcing-Start ohne GPU

Aus dem Log-Monitor kann entnommen werden, wann die Software sich vorbereitet und anschliessend der Brute-Force-Prozess begonnen hat.

Weiter können aktuelle Informationen aus der Grafik oder der Statusanzeige entnommen werden.

## Erfolgreiches Brute-Forcing ohne GPU

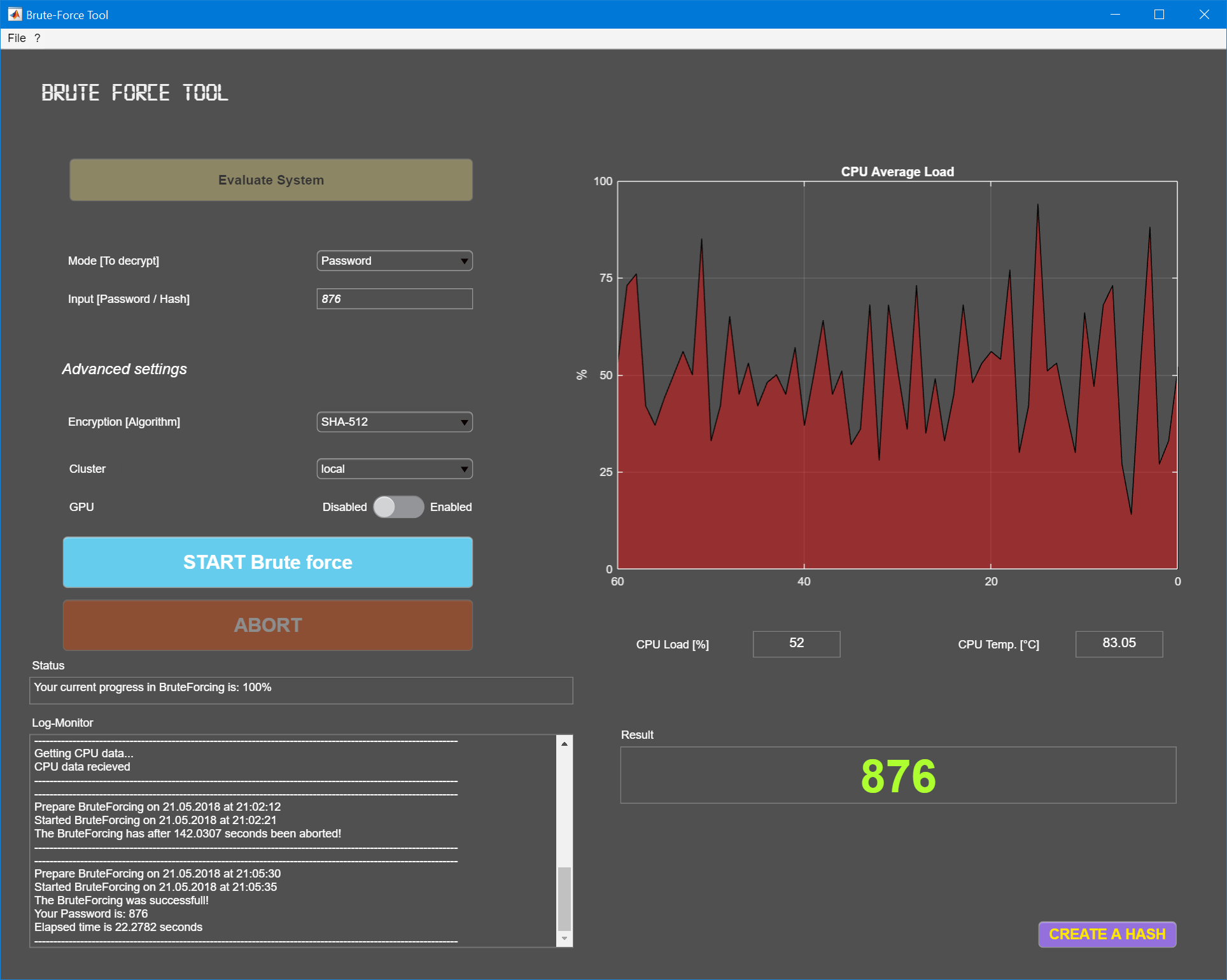


Abbildung 8 Erfolgreiches Brute-Forcing ohne GPU

Nach erfolgreichem Brute-Forcing des Passwortes oder Hashes wird dieses im Resultat-Ausgabefeld visualisiert und ein neuer Vorgang kann anschliessend wieder gestartet werden.

## Abbruch des Brute-Forcing

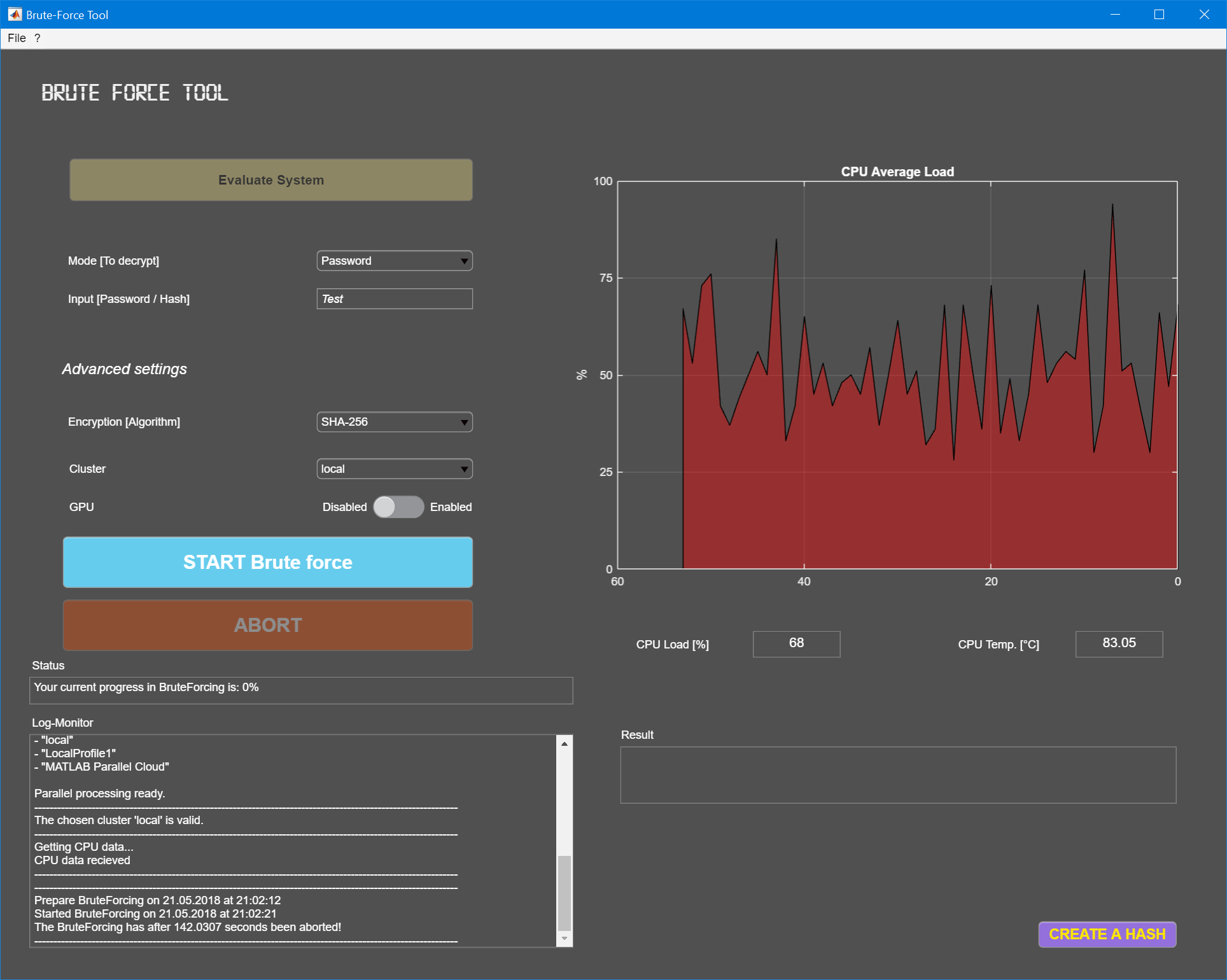


Abbildung 9 Abbrechen des Brute-Force-Vorgangs

Für einen Abbruch des Vorganges kann dies mittels des Abort-Knopfes durchgeführt werden. Sobald der Star-Knopf aktiv wird und im Log-Monitor das Abbrechen bestätigt wurde, kann ein neuer Prozess gestartet werden.

Dieser Vorgang ist unabhängig mit welcher Hardware der Prozess gestartet wurde.

## Brute-Forcing mit der GPU

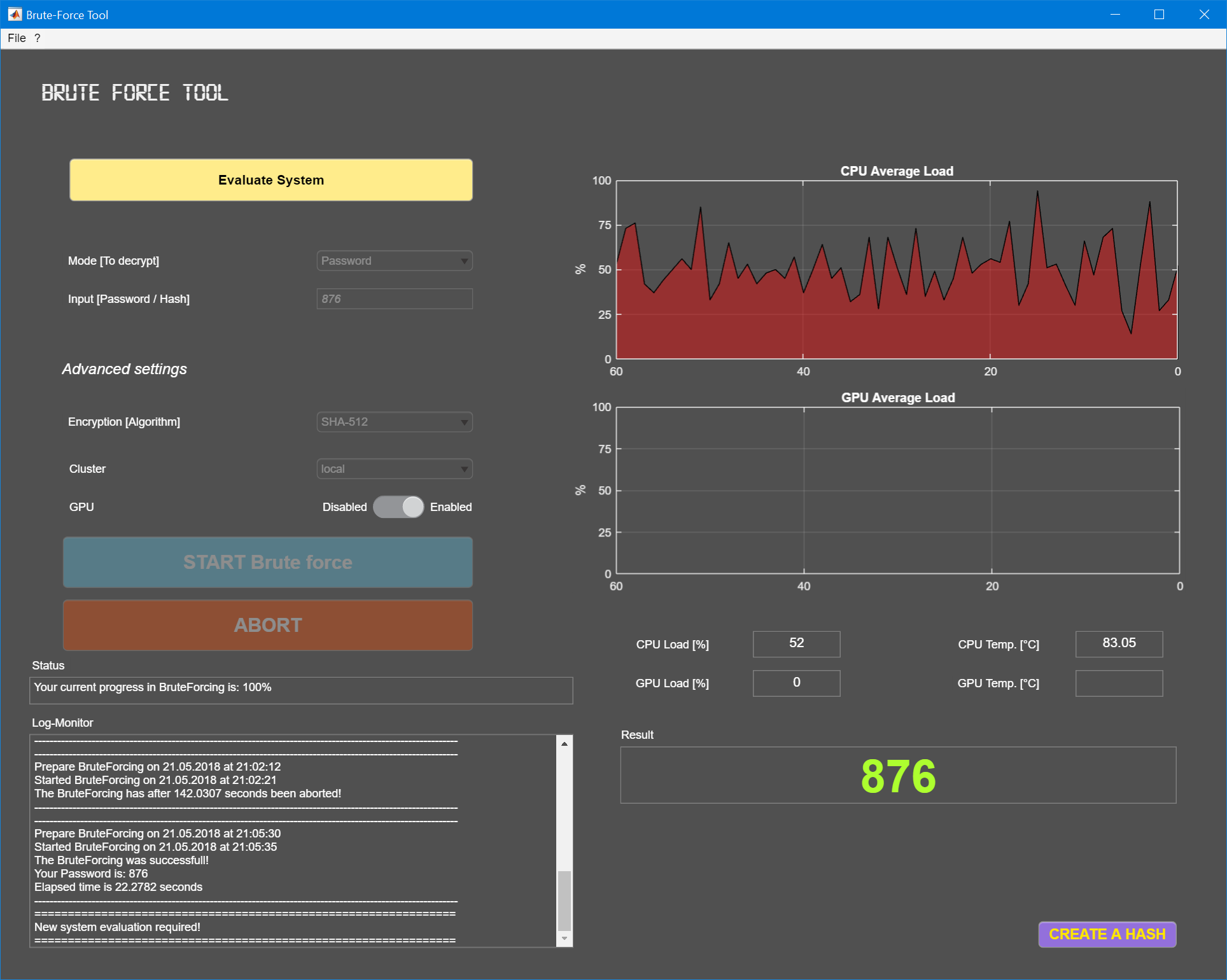


Abbildung 10 Aktivieren der GPU

Sobald die GPU auf aktiv gesetzt wird und diese zuvor nie verwendet wurde, ist eine neue Systemevaluation notwendig. Dies wird dem User per Log-Monitor mitgeteilt. Des Weiteren wird das User Interface mit den GPU-Daten erweitert.

Sobald das System nochmals evaluiert wurde, kann die Software wie ab Kapitel 3.4 betrieben werden.

## Generieren eines Hashs

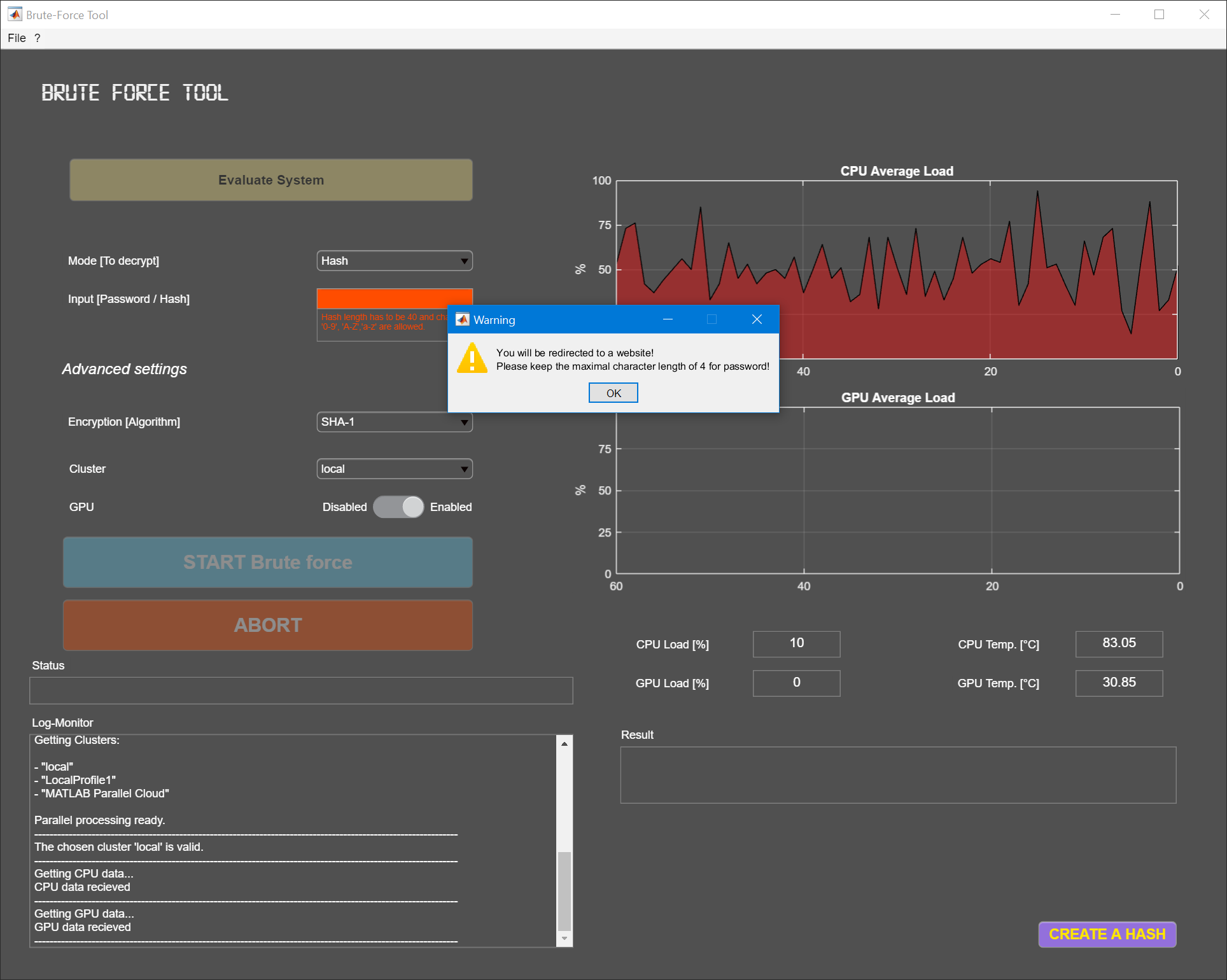


Abbildung 11 Hash Generator

Sobald der Modus auf Hash gestellt wird, überprüft die Software das Eingabefeld. Nun muss dort zwingend ein Hash, welcher abhängig von der Verschlüsselung ist, eingetragen werden.

Dafür ist der Knopf «Create a hash» unten rechts vorgesehen. Wird dieser gedrückt erscheint eine Warnmeldung in der nochmals auf die Passwortlänge eingegangen wird. Durch bestätigen dieser Warnmeldung öffnet sich im Browserfenster der Hash Generator (siehe Abbildung 12).

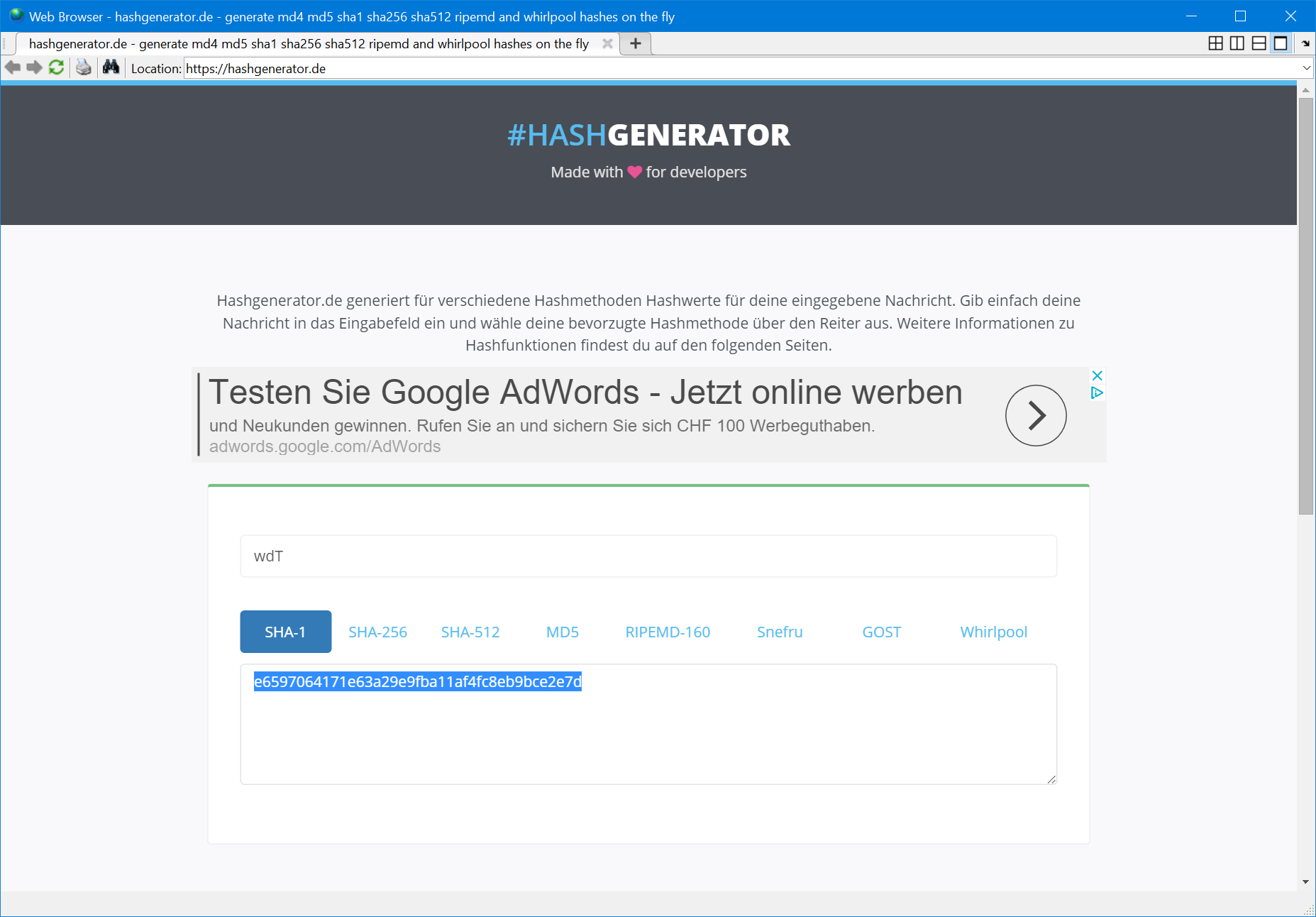
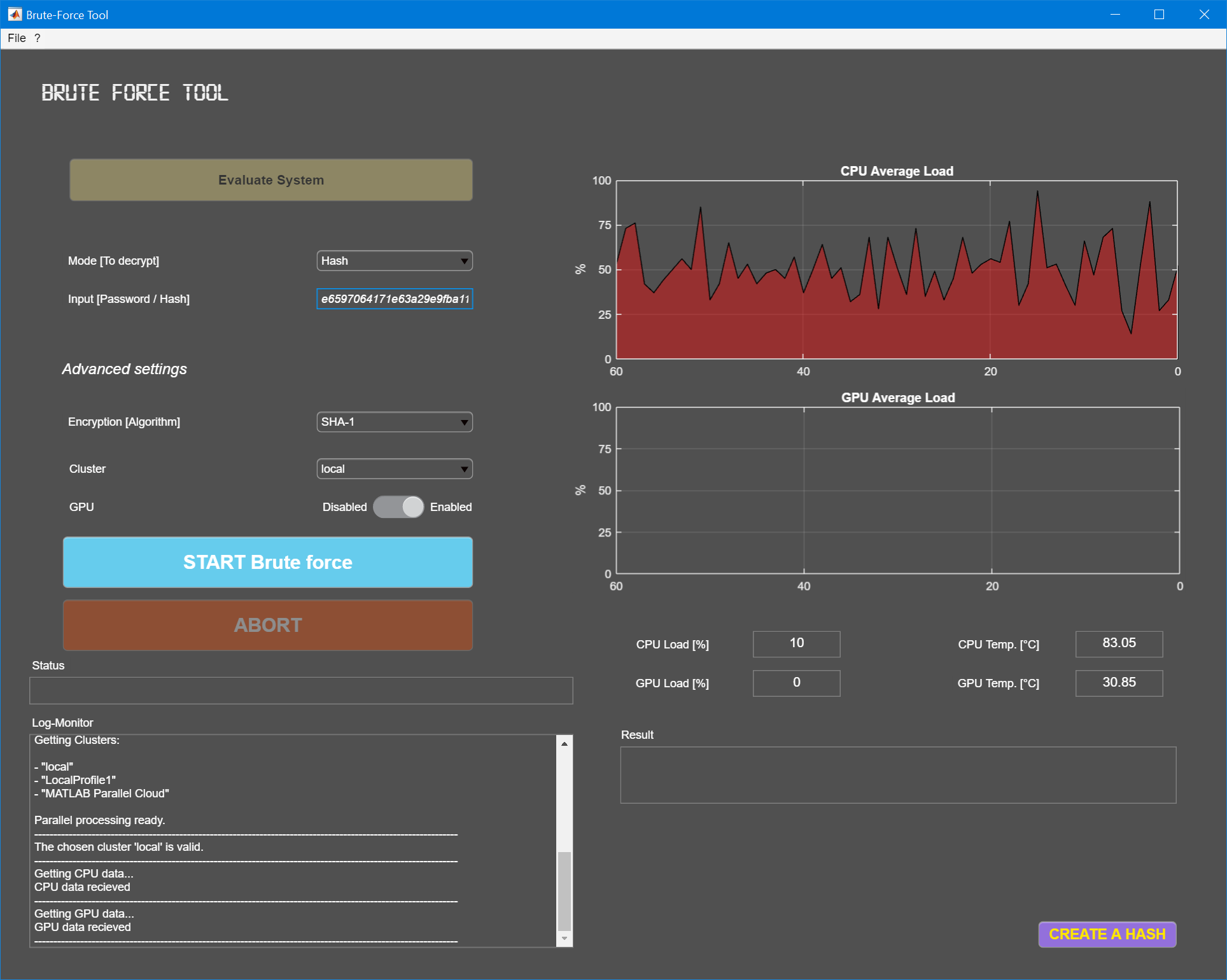
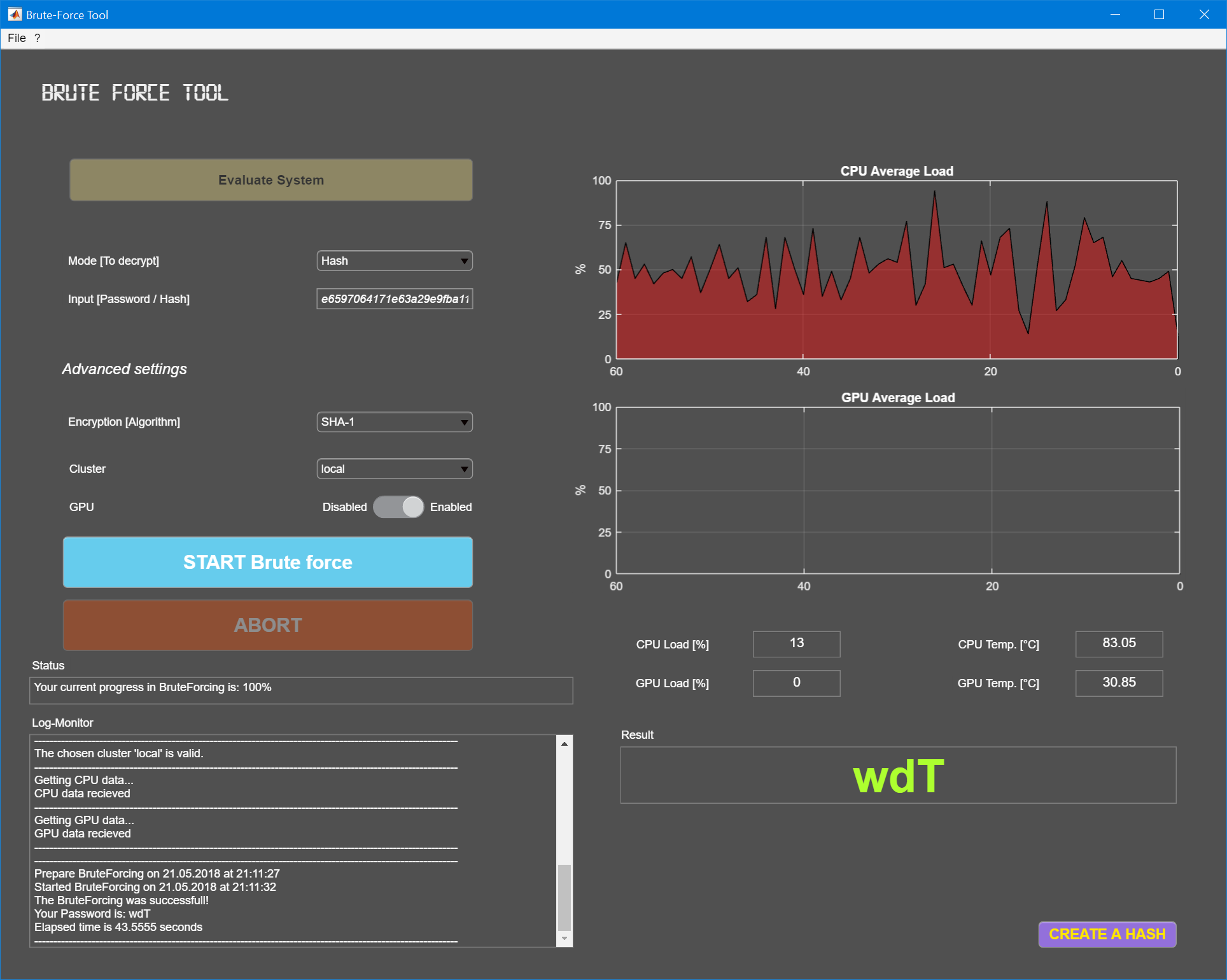


Abbildung 12 Browserfenster mit dem Hash Generator





# Projektmanagement

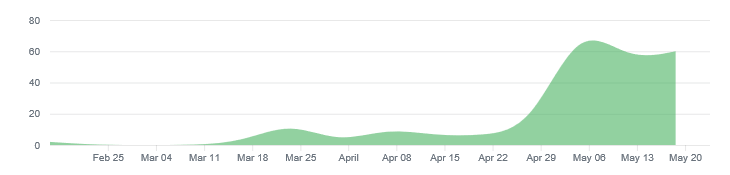
## Soll – Ist Vergleich des Lastenheftes

Die Vergleiche von Soll- und Ist-Zuständen sind nur vorgenommen worden, falls Diskrepanzen bestehen.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Soll | Ist | Begründung |
| **Kapitel 1.1** Nr. 7 | Visualisierung der Parameter CPU/GPU- Temperatur und Lüfterdrehzahl. | Die Temperaturen werden visualisiert. Die Lüfterdrehzahlen nicht. | Hardwaremässig nicht verfügbar. |
| **Kapitel 1.1** Nr. 8 | Fortschrittsanzeige (Progressbar) | Prozentuale Fortschrittanzeige. | Um den Fortschritt aussagekräftig zu visualisieren, wurde auf die «Progressbar» verzichtet und auf prozentuale Darstellung gesetzt. |
| **Kaptiel 3.1** | Verwendung der MATLAB®-Version R2015a | Verwendbarkeit ab MATLAB®-Version R2017b. | Der appdesigner verwendet Komponenten, die nicht Rückwärtskompatibel sind. |
| **/F80/** | Schreiben der Daten bei erfolgreichem Ausführen der Software (Passwort, benötigte Zeit, Anzahl Kombinationen | Schreiben der Daten bei erfolgreichem Ausführen der Software (Passwort, benötigte Zeit | Die Anzahl Kombinationen zu erhalten ist grundsätzlich möglich. Der Aufwand dafür wäre allerdings sehr hoch mit einer sehr geringen Aussagekraft. |
| **/D30/** | Logs mit Angaben zu Versuchsnummer, aktueller Zeit, Belastung der Ressourcen, versuchte Kombinationen werden auf dem UI ausgegeben. | Logs mit Angaben zur aktuellen Zeit werden auf dem UI ausgegeben. | Die Aussagekraft von einigen Angaben ist sehr gering und nicht sinnvoll. |
| **/L10/** | Die Software soll jene Passwörter, welche den korrekten Aufbau aus Kapitel 1.3 auf-weisen, knacken können | Die Software knackt Passwörter mit dem Aufbau aus Kapitel 1.3 aus dem Lastenheft, allerdings wurde die Zeichenlänge auf die Hälfte reduziert. | Massive Zeitersparnisse mit gewöhnlichen Rechenleistungen. |

## Commits

Der folgende Graph zeigt die Anzahl der Commits auf das Versionierungssystem, welche während der gesamten Projektdauer getätigt wurden.



## Aufwand pro Funktion

|  |  |  |
| --- | --- | --- |
| **Funktion** | **Stunden (A. Gonzalez)** | **Stunden (B. Hürzeler)** |
| **BruteForce** |  |  |
| createString.m | 6 | 0 |
| DataHash.m | 1.5 | 0 |
| doBruteForce.m | 25 | 0 |
| doBruteForceAscendingly.m | 2 | 0 |
| doBruteForceAscendinglyUIUpdate.m | 4 | 0 |
| doBruteForceRandomly.m | 2 | 0 |
| initBruteForce.m | 2 | 0 |
| **Get-Set-Data** |  |  |
| Cpuinfo.m | 1.5 | 1.5 |
| displayData.m | 0 | 6 |
| getCpuData.m | 0 | 12 |
| getGpuData.m | 0 | 12 |
| saveFile.m | 0.5 | 3 |
| **UI** |  |  |
| deleteApp.m | 1 | 0 |
| initApp.m | 2 | 0 |
| userInterface\_script.m | 30 | 36 |
| **Scripts** |  |  |
| get-cpu-load.ps1 | 0 | 0.5 |
| get-gpu-load.ps1 | 0 | 0.5 |
| get-gpu-temperature.ps1 | 0 | 0.5 |
| get-temperature-TZ0-ps1 | 0 | 0.5 |
| **src** |  |  |
| runApp.m | 2 | 0 |
| systemFileBF.exe | 0 | 2 |
| **Dokumentation** |  |  |
|  | 3 | 6 |
|  | 82.5h | 80.5h |

Tabelle 3 Zeitaufwand in Stunden

# Abnahmetests

## Abnahmetest der Module

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Entschlüsseln | UI ist erfolgreich evaluiert. | Eingabe Hash der Methode SHA-1:  f10e2821bbbea527ea02200352313bc059445190 | Ausgabe Passwort «asd» | 58sec | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Entschlüsseln | UI ist erfolgreich evaluiert. | Eingabe Hash der Methode SHA-256:  651de3316fa27e74d6a9de619ebac68e3b781e9527fdd00c5ef7143b1fa581b6 | Ausgabe Passwort «omg» | 65sec | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Entschlüsseln | UI ist erfolgreich evaluiert. | Eingabe Hash der Methode SHA-512:  85271d45fbb5112d49f9492e1ae55e196a32b1cd634a3  180d52e29ae7faf1e59c678538d6b6898f3b00a1db6e  834c49ba159ac45675df2ea53d6e7e0d03cb303 | Ausgabe Passwort «wtf» | 67sec | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Entschlüsseln | UI ist erfolgreich evaluiert. | Eingabe Hash der Methode MD5:  6dc13510ac2bcf8ac6c2226b2e16e51e | Ausgabe Passwort «0815» | 204sec | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Entschlüsseln | UI ist erfolgreich evaluiert. | Eingabe Passwort der Methode SHA-1:  «Ui1» | Ausgabe Passwort «Ui1» | 3.3sec | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Entschlüsseln | UI ist erfolgreich evaluiert. | Eingabe Passwort der Methode SHA-256:  «pN0» | Ausgabe Passwort «pN0» | 2.9sec | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Entschlüsseln | UI ist erfolgreich evaluiert. | Eingabe Passwort der Methode SHA-512:  «wS7» | Ausgabe Passwort «wS7» | 11sec | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Entschlüsseln | UI ist erfolgreich evaluiert. | Eingabe Passwort der Methode MD5:  «Md5» | Ausgabe Passwort «Md5» | 8.8sec | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Entschlüsseln | UI ist erfolgreich evaluiert. | Eingabe Passwort mit Sonderzeichen:  «W!a» | Keine Ausgabe möglich. Warnung bei Eingabefeld. |  | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Entschlüsseln | UI ist erfolgreich evaluiert. | Eingabe Passwort mit Sonderzeichen:  MaTlAb | Keine Ausgabe möglich. Warnung bei Eingabefeld. |  | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Korrektheit | UI ist geschlossen. | Öffnen der Applikation ohne Windows Admin-Rechte. | Warnung an den Benutzer. |  | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Korrektheit | UI ist gestartet und noch nicht evaluiert. | Zweite Instanz der Applikation erzeugen. | Errormeldung an den Benutzer und wurf einer Exception. |  | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Korrektheit | UI ist gestartet und noch nicht evaluiert. | Evaluierung der Software mit geöffnetem Parallel-Pool. | Eintrag in Log-Monitor und Schliessen des Parallel-Pools. |  | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Korrektheit | UI ist erfolgreich evaluiert. | Start des BruteForcing mittels dem Cluster “MATLAB® Distributed Cloud» | Eintrag in den Log-Monitor. BruteForcing nicht ausführbar. |  | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Korrektheit | UI ist erfolgreich evaluiert. | Start des BruteForcing mit geöffnetem Parallel-Pool. | Eintrag in Log-Monitor und Schliessen des Parallel-Pools. Start BruteForcing. |  | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Korrektheit | UI ist erfolgreich evaluiert. | Im Menü wird die Speicherfunktion aufgerufen. | Erstellung eines Files mit den Daten aus dem Log-Monitor. |  | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Korrektheit | UI ist erfolgreich evaluiert. | Im Menü wird die NewRun Funktion aufgerufen. | Rückstellung der Komponenten auf den Zustand vor der Evaluierung. |  | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Korrektheit | UI ist erfolgreich evaluiert. | Im Menü wird die NewRun Funktion aufgerufen **ohne** das der Log-Monitor im Vorhinein gespeichert wurde. | Erscheinen eines Nachrichtenfeldes mit Speichermöglichkeit. |  | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Korrektheit | UI ist erfolgreich evaluiert. | Im Menü wird die NewRun Funktion aufgerufen wobei das der Log-Monitor im Vorhinein gespeichert wurde. | Erscheinen eines Nachrichtenfeldes ohne Speichermöglichkeit. |  | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Korrektheit | UI ist erfolgreich evaluiert. | Im Menü wird die Exit Funktion aufgerufen **ohne** das der Log-Monitor im Vorhinein gespeichert wurde. | Erscheinen einer Warnung mit Hinweis auf Speicherung. |  | OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Funktion** | **Ausgangslage** | **Ereignis** | **Ergebnis Soll** | **Benötigte Zeit** | **Ergebnis Ist** |
| Korrektheit | UI ist erfolgreich evaluiert. | Im Menü wird die Exit Funktion aufgerufen wobei das der Log-Monitor im Vorhinein gespeichert wurde. | Erscheinen einer Warnung **ohne** Hinweis auf Speicherung. |  | OK |

# Ehrlichkeitserklärung

«Hiermit erklären wir, die vorliegende Dokumentation selbständig, ohne Hilfe Dritter und nur unter Benutzung der angegebenen Quellen verfasst zu haben»

Herr Adrian Gonzalez, 25. Mai 2018, Windisch

Herr Bruno Hürzeler, 25. Mai 2018, Windisch

# Anhang

## Klassendiagramm

