Thema der Aufgabe:

Deklaration einer Zeichenkette (String) und Provozieren eines Buffer-Overflows.
Umgang mit Zeichenketten!

Aufgabenbeschreibung:

Schreiben Sie ein Programm für die HAM in Assembler, dass eine Zeichenkette deklariert und mit Zeichen füllt. Es soll eine Subroutine erstellt werde, die diese Zeichenkette in einen anderen Speicherbereich kopiert, allerdings durch ein fehlendes Terminierungszeichen ‚\0' oder 0x00 in der Ursprungszeichenkette gestört wird.

- Deklaration der Zeichenkette **ohne** Terminierungszeichen!
- Überschreiben des Speichers nach dem letzten Zeichen der Zeichenkette provozieren

Frage 1: Warum benötigt man das sog. Terminierungszeichen?
Frage 2: Was geschieht beim Überschreiben des Speicherbereichs nach den String?

Info: Dieser Buffer-Overlow wird häufig bei den sog. Denial-of-Service-Angriffen provoziert, um Administrationsrechte des Systems zu erlangen oder das System lahmzulegen, das angegriffen wird. Häufig sind solche Angriffe möglich, weil Programmierer ihre Eingaben nicht gegen das Überschreiben von Speicherbereiche schützen. Das würde Mehraufwand bei der Entwicklung ohne nennenswerten Funktionalitätzuwachs bedeuten, also Aufwand ohne Kostenausgleich!

Zusatzaufgabe: Überdenken Sie ein möglichen Schutzmechanismus, um den Code vor Buffer-Overflow zu schützen und bewerten Sie den Mehraufwand dafür!

Tipp: Nutzen Sie die indirekte Addressierung mit `LDI/STI`.

Eingabe:

-

Ausgabe:

-

Beispiel:

- Ping of Death: DoS-Angriff auf Basis eines Buffer-Overflows