演習 3 - レポート課題

森田 蓮

2025 年 1 月 28 日

1　目的

本演習では，バッファオーバーフロー (Buffer Overflow) 脆弱性を悪用した攻撃手法と，その対策について学ぶ．この脆弱性は，プログラムが入力データのサイズを適切に検証せずにメモリに格納することで発生する．本レポートでは，脆弱性の原因，攻撃手法，及びその防止策について学んだことを説明する．

2　学んだことを

バッファオーバーフローとは，固定サイズのバッファ（配列など）に対して，許容サイズを超えるデータを書き込むことにより，メモリの隣接領域が上書きされる問題を指す．gets() 関数は入力データの長さを制限しないため，64 バイトを超えるデータを入力すると，スタックに保存されている他のデータ（例えばリターンアドレス）が上書きされる可能性がある．

3　攻撃手法

演習では，以下の手順を通じて攻撃を実行した．

1. ソースコードの解析: ソースコードを確認し，脆弱性のある箇所（gets() 関数の使用）を特定できた．また，未使用の shell() 関数が含まれており，これを悪用してシェルを起動することが可能であると判断した．
2. アドレスの特定: 実行ファイルを逆アセンブルして，shell() 関数とリターンアドレスの位置を特定する．
3. 攻撃コードの作成:Pwntools を使用して，攻撃コードを作成する．

4　防御手法

gets() 関数の代わりに fgets() 関数を使用し，入力データの長さを制限する．この操作により，バッファサイズ以上の文字列を読み込まないようにできた．

5　感想

今回の演習は，とても難しくこのレポートを書いていて調べている中で学べることが多い演習だったと感じた．座学だけではイメージができないことが実際に動作するところを見ることで危険性が理解できた．自分で作る任意のコードを実行できることはとても怖いことだとこの 3 回の演習を行って感じた．危険な関数があることを知ってなぜ危険なのかを理解できた．これからのたくさんのコーディングを行う際に危険な関数を使ってユーザに不利益を被らないような技術者になりたいと思う．