УТВЕРЖДЕНО

САФТ.00001–01 31 01-ЛУ

ПРОГРАММНЫЕ СРЕДСТВА ЗАЩИТЫ ДАННЫХ В СЕТЯХ И ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

МЕЖСЕТЕВОЙ ЭКРАН «Altell NEO»

(шифр «Altell NEO»)

Описание применения

САФТ.00001–01 31 01

Инв.N подл.	Подп. и дата	Взам.инв.N	Инв.N дубл.	Подп. и дата

Листов 37

2010

Литера

АННОТАЦИЯ

Настоящее Описание применения входит в состав программной документации межсетевого экрана «Altell NEO» (шифр «Altell NEO») и представлено с целью предоставления сведений о назначении программного продукта, условиях его применения, задачах, решаемых при помощи данного программного продукта, входных и выходных данных.

Описание применения разработано в соответствии с требованиями ГОСТ 19.502-78 Описание применения. Требования к содержанию и оформлению.

Содержание

1 НАЗНАЧЕНИЕ ПРОГАММЫ

Межсетевой экран «Altell NEO» (МЭ «Altell NEO») представляет собой многофункциональное компьютерное устройство, состоящее из функционально распределенного программного обеспечения и аппаратных ресурсов.

1.1 Назначение и возможности МЭ «Altell NEO»

МЭ «Altell NEO» предназначен для организации взаимодействия в сетях, построенных по технологиям Ethernet 10/100/1000, описанным в стандартах IEEE 802.2.

В зависимости от типа предполагаемого аппаратного обеспечения и количественных скоростных характеристик, МЭ «Altell NEO» делится на три типа:

1) «Altell NEO» Office;
2) «Altell NEO» Workgroup;
3) «Altell NEO» Enterprise.

Данное деление обуславливает различные скоростные и объѐмные характеристики для сетевого потока.

Функции МЭ «Altell NEO» включают в себя программные компоненты, позволяющие осуществлять логический контроль и обработку передачи пакетов данных по сетям стандарта Ethernet 10/100/1000. Данные программные компоненты включают в себя следующие определения:

1) статический и динамический маршрутизатор;
2) пакетный фильтр.

При этом МЭ «Altell NEO» обеспечивает следующие особенности работы:

1) обеспечивает соединение между двумя и более различными сетями;
2) контролирует поток данных, начиная с 2-го уровня OSI и заканчивая 7-м уровнем;

3) отслеживает состояние активных соединений;
4) позволяет осуществлять фильтрацию потока с учѐтом контекста.

1.2 Ограничения на применение МЭ «Altell NEO»

МЭ «Altell NEO» может применяться в автоматизированных системах (АС) для разделения сегментов информационной сети с целью обеспечения защиты информации от несанкционированного доступа (НСД) в сетях пакетной передачи данных как внутри АС, так и между разными АС.

МЭ «Altell NEO» ориентирован на использование в распределенных АС, построенных на базе технологии Ethernet с пропускной способностью 10/100/1000 Мбит/с и работающих на базе стека протоколов пакетной передачи данных IP.

МЭ «Altell NEO» может использоваться при обработке информации в аттестованных автоматизированных системах. При этом должны неукоснительно соблюдаться все требования, изложенные в документации на изделие.

2 УСЛОВИЯ ПРИМЕНЕНИЯ

МЭ «Altell NEO» имеет моноблочную конструкцию в едином металлическом корпусе 19”.

МЭ «Altell NEO» предназначен для работы в закрытых помещениях при следующих основных характеристиках окружающей среды:

- температура окружающего воздуха от +10°С до +35°С;
- относительная влажность от 40% до 80% при температуре воздуха +25°С;
- атмосферное давление от 84 кПа до 107 кПа.

Межсетевой экран работает при электропитании от однофазной сети переменного тока с дополнительным заземлением номинальным напряжением 220 В и частотой переменного тока 50 Гц. При этом качество электрической энергии должно соответствовать нормам качества в соответствии ГОСТ 13109-87 «Требования к качеству электрической энергии в электрических сетях общего назначения». Электрическое сопротивление и прочность на пробой изоляции между внешними токоведущими цепями и корпусом комплекса должно быть не менее 20 МОм и не ниже 1500 В соответственно при нормальных климатических условиях по ГОСТ 21552-84 «Средства вычислительной техники. Общие технические требования, приемка, методы испытаний, маркировка, упаковка, транспортировка и хранение».

2.1 Необходимые технические средства

Для корректной работы межсетевого экрана (МЭ) «Altell NEO» необходимы следующие технические средства:

1) процессор – 1 ГГц архитектуры i386, или 500МГц архитектуры MIPS64;
2) RAM – 1 ГБайт;
3) Compact Flash – 512 МБайт;

4) LAN – 2 x RJ45 10/100/1000 Base-T;
5) интерфейсы ввода-вывода – USB v2.0, COM;
6) чипсет – с поддержкой контроллеров IDE(ATA), PS/2.

При работе комплекса используются следующие носители данных:

1) Compact Flash — носитель содержит полноценный образ МЭ, необходимого для запуска и функционирования МЭ «Altell NEO». Образ МЭ включает в себя мультизагрузчик и операционную среду с сервисами МЭ. Это стандартный носитель, применяемый для хранения программного (ПО) в составе аппаратной платформы. Compact Flash — носитель представляет собой набор энергонезависимой Flash памяти, подключаемой посредством ATA протокола к аппаратным ресурсам МЭ. По сравнению с традиционными HDD, данный накопитель имеет более высокую надѐжность и устойчивость к вибрациям и ударам, т.к. не содержит движущихся механических частей.
2) HDD — носитель, подключаемый по протоколу SATA, используемый для хранения системного журнала.

2.2 Общие характеристики входной и выходной информации

Входной и выходной информацией являются фреймы стандарта Ethernet T-BASE 10/100/1000.

3 ОПИСАНИЕ ЗАДАЧИ

Общесистемные программные средства МЭ «Altell NEO» представляют собой набор программного обеспечения, функционирующего в составе МЭ «Altell NEO» и обеспечивающего решение следующего круга задач:

1) реализации механизмов контроля доступа;
2) реализации механизмов идентификации и аутентификации;
3) проверки целостности и подлинности компонентов комплекса;
4) организации изолированной программной среды;
5) защиты от сбоев и восстановления работоспособности;
6) очистки памяти для защиты от компрометации конфиденциальных данных;
7) защиты ввода-вывода конфиденциальной информации на отчуждаемый носитель;
8) организации локального и удаленного администрирования;
9) регистрации событий.

3.1 Механизмы контроля доступа

МЭ «Altell NEO» при реализации средств защиты от несанкционированного доступа (НСД) к информации использует механизмы дискреционного и мандатного контроля доступа. При этом в комплексе предусмотрен многоуровневый подход к реализации защиты, обеспечивающий применение защитных механизмов на операционном уровне.

3.1.1 Формальная модель механизма управления доступом

Все рассматриваемые модели безопасности основаны на следующих базовых представлениях:

1) Система является совокупностью взаимодействующих сущностей — субъектов и объектов. Безопасность обработки информации обеспечивается путем решения задачи управления доступом

субъектов к объектам в соответствии с заданным набором правил и ограничений, образующих политику безопасности. Система безопасна, если субъекты не имеют возможности нарушить правила политики безопасности.

2) Все взаимодействия в системе моделируются установлением отношений определенного типа между субъектами и объектами. Множество типов отношений определяется в виде набора операций, которые субъекты могут производить над объектами.
3) Все операции контролируются средствами контроля и либо запрещаются, либо разрешаются в соответствии с правилами политики безопасности.
4) Политика безопасности задается в виде правил, в соответствии с которыми должны осуществляться все взаимодействия между субъектами и объектами. Взаимодействия, приводящие к нарушению этих правил, пресекаются средствами контроля доступа и не могут быть осуществлены.
5) Совокупность множеств субъектов, объектов и отношений между ними определяет состояние системы.

3.1.1.1 Дискреционная модель Харрисона-Руззо-Ульмана

Данная модель реализует произвольное управление доступом субъектов к объектам и контроль за распространением прав доступа.

3.1.1.1.1 Обозначения

Обозначим: S — множество субъектов (осуществляют доступ к информации); O — множество объектов, содержащих защищаемую информацию; $R=\{r1,...,rn\}$ — конечное множество прав доступа, означающих полномочия на выполнение соответствующих действий (чтение, запись, выполнение). Принято считать, что $S \subset O$, т.е. субъекты одновременно являются и объектами (это сделано для того, чтобы включить в область действия модели отношения между субъектами).

Поведение системы моделируется с помощью понятия состояния. *O × S × R* — пространство состояний системы;

M — матрица прав доступа, описывающая текущие права доступа субъектов к объектам (строки - субъекты, столбцы - объекты); *Q = (S,O,M)* — текущее состояние системы.

Любая ячейка матрицы *M[s,o]* содержит набор прав доступа *s* к объекту o, принадлежащих множеству прав доступа *R*.

3.1.1.1.2 Поведение системы

Поведение системы во времени моделируется переходами между различными состояниями. Переход осуществляется путем внесения изменений в матрицу *M* с помощью команд следующего вида:

```
command a(x1,...,xk)
    if (условия выполнения команды)
        r1 in M[xS1,x01] and  r2 in M[xS2,x02] and rm
        in M[xSm ,x0m ]
    then (операции, составляющие команду)
        op1,op2,...,opn
```

где *a* — имя команды; *xi* — параметры команды, являющиеся идентификаторами субъектов и объектов; *si* и *oi* — индексы субъектов и объектов, используемых в параметрах команды; *opt* — элементарные операции (выполняются только в том случае, если все условия, означающие присутствие указанных прав доступа в ячейках матрицы *M*, являются истинными).

3.1.1.1.3 Элементарные операции

В модели допустимы следующие элементарные операции:

enter r into M[s,o] — добавление субъекту *s* права *r* для объекта *o*;

delete r from M[s,o] — удаление у субъекта *s* права *r* для объекта *o*;

create subject s — создание нового субъекта *s*;

create object o — создание нового объекта *o*;

destroy subject s — удаление существующего субъекта *s*;

destroy object o — удаление существующего объекта *o*;

Применение любой элементарной операции op в системе, находящейся в состоянии *Q=(S,O,M)*, влечет за собой переход в другое состояние *Q'=(S',O',M')*, которое отличается от предыдущего состояния *Q* по крайней мере одним компонентом. Выполнение базовых операций приводит к следующим изменениям в состоянии системы:

enter r into M[s,o] (где $s \in S$, $o \in O$)

$O'=O$

$S'=S$

$M'[xs,x0]=M[xs,x0]$, если $(xs,x0) \neq (s,o)$

$M'[s,o]=M[s,o] \cup \{r\}$

Данная операция вводит право *r* в существующую ячейку матрицы доступа. Содержимое ячейки рассматривается как множество, т.е. если это право уже имеется, то ячейка не изменяется. Операция *enter* называется монотонной, т.к. она только добавляет права в матрицу и ничего не удаляет. Предусловие выполнения операции — существование ячейки (существование соответствующих субъекта и объекта).

delete r from M[s,o] (где $s \in S$, $o \in O$)

$O'=O$

$S'=S$

$M'[xs,x0]=M[xS,x0]$, если $(xs,x0) \neq (s,o)$

$M'[s,o]=M[s,o] \setminus \{r\}$

Данная операция удаляет право *r* из ячейки матрицы доступа, если оно там присутствует. Содержимое ячейки рассматривается как множество, т.е. если удаляемое право отсутствует в данной ячейке, то данная операция ничего не делает. Операция *delete* называется немонотонной, т.к. она удаляет информацию из матрицы. Предусловие выполнения операции — существование ячейки (существование соответствующих субъекта и объекта).

create subject s (где $s \in S$)

$O'=O\cup\{s\}$

$S'=S\cup\{s\}$

$M'[xs,xo]=M[xs,xo]$ для всех $(xs,xo)\in S\times O$

$M'[s,x0]=\varnothing$ для всех $x0\in O$

$M'[s,xs]=\varnothing$ для всех $xs\in S$

Операция является монотонной. Предусловие выполнения операции — отсутствие создаваемого субъекта / объекта.

destroy subject s (где $s\in S$)

$O'=O\setminus\{s\}$

$S'=S\setminus\{s\}$

$M'[xs,x0]=M[xs,x0]$ для всех $(xs,x0)\in S'\times O'$

Операция является немонотонной. Предусловие выполнения операции — наличие субъекта.

create object o (где $o\in O$)

$O'=O\cup\{o\}$

$S'=S$

$M'[xs,x0]=M[xs,x0]$, если $(xs,x0)\in S\times O$

$M'[xs,o]=\varnothing$ для всех $xs\in S'$

Операция является монотонной. Предусловие выполнения операции — отсутствие создаваемого объекта.

destroy object o (где $o\in O$)

$O'=O\setminus\{o\}$

$S'=S$

$M'[xs,x0]=M[xs,x0]$ для всех $(xs,x0)\in S'\times O'$

Операция является немонотонной. Предусловие выполнения операции — наличие объекта.

3.1.1.1.4 Формальное описание системы Σ(Q,R,C)

Формальное описание системы Σ(Q,R,C) состоит из следующих элементов:

1) конечный набор прав доступа $R=\{r1, \ldots, rn\}$;
2) конечные наборы исходных субъектов $S0=\{s1,\ldots,sl\}$ и объектов $O0=\{o1,\ldots,oт\}$, где $S0\subseteq O0$;
3) исходная матрица доступа, содержащая права доступа субъектов к объектам — $M0$;

4) конечный набор команд *C={a(x1, ... ,xk)}*, каждая из которых состоит из условий выполнения и интерпретации в терминах перечисленных элементарных операций.

Поведение системы во времени моделируется с помощью последовательности состояний *Qi*, причем *Qi+1=Ci(Qi)*, где *C* — множество команд. Попадание системы в то или иное состояние для заданного начального состояния зависит только от условий команд из *C* и составляющих их операций. Каждое состояние определяет отношения доступа, которые существуют между сущностями системы в виде множества субъектов, объектов и матрицы прав.

3.1.1.1.5 Критерий безопасности модели Харрисона-Руззо-Ульмана

Критерий формулируется следующим образом:

Для заданной системы начальное состояние Q0=(S0,O0,M0) является безопасным относительно права r, если не существует применимой к Q0 последовательности команд, в результате которой право r будет занесено в ячейку памяти матрицы M, в которой оно отсутствовало в состоянии Q0.

Смысл данного критерия состоит в том, что для безопасной конфигурации системы субъект никогда не получит право *r* доступа к объекту, если он не имел его изначально.

Удаление субъекта или объекта приводит к уничтожению всех прав в соответствующей строке или столбце матрицы, но не влечет за собой уничтожение самого столбца или строки и сокращение размера матрицы. Следовательно, если в какой-то ячейке в начальном состоянии существовало право *r*, и после удаления субъекта или объекта, к которым относилось это право, ячейка будет очищена, но впоследствии появится вновь (в результате создания субъекта или объекта), и в эту ячейку с помощью соответствующей команды *enter* снова будет занесено право *r*, то это не будет означать нарушения безопасности.

Из критерия безопасности следует, что для данной модели ключевую роль играет выбор значений прав доступа и их использование в условиях команд. Хотя модель не налагает никаких ограничений на смысл прав и считает их равнозначными, т.е. из них, которые участвуют в условиях выполнения команд, фактически представляют собой не права доступа к объектам, а права управления доступом, или права на осуществление модификаций ячеек матрицы доступа.

Таким образом, данная модель описывает не только доступ субъектов к объектам, но и распространение прав доступа от субъекта к субъекту, т.к. именно изменение содержания ячеек матрицы доступа определяет возможность выполнения команд (в том числе команд, модифицирующих саму матрицу доступа), которые потенциально могут привести к нарушению критерия безопасности.

Положительные стороны модели:

1) данная модель является простой в реализации (т.к. не требует применения сложных алгоритмов);
2) данная модель является эффективной в управлении (т.к. позволяет управлять полномочиями субъектов с точностью до операции над объектом);
3) критерий безопасности данной модели является весьма сильным в практическом плане (т.к. позволяет гарантировать недоступность определенной информации для субъектов, которым изначально не выданы соответствующие полномочия).

Отрицательные стороны модели:

1) доказано, что в общем случае не существует алгоритма, который может для произвольной системы, ее начального состояния $Q0(S0,O0,M0)$ и общего правила r решить, является ли данная конфигурация безопасной;
2) существует уязвимость к атаке с помощью «троянского коня» (т.к. в дискреционных моделях контролируются только операции

доступа субъектов к объектам, а не потоки информации между ними).

3.1.1.2 Мандатная модель Белла-ЛаПадулы

Мандатная модель управления доступом основана на правилах секретного документооборота, принятых в государственных и правительственных учреждениях.

Основным положением является назначение всем участникам процесса обработки защищаемой информации и объектам, в которых она содержится, специальной метки. Такая метка называется *уровнем безопасности*. Все уровни безопасности упорядочиваются с помощью установленного отношения доминирования.

Контроль доступа осуществляется в зависимости от уровней безопасности взаимодействующих сторон на основании двух правил:

1) уполномоченное лицо (субъект) имеет право читать только те документы (объекты), уровень безопасности которых не превышает его собственный уровень безопасности;
2) уполномоченное лицо (субъект) имеет право заносить информацию только в те документы (объекты), уровень безопасности которых не ниже его собственного уровня безопасности.

Таким образом, мандатная модель управляют доступом неявным образом — с помощью назначения всем сущностям системы уровней безопасности, которые определяют все допустимые взаимодействия между ними. Следовательно, мандатное управление доступом не различает сущностей, которым присвоен одинаковый уровень безопасности, и на их взаимодействия ограничения отсутствуют. Поэтому в тех ситуациях, когда управление доступом требует более гибкого подхода, мандатная модель применяется обязательно совместно с дискреционной, которая используется для контроля за взаимодействиями между сущностями одного уровня и для

установки дополнительных ограничений, усиливающих мандатную модель.

3.1.1.2.1 Формальное описание системы

Модель Белла-ЛаПадулы описывает доступ активных объектов, называемых субъектами, к пассивным объектам, называемых объектами. Один объект, в зависимости от типа доступа, может представать в обеих ролях.

Различия между доступом на чтение и запись, могут различаться по четырем способам доступа: ни на чтение, ни на запись (*execute, e*), только чтение (*read, r*), только запись (*append, a*) и чтение-запись (*write, w*).

Набор всех типов доступа назван *A*.

Каждое обращение субъекта *Si* к объекту *Oj* способом *x* описывается таким набором как (*Si,Oj,x*).

Вся эта троица вместе составляет набор текущих обращений *b* .

Объекты структурированы по принципу Отец-Сын и образуют иерархию *H* в одном, или более, независимых деревьях.

Степень защиты задается парой (*S,C*)

S — уровень доступа, значение происходящее из иерархии (например: публично, конфиденциально, секретно, сверхсекретно).

C — множество категорий (категории, к которым принадлежит объект или субъект, формально назначаются согласно рабочему окружению).

Вводится следующее отношение доминирования:

Один объект со степенью защиты (*S1,C1*) доминирует над другим объектом (*S2,C2*), если *S1>=S2* и $C1 \supseteq C2$. Объект, доминирующий над всеми другими объектами, образует отношение доминирования *D*.

Назначение степени защиты для субъектов и объектов.

Функция классификации *F* задается кортежем (*fS, fO, fC*) функций назначения степени защиты, где:

fS(Si) является максимальной степенью защиты для субъекта *Si*,

fO(Oj) является степенью защиты объекта *Oj*

fC(Si) является текущей степенью защиты субъекта *Si*.

Таким образом различаются максимальная и текущая степени защиты субъектов. Для всех *Si fS(Si)* всегда должно доминировать над *fC(Si)*.

Состояние модели *z* задается как набор из элементов (*b, F, H*), таким образом система представляется в виде последовательности состояний с некоторым начальным состоянием *z0*.

Первое используемое свойство это простое свойство защиты — не считываемость. Оно состоит в том, что субъект *Si* может иметь право доступа на чтение к объекту *Oj ((Si,Oj,r)* или *(Si,Oj,w)* это текущий доступ), если *Si* доминирует над *Oj*.

Во избежание копирования злоумышленником объекта в уровень с меньшей степенью защиты добавляется еще одно свойство — не списываемость. Это свойство означает: если субъект *Si* имеет доступ на чтение к объекту *O1* и доступ на запись к объекту *O2*, тогда *O2* должно доминировать над *O1* (*O1* имеет более низкий уровень защиты, чем *O2*). Таким образом информационный поток ограничен свыше.

Каждое свойство добавляется к другим и никогда не понижает защиту системы, т.е. для разрешения доступа требуется выполнение одновременное условий всех свойств. Состояние, удовлетворяющее всем этим требованиям, называется безопасным.

Таким образом, текущий допуск *(Si,Oj,x)* предоставляется только в том случае, если выполнены следующие условия:

1) *Si* доминирует над *Oj*, если *x = r* или *x = w* (*x* заключает в себе доступ на чтение)
2) *Oj* доминирует над текущим уровнем *Si*, если *x = a*
3) уровень *Oj* равен текущему уровню *Si*, если *x = w*
4) текущий уровень *Si* доминирует над уровнем *Oj*, если *x = r*

3.1.2 Принцип и механизм разграничения и управления доступом

МЭ «Altell NEO» имеет многоуровневую структуру и реализует независимые механизмы разграничения и управления доступом на различных своих уровнях.

3.1.2.1 Операционный уровень

На операционном уровне функционирует ОС SeOS.v2, которая включает поддержку расширенных политик безопасности, реализующих механизмы контроля доступа. Данная система называется «расширенная система контроля доступа», или РСКД. Для контроля доступа в системе используются параллельно два принципа — дискреционный контроль доступа и мандатный контроль доступа, что позволяет быстро и эффективно решать задачу разделения и управления доступом к имеющимся ресурсам системы.

РСКД работает на уровне ядра ОС SeOS.v2, при этом отсутствуют возможности каким-либо образом обойти системы контроля. Все необходимые проверки в ядре осуществляется на уровне подсистем управления памятью, работы с устройствами ввода-вывода, поддержки файловых систем, контроля процессов и на уровне системных вызовов.

РСКД состоит из двух компонентов: модулей в составе кода ядра ОС SeOS.v2 и модулей правил, реализующих модели защиты.

РСКД добавляет свои собственные проверки на правомочность запрашиваемых действий к стандартным проверкам в системных вызовах. Как результат, логика работы ядра не меняется, но его интерфейсная часть дополняется новыми возможностями. В архитектуре РСКД механизм защиты делится на три части: модуль контроля системных вызовов (МКСВ), модуль принятия решения (МПР), хранилище информации о правах доступа (ХИПД) и правила контроля доступа (ПКД) приведены на (рис. 1).

Общая архитектура РСКД

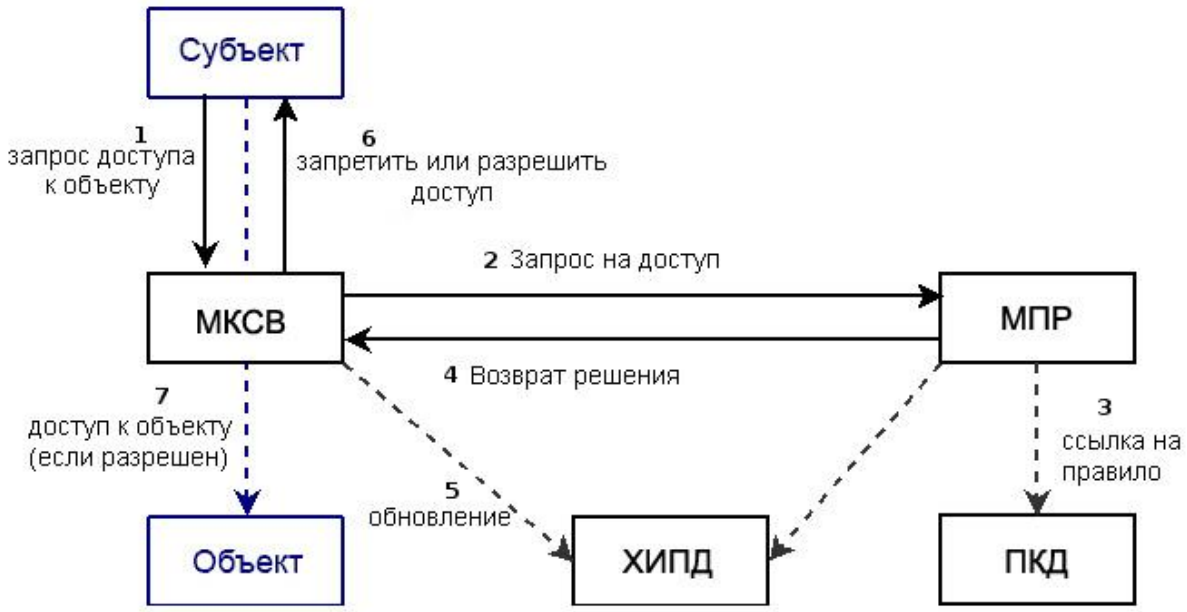

Рис. 1

Принцип работы данных механизмов следующий: системный вызов преобразуется в один из стандартных запросов к модулю принятия решения (МПР), который, выносит вердикт — разрешить или запретить дальнейшее выполнение системного вызова.

Модуль, реализующий модель защиты, получает информацию для принятия решения из атрибутов участников системы, которые хранятся в ХИПД.

В реализации РСКД есть два привилегированных субъекта ОС SeOS.v2: *root*, который обладает правами на администрирование системы, и *secoff*, контролирующий права доступа. При этом *root*, не может ни повлиять на защиту РСКД, ни прочитать значительную часть хранимой в системе информации. Контролем прав в РСКД занимается субъект ОС SeOS.v2 *secoff* («офицер безопасности»), который в остальном — обычный субъект,

неспособный своими действиями повредить систему. Тем не менее, только у него есть полномочия на изменение поведения защиты РСКД. В результате, взлом ОС SeOS.v2 многократно усложняется, поскольку для полного захвата необходимы привилегии двух типов — администратора и офицера безопасности — которые не встречаются одновременно у любого из субъектов внутри ОС.

3.1.2.1.1 Дискреционный контроль доступа

Дискреционный контроль доступа (ДКД) обеспечивает контроль доступа наименованных субъектов ОС SeOS.v2 к наименованным объектам (файлам, программам, томам и т.д.), при этом контроль доступа применим к каждому объекту и каждому субъекту.

Для каждой пары (субъект – объект) задано явное и недвусмысленное перечисление допустимых типов доступа (чтение, запись, выполнение), то есть тех типов доступа, которые являются санкционированными для данного субъекта к данному ресурсу (объекту).

При дискреционном контроле доступа, определенные операции над конкретным ресурсом запрещаются или разрешаются субъектам или группам субъектов. Текущее состояние прав доступа при дискреционном контроле описывается матрицей, в строках которой перечислены субъекты, в столбцах - объекты, а в ячейках - операции, которые субъект может выполнить над объектом.

При дискреционном контроле доступа осуществляется контроль в том числе и для неявных действий пользователя, при которых используются файлы устройств.

3.1.2.1.2 Мандатный контроль доступа

Задачи мандатного контроля доступа (МКД) заключаются в обеспечении сопоставления классификационных меток каждого субъекта и каждого объекта. При этом классификационные метки отражают место субъекта или объекта в соответствующей иерархии. Посредством этих меток

субъектам и объектам назначаются классификационные уровни (уровни уязвимости, категории секретности и т.п.), являющиеся комбинациями иерархических и неиерархических категорий. Данные метки служат основой мандатного принципа разграничения доступа;

При вводе новых данных в систему системой контроля осуществляется сопоставление классификационных меток этим данным. При санкционированном занесении в список пользователей нового субъекта осуществляется сопоставление ему классификационных меток.

МКД реализуется применительно ко всем объектам при явном и скрытом доступе со стороны любого из субъектов. Субъект может читать объект, только если иерархическая классификация в классификационном уровне субъекта не меньше, чем иерархическая классификация в классификационном уровне объекта, и не иерархические категории в классификационном уровне субъекта включают в себя все не иерархические категории в классификационном уровне объекта. Субъект может осуществлять запись в объект, только если классификационный уровень субъекта в иерархической классификации не больше, чем классификационный уровень объекта в иерархической классификации, и все не иерархические категории в классификационном уровне субъекта включаются в не иерархические категории в классификационном уровне объекта.

Реализация мандатных правил разграничения доступа предусматривает возможность их сопровождения – изменения классификационных уровней субъектов и объектов специально выделенными субъектами.

Принцип мандатного управления доступом реализован следующим образом: объектам и субъектам системы назначаются классификационные метки (метки конфиденциальности). При этом классификационная метка состоит из комбинации двух — иерархической и неиерархической:

Иерархические метки:

Иерархическая метка представляются в системе в виде

неотрицательного целого числа и характеризует уровень допуска, при этом на множестве иерархических меток определено отношение порядка — таким образом, что для каждой пары иерархических меток известно, какая из меток соответствует более высокому уровню допуска.

Неиерархические метки:

Неиерархические метки представляет собой набор категорий (a,b,c,d,..), которые поставлены в соответствие объекту или субъекту, и отображаются в системе как набор битов, где каждый бит отвечает за наличие/отсутствие сопоставления объекта или субъекта с определенной категорией. С не иерархическими метками могут осуществляться операции сравнения на основе правил теории множеств. В частности, один набор не иерархических категорий может полностью включать в себя другой набор.

Реализация мандатного доступа использует и иерархическую, и не иерархическую классификации совместно.

Постоянными носителями меток безопасности в каждой из ВМ являются:

1) каталоги;
2) файлы программ и данных;
3) файлы устройств.

Классификационные метки хранятся в файловой системе в хранилище информации о правах доступа.

3.2 Механизмы идентификации и аутентификации

В составе ПО МЭ «Altell NEO» средства идентификации и аутентификации используются исключительно для получения административного доступа для управления и контроля конечного комплекса.

Авторизационные данные составляются из имени пользователя и пароля, предоставляемые пользователем через форму авторизации. Интерфейс управления (далее ИУ) хранит информацию о пользователях в специализированном файле паролей, содержащим информацию об имени

пользователя, его ФИО, параметрах доступа к модулям управления, принадлежности к группе и хеша пароля. Хеш пароля считается по алгоритму MD5.

Авторизация начинается с предоставления имени пользователя и пароля в форме авторизации (первая форма при доступе к ИУ). После ввода данных, ИУ ищет имя пользователя в файле паролей. Если имя пользователя не найдено, ИУ выводит сообщение об ошибке на экран, делается запись в системный журнал. Если пользователь найден, МУ считает MD5 хеш от введѐнного пользователем пароля, который сравнивается с хранящимся в файле паролей md5 хешем, соответствующим данному пользователю. Если посчитанный и хранящийся хеш не совпадают, авторизация считается не успешной, ИУ выводит сообщение об ошибке на экран, делается запись в системный журнал. Если хеши паролей совпадают, то считается что авторизация прошла успешно, ИУ делает запись в системный журнал. Далее ИУ читает информацию о параметрах доступа данного пользователя, и формирует заглавную страницу ИУ в соответствии с указанными параметрами.

Чтобы каждая операция не сопровождалась повторной авторизацией, ИУ формирует ключ сессии, который ассоциируется с сессией пользователя, и прилагается, вместе с именем пользователя, к каждому запросу на выполнение тех или иных команд. Ключ сессии формируется из имени пользователя, IP адреса, с которого осуществляется доступ к ИУ, временной метки начала сессии и случайного числа. Указанные данные формируются в строку, от которой считается хеш по алгоритму MD5. Полученный хеш и является ключом сессии. При этом ИУ хранит данные, из которых ключ сессии был сформирован. При получении запроса с ключом сессии и именем пользователя, ИУ, используя хранящиеся данные, заново высчитывает ключ сессии, сравнивая результат с присланным. Если посчитанный и присланный ключ сессии совпадают, то операция считается авторизованной, то есть разрешѐнной к обработке (обработка происходит согласно параметров

доступа). Если ключи сессии не совпадают, операция считается не авторизованной, и сессия пользователя прекращается, выводится форма авторизации с сообщением об ошибке ключа сессии. Со стороны пользователя ключ сессии хранится в веб браузере в файлах cookies. Время жизни ключа сессии на стороне ИУ — 24 часа. Время жизни ключа сессии на стороне клиента — до конца текущей сессии, то есть до выхода из веб обозревателя.

При принудительном завершении сессии пользователя — отдачи команды на выход, ключ сессии на стороне ИУ уничтожается.

3.3 Проверка целостности и подлинности компонентов

Проверка целостности и подлинности компонентов осуществляется путѐм постоянного наблюдения за работой системы, и регистрацией всех изменений, производимых ПО в файлах (исполняемых, библиотечных и конфигурационных). Все изменения регистрируются в системном журнале.

Механизм наблюдения основан на SCI — интерфейсе системных вызовов ядра ОС SeOS.v2. В области прикладных программ работает специализированный сервис наблюдения, который ожидает сообщений ядра об изменении тех или иных файлов. Ядро, так как является управляющим ресурсом, всегда обладает информацией о том, какая программа что делает с тем или иным объектом. Соответственно, посредством интерфейса системных вызовов может посылать уведомления о тех или иных действиях специализированному сервису, который будет регистрировать изменения. В настоящем ПО МЭ «Altell NEO», регистрируются следующие изменения:

1) Создание файла.
2) Редактирование файла.
3) Удаление файла.

Специализированный сервис наблюдения регистрирует в системном журнале информацию полученную по SCI от ядра в формате:

<дата события><имя хоста><user.alert><событие><имя файла>.

При этом:

1) Дата события — указывается в формате «Мес ДД ЧЧ:ММ:СС».
2) имя хоста — символическое имя конечной системы с ПО МЭ ««Altell NEO»».
3) user.alert — сигнализирует об изменениях в файлах.
4) событие — создание, редактирование или удаление файла.
5) имя файла — полное имя файла на файловой системе (с полным путѐм), например /etc/network/interfaces, interfaces - имя файла, /etc/network — полный путь.

3.4 Организация изолированной программной среды

МЭ «Altell NEO» образует изолированную программную среду, в которой обеспечивается запуск только явно разрешенных сервисов. Программное обеспечение комплекса устойчиво к внедрению (локальному или удаленному) дополнительного ПО. Это достигается такими механизмами как:

1) применением неизменяемых защищенных от записи системных файлов и библиотек;
2) проверкой каждого запускаемого процесса на наличие разрешения на запуск;
3) отсутствием пользовательского доступа к файловой системе;
4) отсутствием в составе комплекса средств разработки программ (редакторов и компиляторов);

3.5 Защита от сбоев и восстановления работоспособности

Защита от сбоев и восстановление подразумевает обнаружение нештатного поведения в работе комплекса, и попытки их устранения. В некоторых случаях возможно только детектирование, без попыток восстановления.

В целом, в соответствии с многоуровневой организаций комплекса, процесс разбит на уровни:

1) Обнаружение на уровне ядра ОС SeOSv2.
2) Обнаружение на прикладном уровне.

3.5.1 Обнаружение на уровне ядра ОС SeOSv2

Обнаружение сбоев на уровне работы ядра делятся на критические и не критические. К критическим сбоям относятся сбои в работе драйверов, сбои в работе ядра. При критических сбоях, происходит перезагрузка комплекса.

Защита от сбоев организованна с помощью использования специализированной файловой системы с журналированием. То есть запись данных осуществляется на транзакционной основе, когда данные сначала записываются в специализированную область диска, и только после того, как они успешно туда записаны — производится перенос их в рабочую область. Конструктивно это выглядит как запись блока данных в выбранное место на диске, при этом выставляются номера секторов с данными в журнал. Когда все сектора записаны на диск, информация с номерами секторов переносится из области журнала в рабочую файловую систему, завершая таким образом транзакцию. То есть, если данные записаны в журнал, но произошѐл сбой, то данные автоматически будут перенесены в положенное им место программой восстановления после сбоев.

3.5.2 Обнаружение на прикладном уровне

На прикладном уровне обнаружение производится системным процессом init, отвечающим за старт прикладных сервисов. Init ведѐт постоянный мониторинг работы запущенных им сервисов, и, в случае сбоя одного из них, производит его перезапуск.

3.6 Очистка памяти

ОС SeOS.v2 включает механизмы по двойной очистке освобождаемой, перераспределяемой и выделяемой памяти методом маскировки случайной последовательностью байт.

ОС SeOS.v2, использует следующие типы памяти:

1) Физическая — для нужд подсистем ядра ОС;

2) Виртуальная — для нужд уровня приложения.

Согласно устройству механизма работы с памятью в ядре ОС, выделение физической памяти для нужд подсистем ядра ОС происходит следующим образом:

1) Запрашивается страница памяти у контроллера памяти;
2) Данная страница отдаётся менеджеру памяти ядра ОС;
3) Менеджер памяти выделяет запрошенный размер блока памяти на странице по запросу от подсистем;
4) Если в выделенной странице не хватает места для выделения блока памяти, запрашивается новая страница у MMU.

Освобождение памяти осуществляется следующим образом:

1) Освобождаемый блок помечается как более не используемый;
2) Менеджер памяти начинает использовать данный блок для нужд других подсистем.

То есть, страницы памяти не возвращаются обратно в MMU до тех пор, пока хотя бы один блок памяти страницы используется для нужд подсистем ядра.

Очистка памяти посредством ее маскировки происходит на следующих этапах:

1) При выделении блока памяти по запросу от подсистем;
2) При пометке освобождаемого блока как более не используемого.

Перераспределение памяти выглядит как последовательность следующего характера:

1) Выделение нового блока памяти;
2) Копирование в неё содержимого старого блока памяти;
3) Освобождение старого блока памяти.

Таким образом, перераспределяемая память так же будет подвергаться маскировке.

Для маскировки создаётся случайная последовательность байт, размером с освобождаемый или выделяемый блок и дважды записывается в

область памяти блока.

Виртуальная память, для использования на уровне приложений, выделяется менеджером памяти ядра ОС из физической, и преобразовывается в виртуальную, доступную для работы в режиме приложения. Виртуальная память запрашивается приложением средствами стандартной библиотеки языка C (libc).

Запрос памяти приложением может осуществляться двумя способами:

1) Запрос памяти из «кучи» через malloc() вызов;
2) Запрос памяти через механизм мапирования области памяти через вызов mmap().

При запросе памяти через malloc(), поведение данной библиотеки схоже с поведением ядра ОС. То есть, при запросе блока памяти приложением, библиотека запрашивает страницу виртуальной памяти у ядра ОС. После получения данной страницы, которая содержит указатели на виртуальную память, библиотека, используя свой собственный менеджер памяти, выделяет приложению запрашиваемый блок. При освобождении приложением данного блока, он помечается менеджером памяти как более не используемый и может быть повторно выделен приложению по запросу.

При использовании механизма мапирования области памяти через вызов mmap(), ядро выделяет область виртуальной памяти, запрашиваемой пользовательским процессом, и подключает данную область памяти в область памяти запросившего его процесса, возвращая указатель на начало области. При освобождении данной области памяти, процесс использует вызов munmap(), для сообщения ядру ОС о том, что данная область памяти более не используется.

Очистка памяти на уровне приложения производится при возвращении блока памяти в кучу и при вызове munmap в библиотеке языка C, а так же при malloc/realloc, mmap. Для этого создаѐтся случайная последовательность байт, размером с выделяемый или освобождаемый блок, и дважды записывается в область памяти блока.

3.7 Организация локального и удаленного администрирования

В МЭ «Altell NEO» локального администрирования как такового нет. Всѐ управление осуществляется удалѐнно, при подключении по специализированному административному интерфейсу (ethernet T-BASE 10/100), с помощью веб обозревателя (веб браузера).

Для осуществления администрирования необходимо подключить кабель типа UTP-5 к порту номер 1 аппаратного комплекса, на котором работает ПО МЭ «Altell NEO», второй конец кабеля можно подключить непосредственно к административному компьютеру, либо к административной сети. Следует учитывать тот факт, что на порту номер 1 аппаратного комплекса, на котором работает ПО МЭ «Altell NEO», работает сервис конфигурации сети (DHCP), который автоматически выделит параметры для конфигурации клиентского интерфейса из диапазона адресов Ipv4 192.168.200.0/24, подсети класса C. Следует учитывать тот факт, что при подключении аппаратного комплекса, на котором работает ПО МЭ «Altell NEO» к сети, в которой так же работает сервис конфигурации сети (DHCP) могут происходить конфликты, поэтому следует исключить подключение к такой сети.

Для доступа к административному ИУ, следует использовать любой, HTML 4.0 совместимый обозреватель. В строке адреса ввести следующий адрес(url): https://192.168.200.1 . Обозреватель покажет окно приглашения к авторизации.

3.8 Регистрации событий

В ПО МЭ «Altell NEO» существует два типа регистрации событий:

1) Системный журнал
2) Журнал ИУ «Altell NEO»

3.8.1 Системный журнал

Системный журнал, или syslog, используется для записи различных

событий, генерируемых работающими сервисами и ядром ОС.

Работу и запись системного журнала обеспечивает специализированный сервис, syslog, имеющий стандартизированный интерфейс доступа из других сервисов и поддержку на уровне библиотеки языка С. То есть любой сервис, с помощью интерфейса системных вызовов и функциональности библиотеки языка С может осуществлять протоколирование своих действий в системный журнал. Конструктивно это выглядит так. Запущенный сервис syslog ожидает сообщений от других сервисов, при получении таковых syslog формирует согласно стандартных правил строку и записывает еѐ в файл журнала. Если сервис syslog не запущен, библиотека языка С, реализующая функционал работы с сервисом системного журнала обеспечивает буферизацию 100кб данных. Данная функциональность полезна при старте системы, так как сервис системного журнала не стартует первым. Соответственно, после старта syslog, все буферизованные данные отправляются непосредственно сервису системного журнала, который производит их обработку. Если сервис системного журнала не запускается, то по истечению лимита в 100кб, буфер данных перезаписывается. В ПО МЭ «Altell NEO», сервис системного журнала запускается вторым сервисом, после сервиса инициализации init.

Хранение записей системного журнала осуществляется в виде простых текстовых файлов со строгой типизированной структурой. Файлы записей системного журнала хранятся в каталоге /var/log.

Структура файл записи системного журнала следующая:

<дата><имя хоста><уровень записи><источник записи><текст>

Где:

1) дата — описание даты и времени в формате «Мес ДД ЧЧ:ММ:СС»
2) Имя хоста — символическое имя конечного комплекса, на котором работает ПО МЭ «Altell NEO», по умолчанию это x86-neo, но это значение можно менять из ИУ

3) Уровень записи — сигнализирует о типе сообщения. Различаются следующие типы сообщений:
4) error — сообщения об ошибке
5) warn — предупреждающие сообщения
6) info — информационные сообщения
7) crit — критические ошибки

Уровень записи формирует сервис, отправляющий сообщение системному журналу:

1) Источник записи — символьное имя сервиса, который осуществляет запись в системный журнал. Если запись осуществляет ядро ОС, то это имя kernel.
2) Текст — произвольный текст сообщения записи, формируемый сервисом, осуществляющим запись в системный журнал.

Все поля записи системного журнала разделены символом пробела. Поле текста записи является последним и может содержать любое количество пробелов и специальных символов. Остальные поля содержат только буквы русского или латинского алфавита, и «:», «-», «.» символы.

В ПО МЭ «Altell NEO» главным файлом записей системного журнала является файл /var/log/messages, в который записываются все сообщения от сервисов и ядра ОС. При достижении размера в 10Мб, файл /var/log/messages переносится в /var/log/messages.0, и новые записи складируются в чистый файл. Если /var/log/messages.0 уже существует, он переносится в /var/log/messages.1 и тд.

3.8.2 Журнал ИУ «Altell NEO»

Журнал ИУ «Altell NEO» не является системным журналом, он предназначен для журналирования действий, осуществляемых пользователем в ИУ при настройке или мониторинге работы МЭ.

Журнал представляет собой текстовый файл, содержащий записи определѐнного формата. Условно каждую запись можно разделить на два

компонента:

1) Общая часть.
2) Специфичная для каждого модуля управления часть.

Общая часть состоит из:

1) Даты — указывается в формате ДД/Мес/ГГГГ.
2) Времени — указывается в формате ЧЧ:ММ.
3) Имени пользователя.
4) Адреса (IP), с которого осуществляется доступ.
5) Имя модуля, в котором производится действие.
6) Ключ сессии пользователя.
7) Текст описания действия.

Специфичная для каждого модуля управления часть следует сразу за текстом описания, и для каждого модуля данная часть отражает детальные характеристики действий конкретного модуля.

1) Модуль сети.

Имеет следующие параметры:

а) Имя интерфейса
б) Действие (создание,удаление,изменение)
в) IP адрес
г) Широковещательный адрес
д) Маска сети

2) Модуль пакетного фильтра
   а) Действие (создание,удаление,изменение)
   б) Тип объекта (таблица,цепочка,правило)
   в) Имя цепочки
   г) Имя таблицы
3) Модуль управления пользователями
   а) Действие (создание,удаление,изменение)
   б) Тип объекта (группа,пользователь)
   в) Имя объекта

г) Список разрешѐнных модулей

4) Модуль системного времени

а) Действие (установка,синхронизация,временная зона)

5) Модуль системы

а) Действие (остановка,перезапуск,запуск)

б) Тип объекта(система или сервис (имя))

Просмотр журнала действий ИУ можно осуществлять выборочно, задавая различные параметры фильтрации. Параметры фильтрации могут быть следующими:

1) Имя пользователя

Может быть задано из списка пользователей, либо вручную. Может быть указано исключение, то есть выборка по всем пользователям кроме выбранного.

1) Имя модуля.

2) Дата

Дату можно задавать не строго, например «За сегодня», или «За вчера», «За неделю», или указывать диапазон дат в формате «от (ДД/Мес/ГГГГ)» - «до (ДД/Мес/ГГГГ)».

4 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Входными и выходными данными являются фреймы стандарта Ethernet T-BASE 10/100/1000.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

МЭ	Межсетевой экран.
ГГц	Гига Герц.
МГц	Мега Герц.
RAM	Запоминающее устройство.
ГБайт	Гига Байт.
RJ45	Разъѐм стандарта «ар-джей».
10/100/1000 Base-T	
USB v2.0	Универсальная последовательная шина.
COM	Последовательный порт компьютера.
IDE	Интерфейс жѐстких дисков (ATA).
ATA	Параллельный интерфейс подключения накопителей (жѐстких дисков и оптических приводов) к компьютеру.
HDD	Накопитель на жѐстких магнитных дисках.
ОС	Операционная система.
ПО	Программное обеспечение.
LAN	Компьютерная сеть.

ПЕРЕЧЕНЬ РИСУНКОВ

Лист регистрации изменений									
	Номера листов (страниц)				*Всего*		*Входящий -*		
Изм	*Изме- ненных*	*Заме- ненных*	*Новых*	*Аннулиро- ванных*	*Листов (страниц) в докум.*	*№ докум.*	*№ сопрово- дительного докум. и дата*	*Подп.*	*Дата*