

Buffer Overflow

Schwachstellen und wie man sie Schließt

Jakob Stühn, John Meyerhoff, Sam Taheri

H-BRS

Inhaltsverzeichnis

1	Grundaufbau	1
2	Geschichte	2
2.1	Bekannte Buffer-overflows	2
2.2	Aktuelle Beispiele	2
3	Grundlegende Theorie	3
3.1	Aufbau des Stacks	3
3.2	Angriffsvorgang	3
3.3	Verhalten bei Overflow	3
4	Shellcode	4
4.1	Code	4
4.2	Erläuterung	4
5	Anwendungsfallbeispiel	5
5.1	Code	5
5.2	Setup des Servers	5
5.3	Böswilliger Client	5
5.4	Erläuterung des Vorgangs	5
6	Gegenmaßnahmen	6
6.1	Struktur	6
6.2	Code-Beispiel	6
6.3	Testen	6
7	Quellen	7

1 Grundaufbau

- Eingabemöglichkeit
- Speichern der Eingabe
- Ablegen von Anweisungen durch übergroße Eingabe
- Ausführen der Anweisungen → Remote Code Execution

2 Geschichte

2.1 Bekannte Buffer-overflows

2.2 Aktuelle Beispiele

3 Grundlegende Theorie

3.1 Aufbau des Stacks

3.2 Angriffsvorgang

3.3 Verhalten bei Overflow

4 Shellcode

4.1 Code

4.2 Erläuterung

5 Anwendungsfallbeispiel

5.1 Code

5.2 Setup des Servers

5.3 Böswilliger Client

5.4 Erläuterung des Vorgangs

6 Gegenmaßnahmen

6.1 Struktur

- Stack-Schutz mit “Canary” (Zufallszahl)
- Safe Pointer Instrumentalisierung
- C Range Error Detector und Out Of Bounds Object
- Hardware-basierte Lösungen
- Statische Code-Analyse
- Betriebssystembasierte Ansätze
- Manuelles Buffer-Overflow Blocken (Input-Bereinigung)

6.2 Code-Beispiel

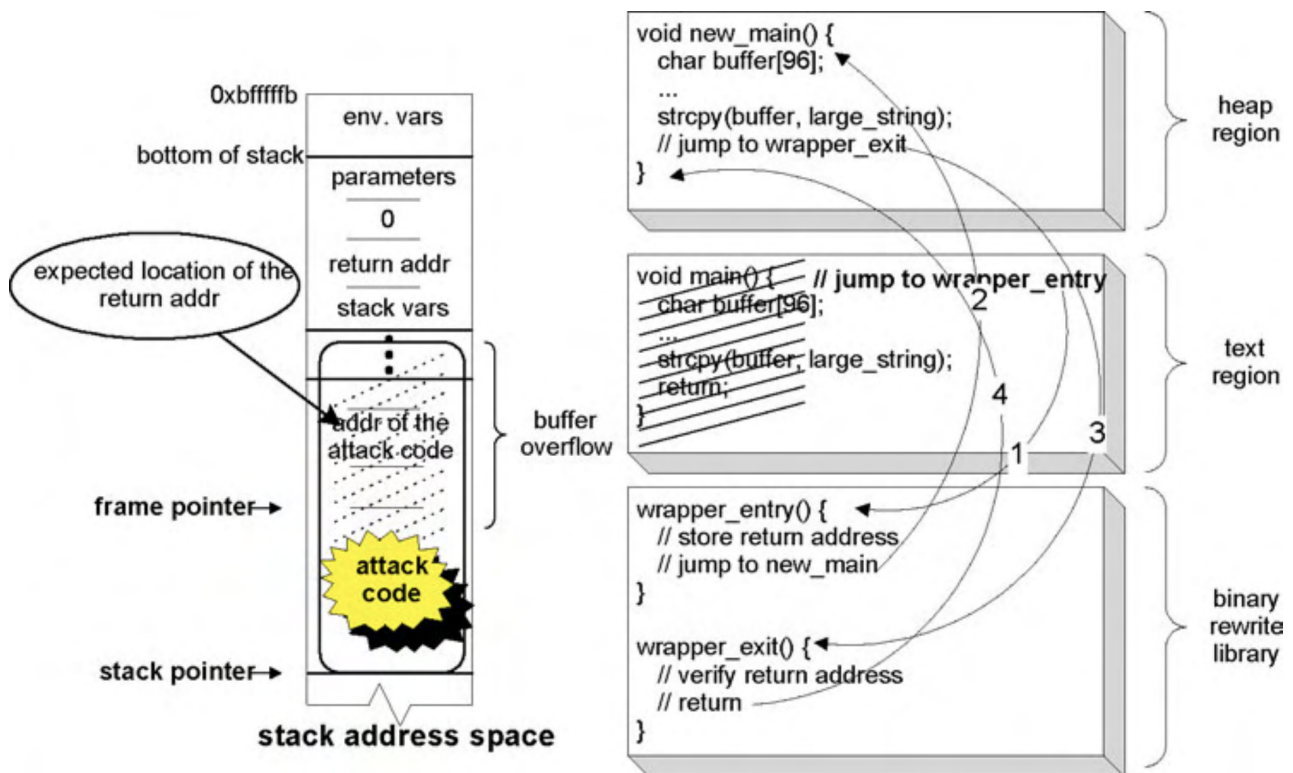


Figure 9: Libverify function call and stack layout

6.3 Testen

- Fuzzy Tests
- Spezifische Payloads

7 Quellen

- <https://www.nds.ruhr-uni-bochum.de/media/nds/attachments/files/2010/11/Survey.on.Buffer.Overflow>