Полнофункциональная реализация мандатного контроля целостности в операционной системе Astra Linux Special Edition

Девянин Петр Николаевич
Главный научный сотрудник
АО «НПО РусБИТех»

Соснин Юрий Владимирович
Директор по развитию
АО «НПО РусБИТех»

Оружейников Александр Львович
Начальник НТЦ-1
ООО «РусБИТех-Астра»

РусБИТех-Астра
ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ

ИСТОРИЯ. МОДЕЛЬ БИБА - 1975

РУСБИТЕХ

Разрабатывалась параллельно с моделью Белла-ЛаПадула для ОС MULTICS под эгидой корпорации MITRE

ABSTRACT

An integrity policy defines formal access constraints which, if effectively enforced, protect data from improper modification. We identify the integrity problems posed by a secure military computer utility. Integrity policies addressing these problems are developed and their effectiveness evaluated. A prototype secure computer utility, Multics, is then used as a testbed for the application of the developed access controls.

МОДЕЛЬ БИБА - ПОЛИТИКА КОЛЬЦА - МКЦ

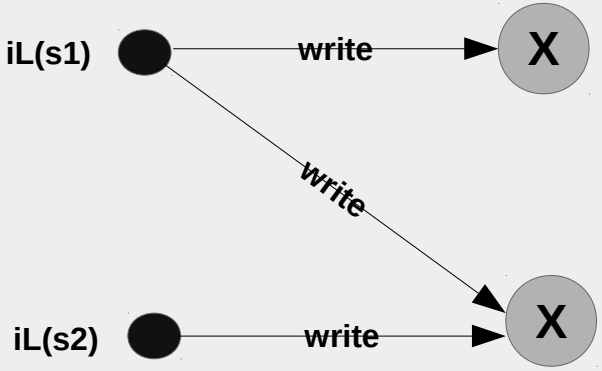
РУСБИТЕХ

The Ring Policy

The policy is defined by two axioms.

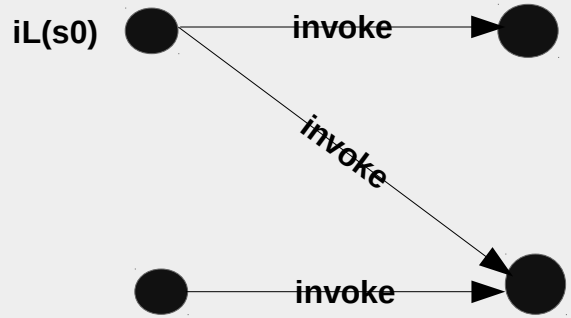
(A3.4) ∀o ∈ O, s ∈ S s m o => il(o) leq il(s).

(A3.5) ∀ s[1], s[2] ∈ S s[1] i s[2] => il(s[2]) leq il(s[1]).



iL(o1) = High

iL(o2)= Low



iL(s1) = High

iL(s2)= Low

ДОСТУП НА READ РАЗРЕШЕН

РУСБИТЕХ

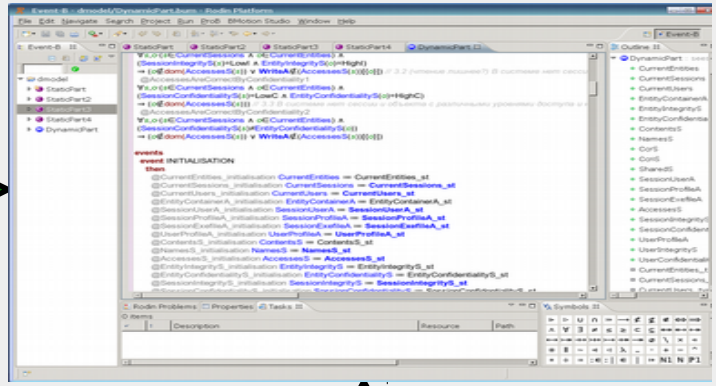
ИСП РАН

Автоматизированная дедуктивная верификация
МРОСЛ ДП-модели в формализованной нотации

Математическая нотация МРОСЛ ДП-модели

access_read(x, x', y)

$x, x' \in S, y \in E \cup R \cup AR$, существует $r \in R \cup AR$: $(x, r, read_a) \in AA$, [если $y \in E$, то $(y, read_r) \in PA(r)$ и либо $(execute\_container(x, y) = true$ и $f_e(y) \le f_s(x))$, либо $(x, downgrade\_admin\_role, read_a) \in AA$], [если $y \in R \cup AR$, то $(y, read_r) \in APA(r)$, $i_r(y) \le i_s(x)$, $Constraint_{AA}(AA') = true$, (для $e \in]y[$ либо $(x, e, read_a) \in A$, либо $(x, e, write_a) \in A$), (либо $f_r(y) \le f_s(x)$, либо $(x, downgrade\_admin\_role, read_a) \in AA)$], [если $y \in R \cup AR$ и $i_r(y) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]	$S' = S$, $E' = E$, $APA' = APA$, $PA' = PA$, $user' = user$, $H_E' = H_E$, $F' = F$, если $y \in E$, то $[A' = A \cup \{(x, y, read_a)\}$, $AA' = AA]$, если $y \in R \cup AR$, то $[AA' = AA \cup \{(x, y, read_a)\}$, $A' = A]$



Rodin (Event-B)

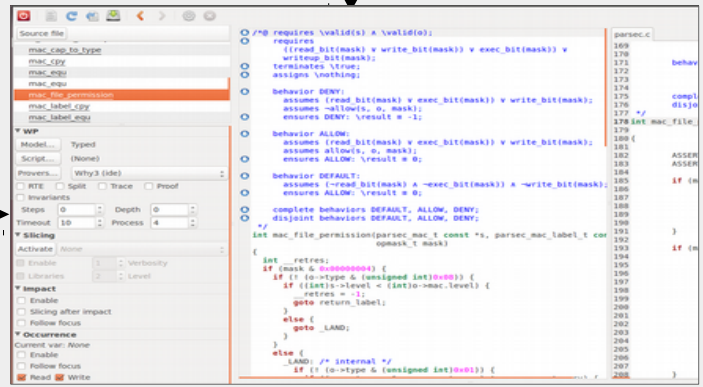
Требования ИТ.ОС.А2.ПЗ
ADV_SPM.1, ADV_FSP.6, AVA_CCA_EXT.1

```
static int access_read(struct task_struct *subject,struct inode *entity)
{
    struct task_security *subject_security;
    struct inode_security *entity_security;
    int ret = -EACCES;

    subject_security = get_task_security(subject);
    entity_security = get_entity_security(entity);

    if(!subject_security || !entity_security)
        return ret;

    if(is_role(entity)) //проверяем возможность получения доступа на чтение к сущности
    {
        ret = can_access(&subject_security->roles, &entity_security->list, MAY_READ, 0);
        if(ret != 0)
            ret = can_access(&subject_security->admin_roles, &entity_security->list, MAY_WRITE, 0);
        if(ret == 0)
        {
            ret = execute_container(subject, entity);
            if(ret != 0)
                ret = is_downgrade_admin_role(&subject_security->admin_roles, MAY_READ);
        }
    }
    else //проверяем возможность получения доступа на чтение к роли или административной роли
    {
        ret = can_admin_access(&subject_security->admin_roles, &entity_security->list, MAY_READ);
    }
    return ret;
```



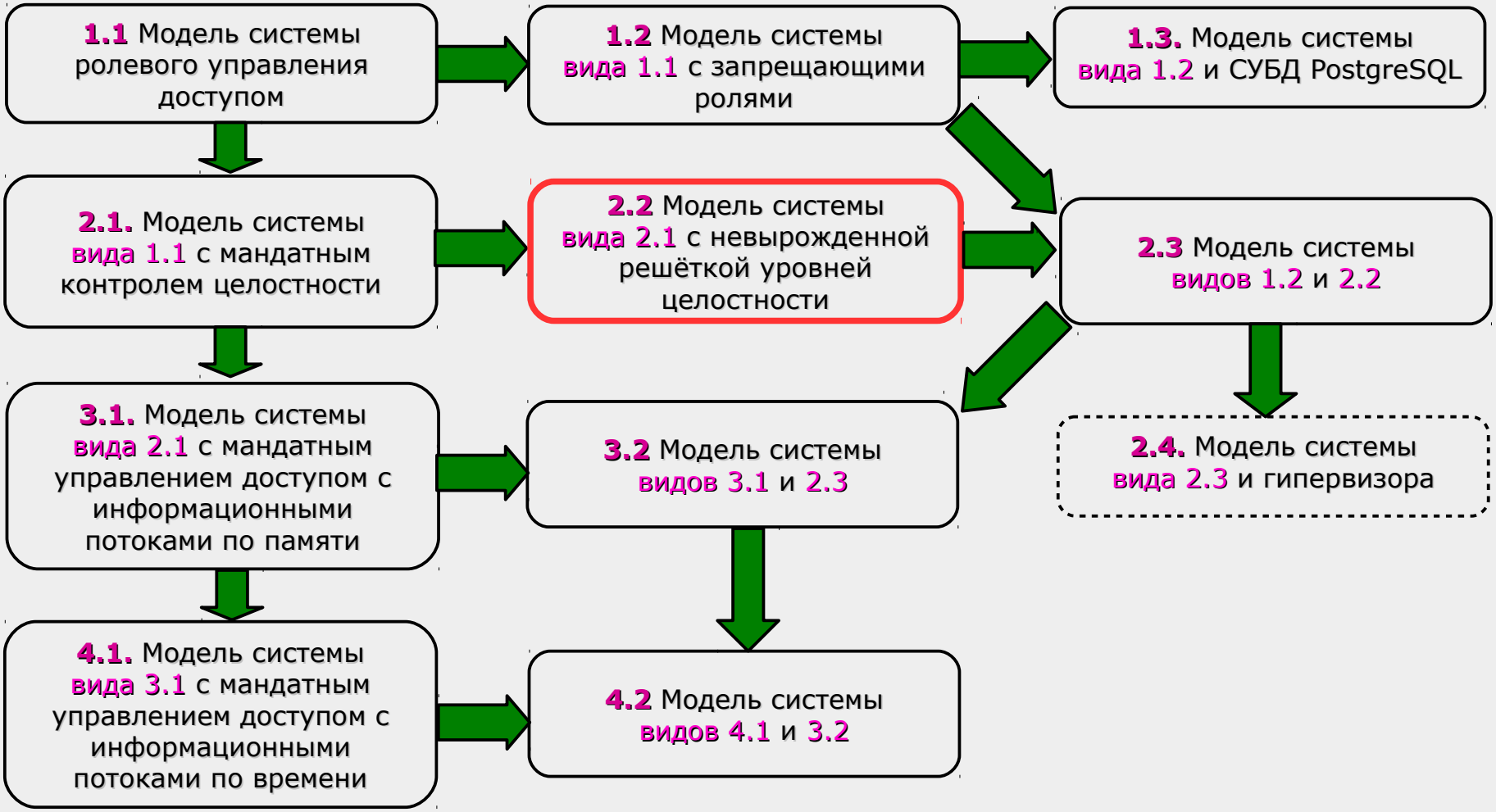
Frama-C (Why3)

Программный код Astra Linux Special Edition

Верификации кода на соответствие спецификациям

ИЕРАРХИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ МРОСЛ ДП-МОДЕЛИ

РУСБИТЕХ



create_user(x, x', u, uc, ui, ue)

$(LI, \leq)$ — решётка уровней целостности данных, где $i\_low, i\_high \in LI$;

$L_U = \{u \in U: i_u(u) = i\_high\}$;

$N_U = \{u \in U: i_u(u) < i\_high\}$ — множество учётных записей недоверенных пользователей

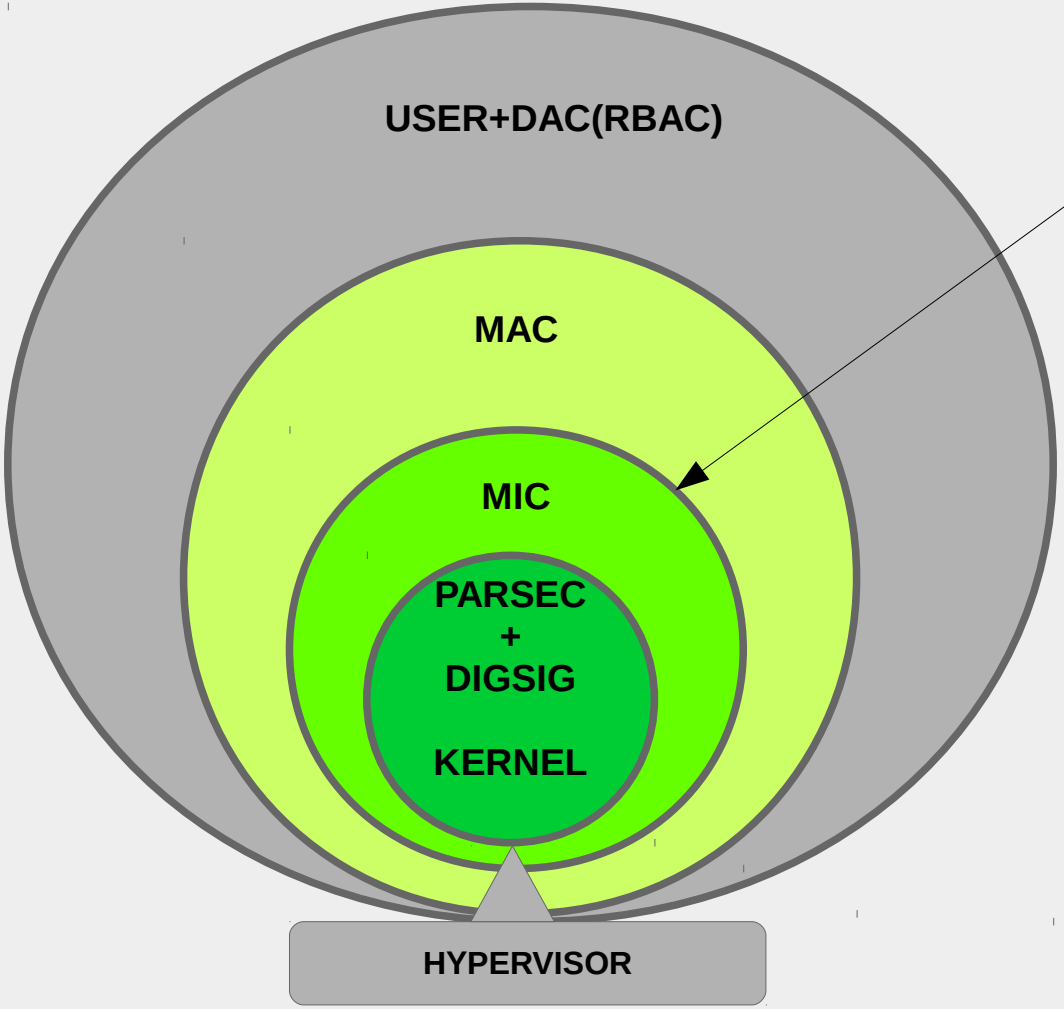
Уровень	Параметры	Исходное состояние G	Результирующее состояние G'
1.1	x, u	$x \in S, u \notin U,$ $(x, users\_admin\_role, read_a) \in AA,$ $(x, roles\_admin\_role, \alpha_a) \in AA,$ $(x, admin\_roles\_admin\_role, \alpha_a) \in AA,$ где $\alpha_a \in \{read_a, write_a\}$	$U' = U \cup \{u\}$
2.1	x', ui, ue	$x' \in S, ue \subset UE, ui \leq i_s(x),$ [для $e \in ue$ выполняется $i_e(e) = ui$]	$]u[= ue, i_u'(u) = ui$, если $ui = i\_high$, то $L_U' = L_U \cup \{u\}$, иначе $N_U' = N_U \cup \{u\}$, для $u' \in U$ верно $i_u'(u') = i_u(u')$, $AR' = AR \cup \{u\_admin\_li: li \leq ui\}, i_r'(u\_admin\_li) = li, PA'(u\_admin\_li) = \varnothing,$ $direct'(u\_admin\_li) = shared\_container'(u\_admin\_li) = true,$ $CCRI'(u\_admin\_li) = false,]u\_admin\_li[= \varnothing, role\_name'(u\_admin\_li) =$ "u_admin_li", где $li \leq ui$, $R' = R \cup \{u\_c\_li: li \leq ui\}, i_r'(u\_c\_li) = li, PA'(u\_c\_li) = \varnothing, direct'(u\_c\_li) = true,$ $shared\_container'(u\_c\_li) = true, CCRI'(u\_c\_li) = false,]u\_c\_li[= \varnothing,$ $role\_name'(u\_c\_li) =$ "u_c_li", где $li \leq ui$, $APA'(admin\_roles\_admin\_role) = APA(admin\_roles\_admin\_role) \cup$ $\{(u\_admin\_li, own_r): li \leq ui\},$ $APA'(roles\_admin\_role) = APA(roles\_admin\_role) \cup \{(u\_c\_li, own_r): li \leq ui\},$ для $ar \in AR$ выполняется $APA'(ar) = APA(ar) \cup \{(u\_admin\_li, execute_r): li \leq ui\} \cup$ $\{(u\_c\_li, execute_r): li \leq ui\}$
2.2	–	Если $ui > i\_low$, то $(x', i\_entity, write_a) \in A$	$H_R'(u\_admin\_li) = \{u\_admin\_li': li' < li$ и не существует $li'' \in LI: li' < li'' < li\},$ $H_R'(u\_c\_li) = \{u\_c\_li': li' < li$ и не существует $li'' \in LI: li' < li'' < li\},$ $APA'(u\_admin\_li) = \{(u\_admin\_li', \alpha_r): li' \leq li$ и $\alpha_r \in \{read_r, write_r, execute_r\}\} \cup$ $\{(u\_c\_li, \alpha_r), (common\_li, \alpha_r): \alpha_r \in \{read_r, write_r, execute_r\}\} \cup$ $\{(r, execute_r): r \in R' \cup AR'\},$ где $li \leq ui$

Определение о.Ц.07.ЦР. Пусть G_0 — безопасное начальное состояние системы $\Sigma(G^*, OP, G_0)$, и существует траектория без кооперации доверенных и недоверенных субъект-сессий $G_0 \vdash_{op1} G_1 \vdash_{op2} \dots \vdash_{opN} G_N$, где $N \geq 1$. Будем говорить, что в состоянии G_N произошло нарушение безопасности системы в смысле мандатного контроля целостности, когда существуют недоверенная субъект-сессия $x \in N_{SN}$ и субъект-сессия $y \in de\_facto\_own_N(x)$ такие, что не верно неравенство $i_s(y) \leq i_s(x)$, и это условие не выполняется в состояниях G_i траектории, где $0 \leq i < N$. Назовём систему $\Sigma(G^*, OP, G_0)$ безопасной в смысле мандатного контроля целостности, когда в ней невозможно соответствующее нарушение безопасности.

Теорема т.Ц.01.ЦР. Пусть G_0 — безопасное состояние системы $\Sigma(G^*, OP, G_0)$. Пусть на всех траекториях системы без кооперации доверенных или недоверенных субъект-сессий $G_0 \vdash_{op1} G_1 \vdash_{op2} \dots \vdash_{opN} G_N$, где $N \geq 1$, и в каждом состоянии G_N для каждой субъект-сессии $s \in S_N$ и сущности $e \in E_N$ выполняются следующие условия.
Условие Ц.1 (корректность уровней целостности сущностей, функционально ассоциированных с субъект-сессиями – без дополнений).
Условие Ц.2 (корректность уровней целостности, а также прав доступа на чтение к сущностям, параметрически ассоциированным с субъект-сессиями – без дополнений).
Условие Ц.3.ЦР (функциональная и параметрическая корректность всех доверенных субъект-сессий относительно всех доверенных субъект-сессий и сущностей). Для всех субъект-сессий $s \in S_N$ таких,
что $i\_low < i_{sN}(s)$, выполняются условия $\{s' \in S_N \mid i\_low < i_{sN}(s') \leq i_{sN}(s)\} \times (E_N \cup S_N) \subset f\_correct_N(s)$,
$\{s' \in S_N \mid i\_low < i_{sN}(s') \leq i_{sN}(s)\} \times (E_N \cup S_N) \subset p\_correct_N(s)$.
Тогда на этих траекториях система $\Sigma(G^*, OP, G_0)$ безопасна в смысле мандатного контроля целостности.

КОЛЬЦА ЗАЩИТЫ ИНФОРМАЦИИ В ASTRA LINUX

РУСБИТЕХ



НАДЕЖНОСТЬ ФУНКЦИОНИРОВАНИЯ СЗИ

ПРОСТЫЕ ПРАВИЛА КОНТРОЛЯ

⇩

НАДЕЖНАЯ РЕАЛИЗАЦИЯ

⇩

ВЫСОКАЯ СТЕПЕНЬ ДОВЕРИЯ

ИНИЦИАЛИЗАЦИЯ СЕРВИСОВ ASTRA LINUX НА УРОВНЯХ ЦЕЛОСТНОСТИ

РУСБИТЕХ

СДЗ

Linux Kernel + Parsec Module

здесь iL(s) = max

здесь iL(s) = mid

здесь iL(s) = 0

РЕЗУЛЬТАТ МКЦ — ВОЗМОЖНОСТЬ ОГРАНИЧИТЬ ФУНКЦИЙ СУПЕРПОЛЬЗОВАТЕЛЯ

РУСБИТЕХ

ROOT iL=max

ROOT

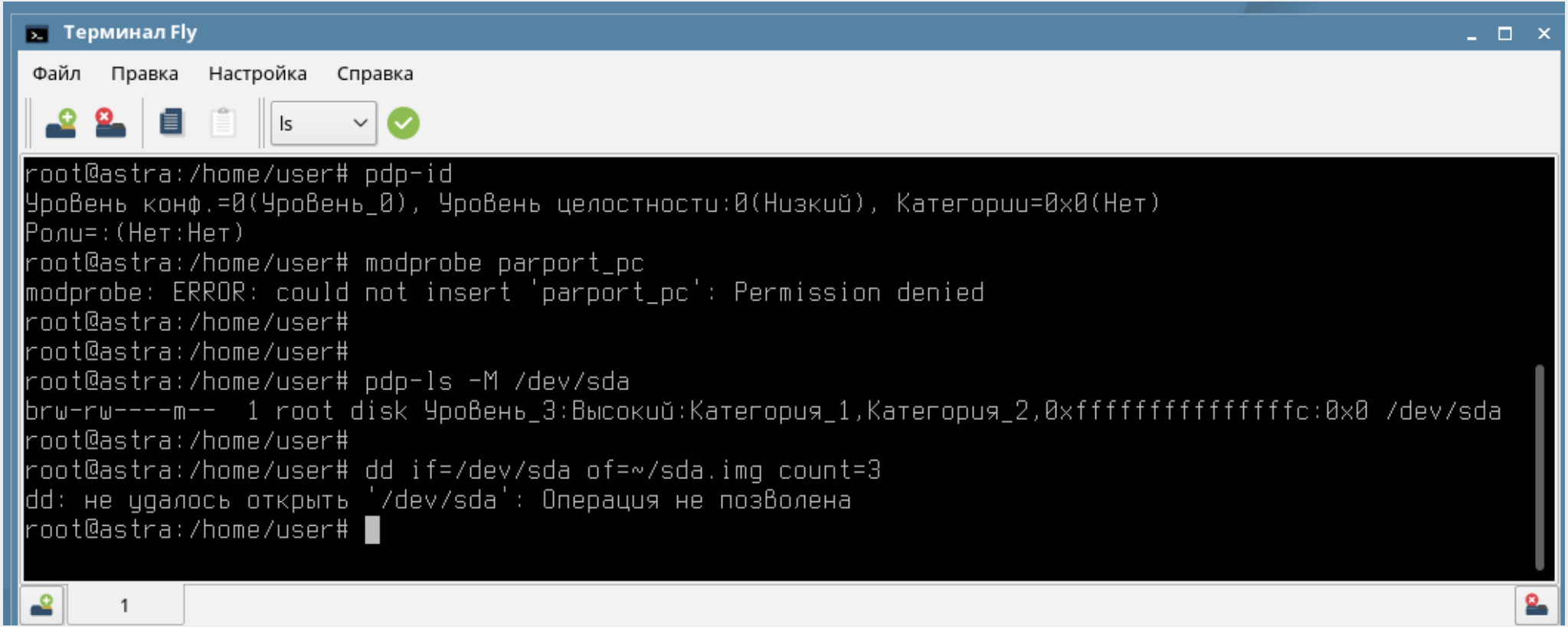
ROOT iL= 0

РАСЩЕПЛЕНИЕ ROOT НА ВЫСОКИЙ УРОВЕНЬ И НИЗКИЙ

БЛОКИРОВКА НАРУШЕНИЯ ЦЕЛОСТНОСТИ И КОНФИДЕНЦИАЛЬНОСТИ

РУСБИТЕХ

ПРИМЕР: Нарушитель с правами обычного пользователя, используя уязвимость, повысил права до суперпользователя (root), но благодаря МКЦ не смог загрузить модуль и снять образ диска с данными



ДАЛЬНЕЙШЕЕ РАЗВИТИЕ ТЕОРИИ И ПРАКТИКИ МКЦ

РУСБИТЕХ

- РОЛЕВАЯ МОДЕЛЬ
- СУБД POSTGRESQL
- БАЗОВЫЕ СРЕДСТВА ВИРТУАЛИЗАЦИИ
- СЕТЕВАЯ ДОМЕННАЯ ИНФРАСТРУКТУРА
-АДМИНИСТРИРОВАНИЕ И КОНФИГУРАЦИЯ

ТРУДОЕМКОСТЬ РЕАЛИЗАЦИИ МРОСЛ ДП-МОДЕЛИ

РУСБИТЕХ

МРОСЛ	KERNEL SPACE	USER SPACE	ВСЕГО
МРД	120	120	240
МКЦ	36	24	60
RBAC	24	36	60
		ИТОГ	360

www.rusbitech.ru

СПАСИБО!

www.astralinux.ru