Н. В. Мозолина

Россия, Москва, Московский физико-технический институт (государственный университет)
Россия, Москва, ЗАО «ОКБ САПР»

ЗАДАНИЕ ЭТАЛОНА ПРИ КОНТРОЛЕ ЦЕЛОСТНОСТИ КОНФИГУРАЦИИ ВИРТУАЛЬНОЙ ИНФРАСТРУКТУРЫ

Предложены три способа выбора эталона при контроле целостности конфигурации виртуальной инфраструктуры: с помощью графов, логических функций и меток. Выделены преимущества и недостатки каждого.

Технология виртуализации за последние годы достигла широкого распространения. Виртуализация позволяет повысить эффективность использования физических ресурсов, памяти и вычислительных мощностей серверов, повысить удобство управления инфраструктурой предприятия, упростить и ускорить процесс тестирования обновлений и многое

другое. Гибкость виртуальной инфраструктуры (ВИ), возможность миграции виртуальных машин, их перенос с одного хранилища на другое помимо преимуществ содержит в себе и опасность использования данной технологии: виртуальная инфраструктура сложна в настройке, обладает большим числом параметров, определенные значения которых могут создавать угрозы безопасности.

Произвести корректную настройку один раз, при вводе системы в эксплуатацию, недостаточно. Мы должны иметь возможность в любой момент жизни системы убедиться, что текущие настройки и связи являются корректными – сравнить их с неким эталоном, – необходим контроль целостности.

Данное требование отражено в нормативных документах РФ [1, 2] – «ЗСВ.7 Контроль целостности виртуальной инфраструктуры и ее конфигураций». Документы [1, 2] и не дают определения конфигурации виртуальной инфраструктуры, но на основе этих документов и рекомендаций [3] можно определить конфигурацию виртуальной инфраструктуры как набор связей между объектами виртуальной инфраструктуры и параметров среды виртуализации [4].

В данной работе рассматриваются связи между объектами виртуальной инфраструктуры. Целостность информации – свойство, заключающемся в существовании информации в неискаженном виде (неизменном по отношению к некоторому фиксированному ее состоянию, эталону). Контроль целостности предполагает сравнение текущего состояния системы с некоторым его выбранным, фиксированным состоянием, принятым за эталон. Если рассматривать эталонную конфигурацию ВИ как некоторый «снимок» конкретного состояния системы, то можно столкнутсяся с тем, что выбор такого эталона зачастую невозможен – например, одним «снимком» нельзя охватить возможность включения виртуальных машин на нескольких хостах, возможность миграции виртуальных машин между хранилищами и вообще любое изменение параметров в некотором диапазоне.

Очевидное решение – создание нескольких эталонных образцов, совпадению с любым из которых говорит о сохранении целостности, – имеет существенные недостатки. С ростом числа объектов в инфраструктуре число эталонов будет стремительно расти. Например, с увеличением количества виртуальных машин (ВМ), каждой из которых разрешена миграция на 2 хоста-гипервизора, рост будет экспоненциальным. Вместе с этим, с ростом числа эталонов значительно будет увеличено и время, необходимое для проверки, соответствует ли текущая конфигурация виртуальной инфраструктуры одному из них: в среднем необходимо число проверок, равное половине числа эталонов.

Предложим различные варианты задания эталона конфигурации виртуальной инфраструктуры и рассмотрим преимущества и недостатки каждого из них. Для начала рассмотрим упрощенную задачу: в виртуальной инфраструктуре существует лишь 2 типа объектов: виртуальные машины и хосты-гипервизоры (хосты). В любой момент времени для каждой виртуальной машины определена принадлежность некоторому хосту-гипервизору (связь между машиной и хостом). В таком случае, конфигурация виртуальной инфраструктуры будет задаваться набором объектов, существующих в данный момент, и связями между этими объектами, – виртуальными машинами и хостами-гипервизорами.

Также для каждой виртуальной машины задано множество хостов-гипервизоров (назовем их разрешенными хостами для данной машины, а также назовем разрешенными связи с такими хостами), которым эта машина может принадлежать. Зафиксируем набор объектов виртуальной инфраструктуры, который будем считать эталонным. Целостность конфигурации виртуальной инфраструктуры определим следующим образом:

Определение 1. В случае если при текущей конфигурации не изменился состав объектов ВИ, а также каждая ВМ работает на одном из разрешенных хостов (или, что то же самое, все связи между машинами и хостами являются разрешенными), мы считаем, что целостность конфигурации системы не нарушена. Такую конфигурацию назовем корректной.

Рассмотрим способ задания эталона с помощью графа [4]. Зададим следующую математическую модель.

Рассмотрим 2 упорядоченных множества объектов O_1 и O_2. Объекты $f \in O_1$ могут быть связаны только с объектами $s \in O_2$, при этом 2 объекта из O_1 или 2 объекта из O_2 не могут быть связны между собой.

Если каждый объект представить в виде вершины, а связь между объектами в виде ребра, то мы получим граф $G = \{V, E\}$, где $V = O_1 \cup O_2$ – множество вершин и $E \subseteq \{(f, s) | f \in O_1, s \in O_2\}$ – множество ребер.

Пусть $G^0 = \{V^0, E^0\}$ – некоторый граф, который назовем эталонным.

Будем называть граф $G' = \{V', E'\}$ - разрешенным для G^0, если выполняются следующие 2 условия:

1. G' – суграф G^0, т.е. $V' = V^0, E' \subseteq E^0$; (1)

2. $\forall a \in V' \exists e' \in E' : e' = (a, b), b \in V'$, если $\exists e \in E : e = (a, b), b \in V^0$. (2)

Покажем, что если в качестве объектов $O_1 = VM = \{vm_i, i = \overline{1,k}\}$ рассматривать виртуальные машины в ВИ, а в качестве $O_2 = H = \{h_i, i = \overline{1,l}\}$ – хосты, то заданная выше математическая модель позволит контролировать целостность упрощенной ВИ, состоящей только из хостов и виртуальных машин.

Эталонный граф G^0 зададим следующим образом: $V^0 \subset O_1 \cup O_2$ будет содержать все объекты (виртуальные машины и хосты-гипервизоры), существующие в системе, конфигурацию которой мы считаем корректной, а E^0 – все разрешенные связи (между виртуальными машинами и их разрешенными хостами). То есть, если мы разрешаем включение виртуальной машины $vm_a \subset V^0$ только на хостах $h_i \in V^0, i - \overline{1,n}$, то E^0содержит все ребра вида $(vm_a, h_i), i = \overline{1,n}$ и не содержит ни одного ребра $(vm, h_j), j \notin \overline{1,n}$.

Зададим граф G^c– граф текущего состояния следующим образом: $V^c \subset O_1 \cup O_2$ будет содержать все объекты (виртуальные машины и хосты-гипервизоры), существующие в системе в данный момент, а E^c – все существующие связи между объектами.

Докажем, что определение 1 эквивалентно выполнению условий (1)–(2) для графа G^c.

Если выполнено условие (1), то:

- состав объектов виртуальной инфраструктуры не изменился: $V^c = V^0$, то есть множества виртуальных машин и хостов-гипервизоров остались неизменными;
- связи между ВМ и хостами в текущем состоянии являются разрешенным: $E^c \subseteq E^0$, то есть любая связь, которая есть в G^c, является разрешенной, так как существовала и в G^0.

Если выполнено условие (2), то не существует виртуальных машин, не подключенных к какому-либо хосту. Это условие является проверкой, что рассматриваемый граф G^c имеет смысл, то есть действительно отражает некоторое состояние системы.

То есть из выполнения условий (1) – (2) следует, что целостность конфигурации системы не нарушена.

В другую сторону: если целостность конфигурации не нарушена, то каждой виртуальной машины соответствует некоторый хост, то есть выполняется условие (2), а также состав объектов виртуальной инфраструктуры не изменился ($V^c = V^0$), а каждая связь между машиной и хостом является разрешенной, то есть существовала в G^0 ($E^c \subseteq E^0$) – выполнено условие (1).

Значит, из целостности конфигурации виртуальной инфраструктуры следует выполнение условий (1)–(2).

Таким образом, в рамках упрощенной задачи определение целостности конфигурации виртуальной инфраструктуры эквивалентно выполнению условий (1)–(2) для графа G^c.

В реальных виртуальных инфраструктурах типов объектов больше, чем 2 – нельзя исключать из рассмотрения хранилища, сети, кластеры, в которые объединяются хосты, и

другие объекты. Тогда для каждой пары типов объектов будем представлять свой эталонный граф, а текущую конфигурацию будем считать корректной, если граф текущего состояния, построенный для каждой пары типов объектов, является разрешенным. При таком подходе мы встречаем ограничение: связи между различными парами объектов считаются независимыми.

Приведем пример использования эталонных графов для контроля целостности конфигурации виртуальной инфраструктуры, в которой существуют виртуальные машины, хосты-гипервизоры и хранилища (рис. 1).

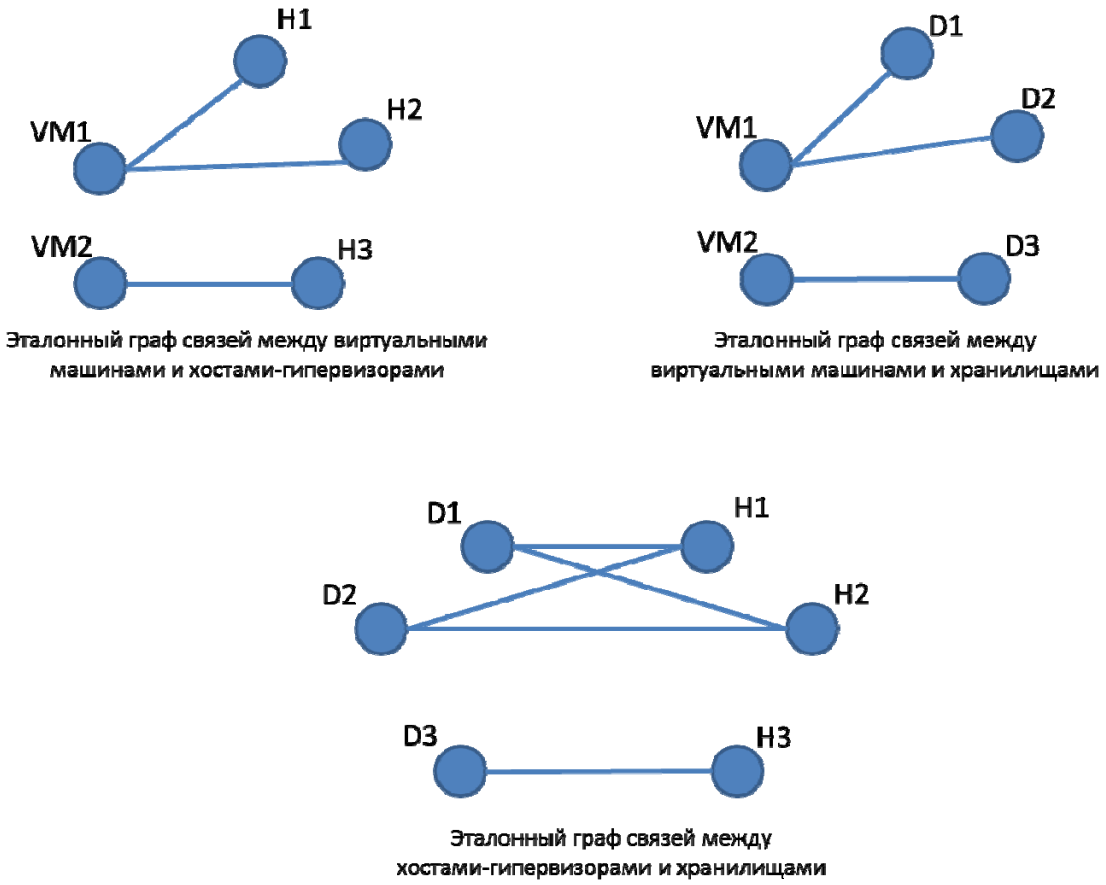

Рис. 1. Эталонные графы

Пример 1. Пусть в виртуальной инфраструктуре существует 2 виртуальные машины $VM = \{VM1, VM2, VM3\}$, 3 хоста-гипервизора $H = \{H1, H2, H3\}$ и 3 хранилища $D = \{D1, D2, D3\}$. Пусть $VM1$ разрешено принадлежать только хостам $H1$ и $H2$, а $VM2$ – только $H3$. Файлы $VM1$ разрешено хранить исключительно на $D1$ или $D2$, файлы $VM2$ – на $D3$. При этом к хостам-гипервизорам $H1$ и $H2$ разрешено подключение каждого из хранилищ $D1$ и $D2$, к $H3$ – только $D3$.

Рассмотрим следующий способ задания эталона, который заключается в описании правил конфигурации виртуальной инфраструктуры на языке функций алгебры логики. Начнем опять же с упрощенной задачи: в виртуальной инфраструктуре существуют объекты 2 типов – виртуальные машины и хосты. $Obj^0 = \{a, b, c, \dots, n, \dots\}$. Зафиксируем набор объектов Obj^0, который будем считать эталонным. Текущий набор объектов – Obj^c.

Каждой возможной связи между объектами ставится в соответствие логическая переменная: $r_{a,b}$, отвечающая за связь между объектами *a* и *b*.

$$r_{a,b} = r_{b,a} = \begin{cases} 1, \text{связь между объектами } a \text{ и } b \text{ реализована (существует)}, \\ 0, \text{в противном случае.} \end{cases}$$

Для каждого объекта $a \in Obj^0$ составляется логическая функция

$$F_a = F\left(r_{a,i_1}, r_{a,i_2}, \ldots\right), \text{где } i_j \in Obj^0, j \in N.$$

Логические функции строятся таким образом, что

$$F_a = \begin{cases} 1, \text{все связи вида } r_{a,i_j} \text{ являются разрешёнными}, \\ 0, \text{в противном случае.} \end{cases}$$

Докажем, что определение целостности конфигурации виртуальной инфраструктуры эквивалентно выполнению условий (3)–(4):

1. $Obj^c = Obj^0$; (3)
2. $\forall a \in Obj^0: F_a = 1$ при текущей конфигурации. (4)

Если выполнено условие (3), то состав объектов виртуальной инфраструктуры не изменился, а выполнение условия (4) гарантирует, что все связи являются разрешенными (по построению логических функций). Значит, из (1) – (2) следует определение 1.

Если целостность конфигурации виртуальной инфраструктуры не нарушена, то по определению не изменился состав объектов, то есть выполнено условие (3), а также все связи являются разрешенными, значит все функции $F_a, a \in Obj^0$ принимают значение 1, то есть выполняется (4). Значит, из определения 1 следует (3)–(4).

Таким образом, в рамках упрощенной задачи данное выше определение целостности конфигурации виртуальной инфраструктуры эквивалентно выполнению условий (3)–(4).

Заметим, что ограничение, которое имеет место при применении графов, – независимость связей между различными парами объектов – при применении логических функций отсутствует.

С ростом числа типов объектов в виртуальной инфраструктуре повысится сложность функций F_a, но сам подход останется неизменным.

Приведем пример использования логических функций для контроля целостности конфигурации виртуальной инфраструктуры, в которой существуют виртуальные машины, хосты-гипервизоры и хранилища.

Пример 2. Пусть состав виртуальной инфраструктуры и накладываемые на нее условия совпадают с примером 1. Тогда мы получим следующие логические функции:

$$F_{VM1} = (r_{VM1,H1} + r_{VM1,H2}) \cdot (r_{VM1,D1} + r_{VM1,D2}) \cdot \overline{r_{VM1,H3}} \cdot \overline{r_{VM1,D3}}$$
$$F_{VM2} = r_{VM2,H3} \cdot r_{VM2,D3} \cdot \overline{r_{VM2,H1}} \cdot \overline{r_{VM2,H2}} \cdot \overline{r_{VM2,D1}} \cdot \overline{r_{VM2,D2}}$$
$$F_{H1} = \left(\overline{r_{VM2,H1}} + r_{H1,D1} + r_{H1,D2}\right) \cdot \overline{r_{H1,D3}}$$
$$F_{H2} = (\overline{r_{VM2,H2}} + r_{H2,D1} + r_{H2,D2}) \cdot \overline{r_{H2,D3}}$$
$$F_{H3} = \overline{r_{VM1,H3}} \cdot r_{VM2,H3} \cdot \overline{r_{H3,D1}} \cdot \overline{r_{H3,D2}} \cdot r_{H3,D3}$$
$$F_{D1} = \overline{r_{VM2,D1}} \cdot \left(r_{H1,D1} + r_{H2,D1}\right) \cdot \overline{r_{H2,D1}}$$
$$F_{D2} = \overline{r_{VM2,D2}} \cdot \left(r_{H1,D2} + r_{H2,D2}\right) \cdot \overline{r_{H2,D2}}$$
$$F_{D3} = r_{VM2,D3} \cdot \overline{r_{VM1,D3}} \cdot \overline{r_{H1,D3}} \cdot \overline{r_{H2,D3}} \cdot r_{H3,D3}$$

Рассмотрим следующий способ задания эталона на основе особых меток для каждого объекта виртуальной инфраструктуры.

Начнем с упрощенной задачи – 2 типа объектов, виртуальные машины и хосты, существуют в виртуальной инфраструктуре: $VM = \{vm_i, i = \overline{1,k}\}, H = \{h_i, i = \overline{1,l}\}$. Зафиксируем наборы виртуальных машин и хостов VM^0 и H^0, которые будем считать эталонными. Наборы объектов в текущем состоянии – VM^c и H^c.

Для каждой виртуальной машины $vm \in VM^0$ зададим эталонную метку M^0_{vm}, являющуюся множеством хостов-гипервизоров, которым данной машине разрешено принад-

лежать: $M^0_{vm} = \{h_a, h_b, \dots\}$. Для каждого хоста $hy \in H^0$ определим метку M^0_{hy}, являющуюся множеством виртуальных машин, которым разрешено принадлежать данному хосту: $M^0_{hy} = \{vm_a, vm_b, \dots\}$.

В любой момент времени для виртуальной машины vm можно получить текущую метку $M^c_{vm} = \{h_c\}$, где h_c – хост, которому принадлежит vm. Для хоста-гипервизора hy текущая метка M^c_{hy} – множество виртуальных машин, которые в данный момент принадлежат хосту hy.

Докажем, что определение 1 эквивалентно выполнению условий (5)–(7):

1. $VM^c = VM^0, H^c = H^0$; (5)
2. $\forall vm \in VM^0, \forall hy \in H^0: M^c_{vm} \subseteq M^0_{vm}, M^c_{hy} \subseteq M^0_{hy}$, (6)
3. $\forall vm \in VM^0, M^c_{vm} \neq \emptyset$ (7)

Если выполнено условие (5), то состав объектов виртуальной инфраструктуры не изменился, а выполнение условия (6) гарантирует, что все виртуальные машины находятся на разрешенных хостах. Выполнение условия (7) является проверкой, что каждая машина принадлежит некоторому хосту, то есть рассматриваемое состояние корректно. Значит, из (5)–(7) следует определение 1.

Если целостность конфигурации виртуальной инфраструктуры не нарушена, то по определению не изменился состав объектов, то есть выполнено условие (5), а каждая виртуальная машина работает на разрешенном хосте, из чего следует выполнение (6). Так как каждая машина всегда принадлежит некоторому хосту, то выполняется (7).

Таким образом, в рамках упрощенной задачи данное выше определение целостности конфигурации виртуальной инфраструктуры эквивалентно выполнению условий (5)–(7).

В реальных виртуальных инфраструктурах увеличение числа типов повлияет на условие (5): необходимо будет проверять, что множество объектов каждого типа осталось неизменным. Также несколько изменится способ задания метки как эталонной, так и текущей.

Эталонная метка будет являться набором множеств, в каждое из которых будут записываться все объекты определенного типа, с которыми у данного разрешена связь. В текущую метку объекта будут входить множества, состоящие из тех объектов, с которыми в данный момент у рассматриваемого существует связь.

Тогда, условие (6) преобразуется в условие включения множеств текущей метки в соответствующие множества эталонной.

Необходимость выполнение условия (7) для каждого множества, входящего в текущую метку, должно определяться отдельно: например, виртуальная машина не может не принадлежать некоторому хосту, но ей можно разрешить быть подключенной к некоторой сети или не быть подключенной вовсе.

При таком подходе мы опять же сталкиваемся с ограничением: связи между различными парами объектов считаются независимыми.

Приведем пример использования меток для контроля целостности конфигурации виртуальной инфраструктуры, в которой существуют виртуальные машины, хосты-гипервизоры и хранилища.

Пример 3. Пусть состав виртуальной инфраструктуры и накладываемые на нее условия совпадают с примером 1. Тогда для каждого объекта эталонные метки будут следующими:

$$M^0_{VM1} = \left\{M^0_{VM1,H} = \{H1, H2\}, M^0_{VM1,D} = \{D1, D2\}\right\}$$

$$M^0_{VM2} = \left\{M^0_{VM2,H} = \{H1, H2\}, M^0_{VM2,D} = \{D1, D2\}\right\}$$

$$M^0_{H1} = \left\{M^0_{H1,VM} = \{VM1\}, M^0_{H1,D} = \{D1, D2\}\right\}$$

$$M^0_{H2} = \left\{M^0_{H2,VM} = \{VM1\}, M^0_{H2,D} = \{D1, D2\}\right\}$$

$$M^0_{H3} = \left\{M^0_{H3,VM} = \{VM2\}, M^0_{H3,D} = \{D3\}\right\}$$

$$M^0_{D1} = \left\{M^0_{D1,VM} = \{VM1\}, M^0_{D1,H} = \{H1, H2\}\right\}$$

$$M^0_{D2} = \left\{M^0_{D2,VM} = \{VM1\}, M^0_{D2,H} = \{H1, H2\}\right\}$$

$$M^0_{D3} = \left\{M^0_{D3,VM} = \{VM3\}, M^0_{D3,H} = \{H3\}\right\}$$

Условие (7) в таком случае примет следующий вид:

$$\forall i = \overline{1,2} \;\; M^0_{VMi,H} \neq \emptyset, M^0_{VMi,D} \neq \emptyset$$

$$M^0_{H3,VM} \neq \emptyset$$

$$\forall i = \overline{1,3} \;\; M^0_{Di,H} \neq \emptyset$$

$$M^0_{D3,VM} \neq \emptyset$$

Сравним предложенные варианты задания эталона.

Самым гибким вариантом является задание эталона с помощью логических функций. При задании эталона с помощью графов или меток мы сталкиваемся с ограничением на независимость связей между парами объектов различного типа.

Нахождение ошибок (неверных связей) при применении логических функций более сложная задача, чем в случае использования графов или меток. Графы и метки позволяют сразу найти, какая связь оказалась не разрешенной, в то время как при применении логических функций это требует тщательного анализа самой функции.

Существенным преимуществом применения графов является наглядность, возможность визуализации, что также облегчит поиск ошибок конфигурации. Два других подхода этим преимуществом не обладают.

Задания эталона с помощью эталонных меток напоминает использование мандатной политики с ее неиерархическими категориями, а потому будет удобно и понятно при использовании на практике специалистами по информационной безопасности.

Сравнивая три предложенных способа задания эталона конфигурации виртуальной инфраструктуры, мы можем найти преимущества и недостатки каждого способа. Выбор применяемого на практике способа должен осуществляться в зависимости от целей и задач обеспечения информационной безопасности и конкретной реализации виртуальной инфраструктуры.

Список литературы

1. Приказ № 17 ФСТЭК России от 11 февраля 2013 г. «Об утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах».

2. Приказ № 21 ФСТЭК России от 18 февраля 2013 г. «Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных».

3. Методический документ ФСТЭК России от 11 февраля 2014 г. «Меры защиты информации в государственных информационных системах».

4. Мозолина, Н. В. Контроль целостности виртуальной инфраструктуры и ее конфигураций / Н. В. Мозолина // Комплексная защита информации : материалы XXI науч.-практ. конф. (Смоленск, 17–19 мая 2016 г.). – М., 2016. – С. 167–170.