УДК 004.056.2

Выбор способа задания эталона при контроле целостности конфигурации виртуальной инфраструктуры

Н.В. Мозолина

Московский физико-технический институт (государственный университет)

ЗАО «ОКБ САПР»

1 Введение

Большое число параметров настройки, связей между элементами, которые можно изменять, делает современные информационные системы не только гибкими, удобными и эффективными для решения большого спектра задач, но и оказывает негативное влияние на информационную безопасность системы: с ростом числа параметров контролировать корректность конфигурации системы становится сложнее, в то время как различные настройки или их сочетания могут создавать угрозы безопасности. Становится необходимым отслеживать, что текущая конфигурация системы корректна, то есть отвечает установленным требованиям безопасности, соответствует некоторому эталону, принятому для этой системы – необходим контроль целостности конфигурации.

Технология виртуализации, получившая в последние годы широкое распространение, позволяет создавать информационные системы – виртуальные инфраструктуры, – позволяющие повысить эффективность использования физических ресурсов, памяти и вычислительных мощностей серверов, снизить их время простоя, улучшить управление системой, упростить и ускорить процесс тестирования обновлений и многое другое. Необходимость контроля целостности конфигурации виртуальной инфраструктуры отражена в нормативных документах РФ [1-3].

Наша цель – создать эталон, с помощью сравнения с которым можно сделать вывод, является ли текущая конфигурация системы корректной.

Виртуальная инфраструктура – система динамическая, с возможностью изменений в составе (например, за счёт добавления или удаления виртуальных машин), в связях между объектами (например, миграция виртуальных машин с одного хоста-гипервизора на другой), в параметрах настройки (например, VMware vSphere позволяет включать или отключать DRS на кластерах). Поэтому эталон конфигурации такой системы не может быть задан одним его «снимком», отражающим фиксированные значения всех параметров и связей. Необходимы другие способы задания эталона, учитывающие динамический характер системы.

В дальнейшем рассмотрении сфокусируемся на рассмотрении конфигурации виртуальной инфраструктуры с точки зрения состава объектов и связей между ними.

Во второй части данной работы рассмотрим различные способы задания эталона конфигурации виртуальной инфраструктуры, в третьей – их преимущества и недостатки. В выводе выделим способы, применение которого на практике будет наиболее целесообразным.

2 Способы задания эталона

Для задания эталона при контроле целостности конфигурации виртуальной инфраструктуры можно предложить 4 способа.

1. Набор эталонных «снимков».

2. Эталонные графы.
3. Логические функции.
4. Эталонные метки.

Рассмотрим каждый из способов подробнее сначала на упрощённой виртуальной инфраструктуре, а затем в общем случае.

Будем считать, что в упрощённой виртуальной инфраструктуре существуют только виртуальные машины и хосты-гипервизоры и в любой момент времени для каждой виртуальной машины определена принадлежность некоторому хосту-гипервизору (связь между машиной и хостом). Конфигурация виртуальной инфраструктуры, в таком случае, задаётся набором объектов, существующих в данный момент, и связями между виртуальными машинами и хостами-гипервизорами. Также для каждой виртуальной машины задано множество хостов-гипервизоров (назовём их разрешёнными хостами для данной машины, а также назовём разрешёнными связи с такими хостами), которым эта машина может принадлежать.

2.1 Набор эталонных «снимков»

Для каждого состояния виртуальной инфраструктуры, которое считается корректным, создаётся «снимок» - некоторая модель, определяющая однозначно конфигурацию рассматриваемой системы (например, предложенная в [4]). В результате мы получаем набор эталонных «снимков». Для виртуальной инфраструктуры в текущий момент времени также создаётся «снимок», который сравнивается с каждым из эталонных. В случае совпадения конфигурация виртуальной инфраструктуры является корректной.

Заметим, что при увеличении числа типов объектов в виртуальной инфраструктуре, суть данного способа – построение нескольких «снимков» – не меняется.

2.2 Эталонные графы

Эталонный граф $G^0 = \{V^0, E^0\}$ зададим следующим образом: $V^0 \subset O_1 \cup O_2$ будет содержать все объекты (виртуальные машины $O_1 = VM = \{vm_i, i = \overline{1,k}\}$ и хосты-гипервизоры $O_2 = H = \{h_i, i = \overline{1,l}\}$), существующие в системе, конфигурацию которой мы считаем корректной, а E^0 – все разрешённые связи (между виртуальными машинами и их разрешёнными хостами). То есть, если мы разрешаем включение виртуальной машины $vm_a \in V^0$ только на хостах $h_i \in V^0, i = \overline{1,n}$, то E^0содержит все рёбра 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 $G^c = \{V^c, E^c\}$ – 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.

Конфигурация виртуальной инфраструктуры будет корректной, если граф текущего состояния удовлетворяет следующим условиям:

1. G^c – суграф G^0, то есть $V^c = V^0, E^c \subseteq E^0$; (1)
2. $\forall a \in V^c \; \exists e^c \in E^c : e^c = (a, b), b \in V^c$,
   если $\exists e \in E : e = (a, b), b \in V^0$. (2)

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 разрешённым. При этом, условие (2) должно выполняться лишь для некоторых пар типов объектов (лишь для некоторых графов различных типов объектов): можно требовать принадлежности виртуальных машин хостам-гипервизорам, но не обязательно для каждой сети должны существовать виртуальные машины, подключённые к ней, ровно как и наоборот, - не каждая виртуальная машина может быть подключена к сети.

2.3 Логические функции

Зафиксируем набор объектов виртуальной инфраструктуры $\mathrm{Obj}^0 = \{a, b, c, \dots, n, \dots\}$, 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:

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

$r_{a,b} = r_{b,a}$

Для каждого объекта $a \in \mathrm{Obj}^0$ строится логическая функция $F_a = F\left(r_{a,i_1}, r_{a,i_2}, \dots\right)$, где $i_j \in \mathrm{Obj}^0, j \in N$. Логические функции задаются таким образом, что

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

Текущая конфигурация виртуальной инфраструктуры является корректной, если:

1. $\mathrm{Obj}^c = \mathrm{Obj}^0$;
2. $\forall a \in \mathrm{Obj}^0$: $F_a = 1$ при текущей конфигурации.

2.4 Эталонные метки

Зафиксируем наборы виртуальных машин и хостов $\mathrm{VM}^0 = \{vm_i, i = \overline{1,k}\}$ и $H^0 = \{h_i, i = \overline{1,l}\}$, 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 принадлежать: $M^0_{vm} = \{h_a, h_b, \dots\}$. 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.

Конфигурация виртуальной инфраструктуры в текущий момент времени будет являться корректной, если:

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

3 Преимущества и недостатки предложенных способов

Процесс контроля целостности подразумевает несколько этапов: создание эталона, проверку текущего состояния на соответствие эталону, возможность внесений изменений в эталон при необходимости и поиск ошибок, приведших к нарушению целостности. В связи с этим, рассмотрим преимущества и недостатки каждого из способов с точки зрения простоты создания эталона на основе установленных требований безопасности, возможности изменять эталон при необходимости, проверки инфраструктуры на соответствие эталону, поиска ошибочных связей в системе, а также наглядности, или возможности визуализации установленных требований безопасности.

3.1 Создание эталона и проверка конфигурации на соответствие эталону

Обычно в компании уже существует некоторая политика безопасности, положения которой нужно учесть при построении эталона.

С ростом числа виртуальных машин, каждая из которых может мигрировать между 2 или более хостами (то есть число разрешённых хостов ≥2), количество корректных конфигураций растёт экспоненциально. Экспоненциальным будет и рост числа эталонных «снимков», что является существенным минусом применения данного способа и с точки зрения построения эталона, и с точки зрения проверки инфраструктуры – в среднем потребуется число сравнений текущего «снимка», равное половине эталонных.

Число эталонных графов для виртуальной инфраструктуры и число проверок текущей конфигурации постоянно, невелико и зависит только от числа различных типов объектов в виртуальной инфраструктуре. Но данный способ имеет ограничения: связи между различными парами объектов считаются независимыми. Это ограничивает применение данного способа.

Этим же недостатком обладают и эталонные метки. Проверка соответствия текущих меток эталонным во многом напоминает проверку категорий, неиерархических меток объектов в мандатной политике безопасности, и решение данной задачи (проверки конфигурации виртуальной инфраструктуры) может быть основано на предыдущем опыте реализации мандатной политики [5].

Применение логических функций позволяет описать любые правила политики безопасности, но их построение связано с некоторыми трудностями: например, для описания инфраструктур требуется число функций, равное числу объектов, которое может быть велико.

Так, логические функции предоставляют самый универсальный и гибкий с точки зрения задания эталона на основе политик безопасности способ.

3.2 Изменение эталона

В процессе работы виртуальной инфраструктуры может возникнуть необходимость в изменении состава системы, связей между ними её объектами, изменении требований безопасности, накладываемых на виртуальную инфраструктуру, в таких случаях необходимо изменить эталон, пересчитать его.

Добавление объектов в виртуальную инфраструктуру в случае использования набора эталонных «снимков» влечёт за собой полное изменение набора – уже существующие «снимки» должны быть изменены и, возможно, будут созданы новые. Такой пересчёт эталона влечёт за собой практически создание нового эталона, что опять же трудоёмко и сложно. Аналогичная ситуация возникает и при удалении объектов из инфраструктуры.

Логические функции, используемые для задания эталона, при изменении состава виртуальной инфраструктуры претерпевают значительные изменения. Функции уже существующих объектов оказываются заданными на другом наборе переменных, фактически, строятся заново, также добавляются новые логические функции для каждого нового объекта системы или же, в случае удаления объекта, логическая функция, отвечающая для его связи с другими объектами, исключается из рассмотрения.

Аналогичные изменения происходят и с метками: меняется состав эталонных меток, для каждого нового объекта создаётся новая метка, исключаются из рассмотрения метки удалённых объектов.

В случае использования эталонных графов изменению при добавлении или удалении объектов системы подвергаются не все структуры, входящие в эталон. Например, создание новой виртуальной машины не повлияет на эталонный граф связей между хранилищами и хостами-гипервизорами.

Таким образом, с точки зрения вносимых изменений в эталон при добавлении или удалении объектов самым удобным способом является применение эталонных графов.

3.3 Поиск ошибочных связей

После проверки целостности конфигурации инфраструктуры недостаточно получить результат: сохранена целостность или нет, - нужно в случае нарушений целостности уметь находить ошибочные связи или объекты системы для дальнейшего приведения системы в корректное состояние.

В случае изменения состава объектов виртуальной инфраструктуры при использовании любого способа задания эталона сравнение множеств объектов текущего и эталонного даст ответ, какие изменения произошли.

Поиск ошибочных связей труднее.

В случае использования набора эталонных «снимков» сказать, какая именно связь является ошибочкой, довольно сложно. Находить их можно следующим образом: сравнивать «снимок» текущего состояния с каждым эталонным и находить расстояние (количество отличающихся связей) между ними. Сравнивая текущий «снимок» с тем из эталонных, с которым наименьшее расстояние, находим ошибочные связи. Минусом такого подхода является полный перебор эталонов, а также тот факт, что наименьшим может быть расстояние между текущим «снимком» и несколькими эталонными, что ставит нас в условия неопределённости, что считать ошибками.

Если все логические функции привести к нормальной конъюнктивной форме, то выявление ошибочных связей можно произвести следующим образом: рассмотреть каждую дизъюнкцию и найти те, которые при текущем состоянии конфигурации принимают нулевое значение. В них и будут содержаться ошибочные связи. Дальше рассматривая все возможные наборы значений переменных, входящих в эти «нулевые» дизъюнкции, найдём тот набор, при котором все логические функции, заданные в системе, будут равны 1. Различия с этим набором и будут составлять ошибочные связи.

В случае использования меток и эталонных графов нахождение ошибочных связей наиболее просто. При использовании эталонных графов можно всегда сказать, какая связь выводит текущий граф из множества суграфов данного, а значит, мы можем сразу определить ошибочные связи. И если множество, входящее в текущую метку объекта, не включено в соответствующее множество из эталонной метки, то расходящиеся элементы и являются ошибочными связями.

Получается, что для определения ошибочных связей более всего подходят способы задания эталона с помощью набора эталонных «снимков» и с помощью эталонных меток.

3.4 Наглядность эталона

На данном этапе развития информационных технологий полностью исключить человека из работы с системами защиты виртуальных инфраструктур невозможно, а потому наглядность эталона, возможность с его визуализации, одно из важных его свойств [5].

Наглядность применения эталонных «снимков» будет зависеть от выбора способа задания «снимка», но в любом случае их число может быть настолько велико, что просмотр их человеком становится практически невозможным.

Использование логических функций неудобно с точки зрения наглядности и визуализации – во-первых, при данном способе число переменных велико, во-вторых, исчисление логических функций сложная задача для человека.

Эталонные графы – удобный для визуализации способ. Он наглядно представляет связи между объектами и близок к привычному для администраторов виртуальной инфраструктуры представлению системы [6].

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

Так, среди предложенных способов эталонные графы и эталонные метки являются наиболее удобными с точки зрения наглядности и возможности визуализации.

4 Заключение

В данной работе были рассмотрены 4 способа задания эталона конфигурации виртуальной инфраструктуры: набор эталонных «снимков», эталонные графы, логические функции и эталонные метки. Были выявлены преимущества и недостатки каждого.

Из предложенных способов наиболее удачными для применения на практике являются эталонные графы и логические функции. Их совместное применение, задание эталона с помощью графов и его дополнение при необходимости логическими функциями, позволит на практике использовать преимущества каждого способа: наглядность и простоту изменения эталона, характерные для эталонных графов, а также гибкость и универсальность логических функций.

Литература

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 её конфигураций. // Комплексная защита информации: материалы XXI научно-практической конференции, Смоленск, 17–19 мая 2016 г. М., 2016. С. 167-170.
5. *Маренникова Е. А.* Мандатный механизм разграничения доступа — это просто // Комплексная защита информации: материалы XXI научно-практической конференции, Смоленск, 17–19 мая 2016 г. М., 2016. С. 209-212.
6. Using VirtualCenter maps to display VMware Infrastructure relationships [Электронный ресурс] URL: http://searchvmware.techtarget.com/tip/Using-VirtualCenter-maps-to-display-VMware-Infrastructure-relationships (дата обращения: 12.04.2016)