

Защищенные мультисервисные телекоммуникационные системы

№ 1 Модель Харрисона-Руззо-Ульмана

Задача 1. Построить матрицы и записать в виде команд сценарий атаки с помощью троянской программы в системах, функционирующих на основе модели Харрисона-Руззо-Ульмана (ХРУ)

Дано: Пусть имеется два субъекта: s1 - доверенный пользователь, admin и s2 - рядовой пользователь, user; и два каталога o1 и o2, владельцами которых являются пользователи s1 и s2. В каталоге o1 имеется объект o3 с секретной информацией. Исходная матрица доступа имеет вид:

S	o1	o2	o3
s1	own, r, w, e	r, w, e	own, r, w, e
s2		own, r, w, e	

Решение:

Атакующий s2 должен создать в своем каталоге o2 файл с трояном и дать права rwe пользователю s1 на этот файл. Далее ожидает запуска доверенным пользователем s1 (или исп. команду sleep()) трояна из каталога o2. Главная цель убедить пользователя s1 запустить файл. Когда пользователь s1 запускает файл, троян скопирует секрет из o3 в o2, затем делегирует права доступа для атакующего s2, который теперь может работать с тем же уровнем доступа, что и s1. В итоге файл становится доступным для чтения s2, и атакующий может его прочитать из каталога o2.

```
command "создать файл" (s2, троян):  
  if "write" принадлежит [s2, o2] then  
    Создать объект троян;  
    Ввести {"own", "read", "write", "execute"}  
    в [s2, троян];  
  end if  
  if {"read", "write"} подмножество [s1, o2] then  
    Ввести {"read", "write", "execute"} в [s1, троян];  
  end if  
end command
```

S	o1 (секрет)	o2(секрета нет)	o3(секрет)	o4 (троян)
s1	own, r, w, e	r, w, e	own, r, w, e	r, w, e
s2		own, r, w, e		own, r, w, e

```

command "запустить файл" (s1,троян):
  if {"read","write","execute"} подмножество [s1,троян] then
    Создать субъект s(троян);
    Ввести { "read","write","execute"} в [s(троян),o2];
    Ввести { "read","write","execute"} в [s(троян),троян];
  end if
  if {"own", "read","write","execute"} подмножество [s1,o1] and {"own",
    ↪ "read","write","execute"} подмножество [s1,o3] then
    Ввести {"read","write","execute"} в [s(троян),o1];
    Ввести {"read","write","execute"} в [s(троян),o3];
  end if
end command

```

S	o1 (секрет)	o2(секрета нет)	o3(секрет)	o4 (троян)
s1	own, r, w, e	r, w, e	own, r, w, e	r, w, e
s2		own, r, w, e		own, r, w, e
s(троян)	r, w, e	r, w, e	r, w, e	r, w, e

```

command "скопировать файл o3 программой s(троян) в o2" (s(троян),o3,o2):
  if "read" принадлежит [стр,o3] and "write" принадлежит [s(троян),o2] then
    Создать объект o';
    Ввести {"own", "read", "write", "execute"} в [s(троян),o'];
    Ввести "read" в [s2,o'];
    Читать (s(троян),o3);
    Записать (s(троян), o');
  end if
  Уничтожить субъект s(троян);
end command

```

S	o1 (секрет)	o2(секрета нет)	o3(секрет)	o4 (троян)	o'= o3(секрет)
s1	own, r, w, e	r, w, e	own, r, w, e	r, w, e	
s2		own, r, w, e		own, r, w, e	r

Задача 2. Построить сценарий аналогичной атаки. Отобразить последовательности команд перехода и изменений матрицы доступа.

Дано: Доверенный пользователь s1 в исходном состоянии имеет на каталог o2 только права чтения r. Исходная матрица доступа имеет вид:

S	o1 (секрет)	o2(секрета нет)	o3(секрет)
s1	own, r, w, e	r	own, r, w, e
s2		own, r, w, e	

Решение:

s2, являясь владельцем o2, дает на него недостающие права s1

```
command "дать права на каталог от владельца" (s1,s2,o2):
  if "own" принадлежит [s2,o2] then
    Ввести {"write","execute"} в [s1,o2];
  end if
end command
```

S	o1 (секрет)	o2(секрета нет)	o3(секрет)
s1	own, r, w, e	r, w, e	own, r, w, e
s2		own, r, w, e	

№ 3 Модель с типизованной матрицей доступа

Задача 1. Построить в соответствии с командой A граф отношений наследственности.

Дано: Система настроена в соответствии с типизованной моделью доступа. Пусть имеется три типа сущностей: a, b и c. Начальное состояние системы S0: (s1: A) - субъект s1 типа a. Система переходит в новое состояние S0 aS1 при которой создается объект o1 типа c, инициализируются новые субъекты s2 типа b и s3 типа A. Вновь созданным субъектам предоставляются права доступа r' и r'' на объект o1. Переход осуществляется по следующей команде:

```
A(s1: A; s2: b; o1: ∅)
  Create object o1 of type c;
  Inter r into [s1, o1];
  Create object s2 of type b;
  Inter r' into [s2, o1];
  Create object s3 of type a;
  Inter r'' into [s3, o1];
end A
```

Решение:

Граф отношений наследственности a, b, c

Задача 2. Реализовать сценарий атаки с помощью троянской программы при условии функционирования по типизованной модели доступа

Дано: Пусть имеется два субъекта доступа: (s1: a) – субъект s1 типа a - доверенный пользователь admin (s2: b) - субъект s2 типа b – обычный пользователь user. Три объекта доступа: - Каталог (o1: v)sec – принадлежит пользователю (s1:a) “own” ∅ rs1,o1; - Несекретный каталог (o2: ∅)non sec – принадлежит пользователю (s2:b) “own” ∅ r's2,o2; - Секретный файл (o3: v)sec в каталоге (o1: v)sec – принадлежит пользователю (s1:a) “own” ∅ r''s1,o3;

В исходном состоянии граф наследственности имеет четыре вершины: a, b, v, n.

Построить граф отношений наследственности по сценарию атаки троянским конем со стороны пользователя s2 на секретный файл o3 и записать команды перехода из состояния в состояние в нотации модели ТМД.

Решение:

Граф отношений наследственности a, b, v, n

№ 4 Модель Take-Grant

Задача 1. Построить систему команд перехода передачи субъекту x прав доступа A на объект s от субъекта y

Дано:

Решение:

Объект S_2 наследует у объекта Y право доступа к объекту S (у S_2 есть на это права — t (take)). Затем объект S_2 предоставляет объекту s_1 свое право на объект S (у объекта S_2 есть права $grant$ для делегирования прав). Далее объект X берет право a на объект S у объекта S_1 , так как S_1 имеет доступ к S , а X имеет право t (take) прав у объекта S_1 .

Итоговая цепочка атаки: $TAKE(S_2 Y \Rightarrow S) GRANT(S_2 S_1 \Rightarrow S) TAKE(X S_1 \Rightarrow S)$.

Задача 2. Построить систему команд перехода передачи субъекту x прав доступа A на объект s от субъекта y и оценить возможность передачи прав доступа по tg -пути независимо от направления прав t и g

Дано:

Решение:

Аналогично задаче 1, объект S_2 берет право у Y на доступ к объекту S . Затем S_1 создает новый объект $ABOVA$ со следующими правами: $TAKE + GRANT$, и дает права $GRANT$ для объекта S_2 , чтобы тот мог работать с объектом $ABOVA$. Объект S_2 , имея права на объект S и права $GRANT$ для $ABOVA$, предоставляет объекту $ABOVA$ права на объект S . S_1 , будучи владельцем $ABOVA$, наследует права на S у объекта $ABOVA$.

Итог: S_1 получил доступ к S независимо от направления прав.

Задача 3. Построить систему команд получения субъектом s прав доступа A на объект w от субъекта u при условии того, что команда $grants(A,u,s,w)$ не может быть задействована

Дано:

Решение:

Пусть объект NEW имеет права на объект U, и объект U также имеет права на объект NEW.

Тогда:

1. Объект U предоставляет права на S объекту NEW.
2. Объект S берет у объекта NEW право на объект U.
3. Объект S берет у U права на W.

Итог: Объект S имеет права на W.

№ 4 Модель

Задача 1.

Дано:

Решение: