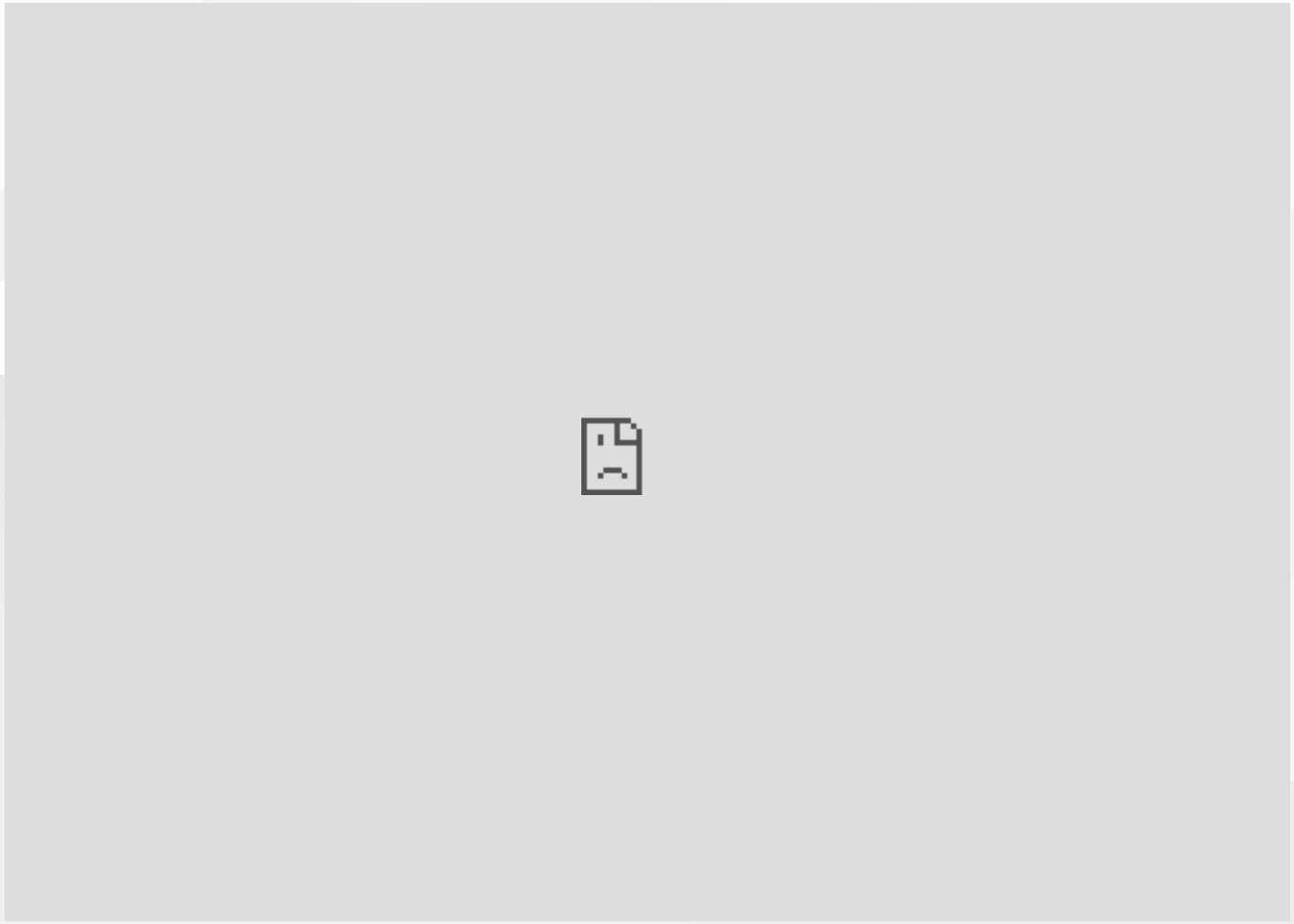
CEN429 Secure Programming

Week-13

Tigress and Diversification Techniques

Download

- PDF
- DOC
- SLIDE
- PPTX



Outline

- Tigress and Diversification Techniques
- Obfuscation Methods
- Defense Against Attacks

Week-13: Tigress and Diversification Techniques

This week, we will examine diversification techniques that make code analysis more difficult and make programs more resistant to attacks, as well as obfuscation tools like Tigress. These techniques ensure that each time the program runs, it behaves differently, making it harder for attackers to analyze the program with the same methods.

This transforms the "fib" function into a switch-based virtual machine.

2. Implementing Diversification in Code

   Theoretical Explanation: Diversification involves randomizing code in different ways to make it harder to analyze. This method makes it difficult for an attacker to reverse-engineer the program. With Tigress, a program can create a unique virtual machine every time it runs.

Techniques for Implementing Code Diversification:

Flattening: Mixing the control flow of the program by placing everything in a loop.

Function Merging: Combining multiple functions into one.

Random Numbers for Diversification: Using random numbers to obfuscate functions and operands.

Instruction Dispatch Types:

Switch-based Dispatching: Using a switch-case to dispatch virtual machine instructions.

Indirect Dispatching: Redirecting branching points indirectly.

Example:

- **Encryption:** Encrypting parts of the code and decrypting them at runtime.
- **Moving Code Around:** Shuffling the code to different locations each time it runs.

- **Granularity Levels:**
  - **File Level:** Encrypting or moving the entire file.
  - **Function Level:** Dynamically changing specific functions.
  - **Basic Block Level:** Shuffling the basic blocks of the program.

Example:

```
void makeCodeWritable(caddr_t first, caddr_t last) {
  // Modify code before execution.
}
```

5. Self-Modifying Code

Theoretical Explanation: Self-modifying code allows a program to modify itself during execution. This method is used to make it harder for attackers to analyze the code

Conclusion

This week, we learned about advanced code obfuscation techniques like diversification and self-modifying code. These techniques make programs more resistant to attacks and harder for attackers to reverse-engineer. Tools like Tigress randomize code, creating a different structure every time, making code analysis and reverse-engineering much more difficult.

$$End - Of - Week - 13$$