

# Контекст как основа модификаторов доступа

shamcode

Аннотация Использование понятия "Контекста"упрощает описание модели модификаторов доступа, в частности *private*, *protected*, а также *friendly methods*

## 1 Модель модификаторов доступа

Введем несколько определений

Определение 1.  $F(C)$  - множество методов класса  $C$

Определение 2.  $a : F(C) \rightarrow \{public, protected, private\}$  - модификатор методов

Определение 3.  $i : A \rightarrow B$  - класс  $A$  наследуется от  $B$

Определение 4.  $[m_1, \dots, m_n]$  - стек вызовов, состоящий из последовательного вызова методов  $m_1, \dots, m_n$  таких, что  $m_i$  был вызван из  $m_{i-1}$ .

Определение 5. Стек вызова  $[m_1, \dots, m_n]$  называется корректным если для него выполняются условия:

$$m_1 \in F(C) \implies a(m_1) = public \quad (1)$$

$$\begin{cases} m_{j+1} \in F(C) \\ a(m_{j+1}) = private \end{cases} \implies m_j \in F(C) \quad (2)$$

$$\begin{cases} m_{j+1} \in F(i(C)) \\ a(m_{j+1}) = protected \end{cases} \implies \begin{cases} m_j \in F(C) \\ m_j \in F(i(C)) \end{cases} \quad (3)$$

Определение 6. Стек вызовов, для которого хотя бы одно из условий 1, 2, 3 не выполнено будем называть некорректным или стеком вызова с ошибками доступа.

## 2 Свойства корректного стека вызовов

Теорема 1. Для корректного стека  $[\dots, m_j, \dots]$  верно:

$$\begin{cases} m_j \in F(C) \\ a(m_j) = private \end{cases} \implies \exists m_k : \begin{cases} k < j \\ m_k \in F(C) \\ a(m_k) \neq private \end{cases} \quad (4)$$

Доказательство. По свойству 2 определения корректного стека 5, получаем, что  $m_{j-1} \in F(C)$ . Если  $a(m_{j-1}) \neq private$ , то мы получаем искомый  $m_k$ , т.к  $j-1 < j$ .

Если же  $a(m_{j-1}) = private$ , то применяя свойства 2, получаем, что  $m_{j-2}$  либо является искомым, либо  $a(m_{j-2}) = private$ .

Последовательно применяя свойства 2 мы либо найдем искомый метод, либо дойдем до  $m_1 \in F(C)$ . Но по свойству 1 получаем  $a(m_1) = public$ . Что и требовалось доказать. ■