УДК 004.75, 004.78, 608.4

СЕРВИС BLACKBOX ДЛЯ ПРОВЕДЕНИЯ СОРЕВНОВАНИЙ ПО ЗАЩИТЕ КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ CAPTURE THE FLAG

Н. И. Анисеня, Д. А. Стефанцов, Т. А. Торгаева

Представляется разработанная командой Томского госуниверситета SiBears система BlackBox для проведения соревнований по защите компьютерной информации Capture the Flag, основанных на выполнении заданий. Описывается функциональность системы, особенности её разработки и администрирования.

Ключевые слова: *SiBears, BlackBox, CTF.*

Широко распространённые в настоящий момент соревнования по защите компьютерной информации Capture the Flag (CTF) [1] являются хорошим дополнением к программе обучения студентов по специальности «Компьютерная безопасность», развивают навыки поиска уязвимостей и защиты от них. Существует несколько форм проведения таких соревнований, основными из которых являются две: 1) командам участников предлагается набор заданий, классифицированных по сложности и по области знаний, используемых при решении; 2) локальные сети команд участников объединяются в игровую сеть с помощью технологии VPN, после чего участники пытаются скомпрометировать серверы других команд и защитить свои серверы. Первая форма используется, в основном, для проведения отборочных соревнований, а вторая — для проведения финалов соревнований.

Команда Томского госуниверситета SiBears [2] принимает активное участие в играх CTF и является организатором нескольких подобных соревнований. Описание программной системы, разработанной SiBears для проведения соревнований второго типа, можно прочитать в [3]. В данной работе описывается программная система BlackBox [4], разработанная командой SiBears, для проведения соревнований первого типа. Разработка BlackBox начата в 2010 г. выпускником кафедры защиты информации и криптографии Томского госуниверситета Алексеем Краснопёровым и студентом той же кафедры Максимом Цоем. В настоящее время поддержкой и модификацией системы занимаются авторы работы. На базе BlackBox проведены соревнования SiBCTF-2011, отборочный тур соревнований RuCTF 2012 [5], соревнования по защите компьютерной информации для школьников School CTF в 2010, 2011 и 2012 гг.

Соревнование в системе BlackBox представляет собой набор заданий, каждое из которых имеет название, текстовое описание и одно или несколько правильных решений, которые могут быть представлены в виде текста. Дополнительно к условию задания могут прилагаться файлы и может предоставляться доступ к виртуальной машине (ВМ), образ которой подготовлен автором задания. Полученный в результате решения задания ответ участник вводит в специальное поле на странице задания, после чего система проверяет корректность этого решения. Если решение признано системой корректным, соответствующему участнику начисляются игровые баллы. Если участник входит в команду, которая участвует в соревновании, частью которого является решённое задание, команде также начисляются баллы.

В настоящий момент в BlackBox существуют две системы подсчёта баллов, полученных за решение заданий во время соревнования: с фиксированным количеством баллов за задание и с переменным. В первом случае автор задания самостоятельно определяет количество баллов, которые начисляются участнику или команде за

правильное решение. Во втором случае каждое задание в соревновании оценивается в 100 баллов, которые распределяются поровну между всеми участниками или командами, решившими его. Таким образом, простые задания, которые решены большим количеством участников, принесут каждому из них мало баллов, и наоборот.

По задумке авторов, сервисом BlackBox может воспользоваться любой желающий для проведения своих собственных соревнований. Поэтому в дополнение к стандартным ролям пользователей системы («наблюдатель», «участник соревнования» и «администратор») реализованы роли «автор соревнования» и «автор задания». Для удобства участников была также введена роль «капитан команды» — специализация роли «участник соревнования». На рис. 1 изображена диаграмма UML вариантов использования системы BlackBox, где показано соответствие между ролями пользователей и функциями системы. Варианты использования, доступные некоторой роли, показаны овалами и соединяются с этой ролью сплошной линией. Стрелки между видами пользователей проводятся от более специализированного вида к более общему. Это означает, что, например, участнику соревнования доступны все функции, доступные наблюдателю, и некоторые дополнительные. Варианты использования на диаграмме обозначены номерами из следующего списка:

1) регистрация в системе и редактирование информации о себе;
2) просмотр рейтингов пользователей и команд — включает в себя просмотр рейтинга команды или пользователя в пределах соревнования или задания или сквозного рейтинга по всем соревнованиям и заданиям;
3) просмотр достижений пользователей и команд — достижения отмечаются при определённых условиях, таких, как победа в соревновании, решение первым трёх задач подряд в соревновании, наименьшее количество неправильных ответов в соревновании;
4) просмотр журнала соревнования и подсчёт рейтинга по нему — пользователям предоставляется возможность посмотреть журнал событий, произошедших в соревновании;
5) решение заданий из архива — после проведения соревнования задачи из него попадают в архив, который может использоваться пользователями для тренировки;
6) комментирование страниц — на веб-страницах сервиса можно оставлять комментарии с уточняющими вопросами к жюри;
7) решение заданий во время соревнования — за решение заданий во время соревнования участникам или командам начисляются баллы;
8) управление командой — капитан команды может пригласить участников в неё, участники могут согласиться или отказаться;
9) создание соревнований и управление ими — включает в себя определение названия, описания, способа вычисления рейтинга, времени проведения и группы авторов заданий соревнования;
10) создание и редактирование заданий — включает в себя определение названия, описания, дополнительных файлов, образа ВМ и корректного ответа задания; вместо корректного ответа автор задания может предоставить программу для проверки ответов пользователей на корректность;
11) просмотр, редактирование и удаление пользователей, соревнований и заданий — администратору системы предоставляется возможность корректировки любых данных системы.

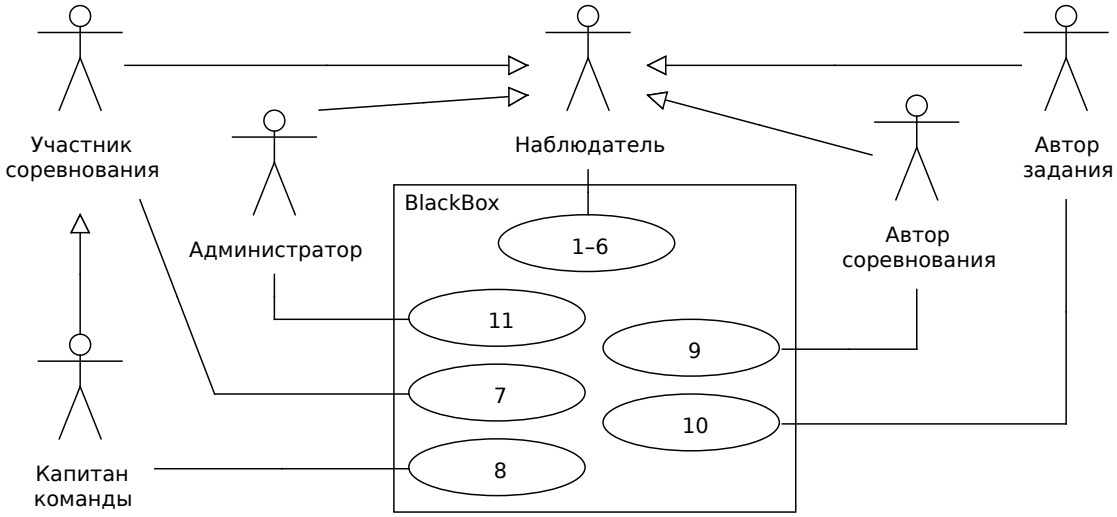

Рис. 1. Диаграмма вариантов использования сервиса BlackBox

Большая часть системы BlackBox реализована на языке программирования Python [6] с использованием библиотеки Django [7]. Основные сложности в реализации системы связаны, во-первых, с предоставлением всем пользователям возможности создания заданий и, во-вторых, с необходимостью синхронизации множества событий, происходящих в системе одновременно.

Поскольку авторами заданий в системе BlackBox могут стать пользователи, которые являются участниками других соревнований в этой же системе, необходимо реализовать механизмы защиты от возможных атак на систему с целью узнать ответы на задания или вызвать отказ системы в обслуживании. При составлении задания автору предоставляется возможность загрузки программы для проверки корректности ответов, присылаемых пользователями. Такая возможность необходима в заданиях, предполагающих множество правильных ответов. Загруженная автором задания программа запускается на одном из серверов системы BlackBox и при этом не является доверенной. Для предотвращения нежелательных действий программы во время выполнения применяется модуль AppArmor [8] подсистемы LSM ядра Linux. Для ограничения программы по времени работы и объёму доступной памяти применяется механизм ограничения ресурсов, выделяемых ядром Linux [9].

Задания могут предполагать доступ участников к некоторому серверу в процессе решения. Например, задание на реализацию атаки типа SQL-инъекция может требовать для решения доступа к веб-серверу, подготовленному автором задания. Система BlackBox предоставляет возможность подготовки образов ВМ, которые будут запущены во время решения соответствующих заданий. Реализация этой функциональности осуществляется с помощью технологии OpenStack [10], позволяющей организовать управление созданием и запуском ВМ на нескольких серверах. OpenStack имеет программный интерфейс на языке Python, что позволяет легко интегрировать управление ВМ в пользовательский веб-интерфейс. Для получения доступа к ВМ для решения некоторого задания пользователь подключается к сети ВМ с помощью протокола OpenVPN.

Проблема синхронного доступа к базе данных решена частично с помощью схемы, допускающей модификацию данных только инструкциями типа INSERT [11], частично — использованием библиотеки Celery [12]. Для оптимизации выполнения часто

приходящих одинаковых запросов, например на просмотр рейтинга, применяется кеширование результатов с помощью memcached [13].

Система BlackBox состоит из нескольких крупных компонент: пользовательского веб-интерфейса, обработчиков пользовательских запросов, средства управления базами данных и нескольких баз данных, сервера для управления ВМ и нескольких серверов ВМ. Каждая из этих компонент может быть установлена на отдельный компьютер для повышения производительности системы при большом количестве участников, соревнований и заданий. Некоторые компоненты допускают установку на несколько компьютеров. На рис. 2 изображена диаграмма UML развёртывания системы BlackBox: объёмными прямоугольниками изображены возможные отдельные физические компьютеры, на которые устанавливаются компоненты; сами компоненты изображены прямоугольником со стереотипом «component». Система BlackBox допускает масштабирование: элементы диаграммы, помеченные стереотипом «повторяемый», могут присутствовать в системе в нескольких экземплярах. Для ускорения обработки большого числа пользовательских запросов можно увеличить количество компьютеров с обработчиками запросов, а для увеличения количества одновременно работающих ВМ — увеличить количество серверов ВМ. При этом один сервер ВМ может одновременно исполнять более одной ВМ.

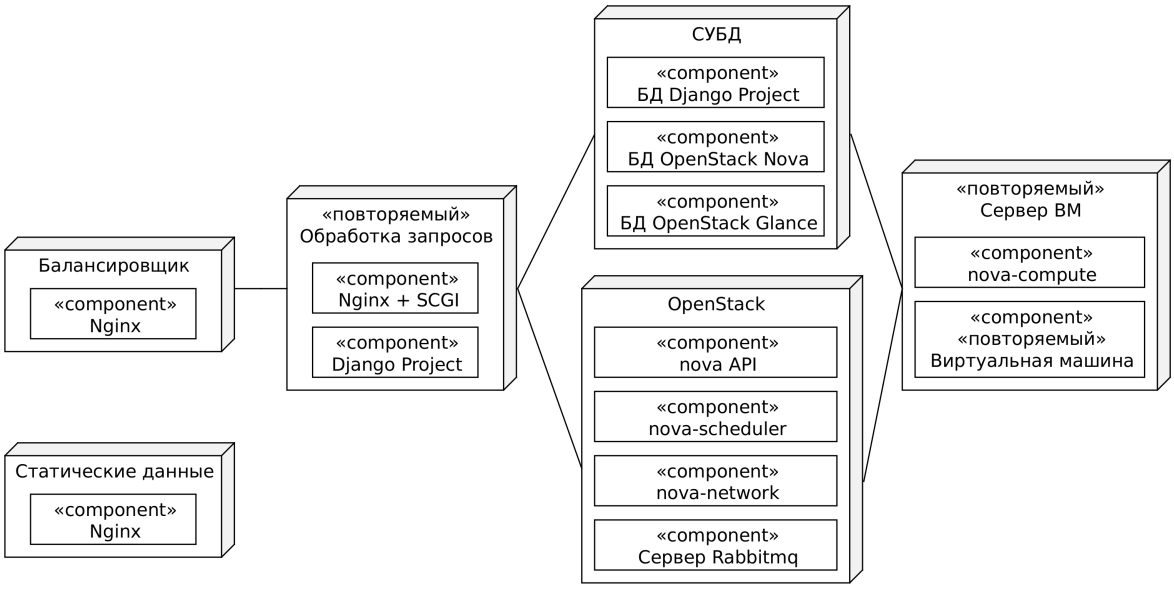

Рис. 2. Диаграмма развёртывания BlackBox

В настоящее время сервис BlackBox удовлетворяет большинству требований, предъявляемых к системам для проведения соревнований CTF, основанных на заданиях. Развитие сервиса видится в его популяризации проведением новых соревнований с большим числом участников, а также интеграцией в него системы для проведения соревнований CTF, основанных на компрометации и защите участниками собственных серверов.

ЛИТЕРАТУРА

1. *Колегов Д. Н., Чернушенко Ю. Н.* О соревнованиях CTF по компьютерной безопасности // Прикладная дискретная математика. 2008. № 2(2). С. 81–83.
2. `http://sibears.ru/` — Команда Томского государственного университета SiBears. 2013.

3. *Ткаченко Н. О., Чернов Д. В.* Разработка и реализация сервера игры CTF // Прикладная дискретная математика. Приложение. 2010. № 3. С. 62–64.
4. http://blackbox.sibears.ru/ — Система для проведения соревнований по защите компьютерной информации. 2013.
5. http://ructf.org/2012 — Всероссийские межвузовские соревнования по защите информации RuCTF 2012. 2012.
6. http://python.org/ — Python Programming Language — Official Website. 2013.
7. https://www.djangoproject.com/ — Django. The Web framework for perfectionists with deadlines. 2013.
8. http://wiki.apparmor.net — AppArmor security wiki project. 2013.
9. http://www.linux.com/learn/docs/man/4047-setrlimit2 — Linux Programmer's Manual (2). getrlimit, setrlimit — get/set resource limits. 2008.
10. http://www.openstack.org/ — OpenStack. Open source cloud computing software. 2013.
11. https://www.simple-talk.com/sql/database-administration/database-design-a-point-in-time-architecture/ — Database Design: A Point in Time Architecture. 2007.
12. http://www.celeryproject.org/ — Celery. Distributed Task Queue. 2012.
13. http://memcached.org/ — Memcached. A distributed memory object caching system. 2012.

УДК 519.718

О БАЗИСАХ С КОЭФФИЦИЕНТОМ НЕНАДЁЖНОСТИ 1[1]

А. В. Васин

Рассматривается реализация булевых функций схемами из ненадёжных функциональных элементов в произвольном полном базисе B. Предполагается, что все элементы схемы независимо друг от друга с вероятностью $\varepsilon \in (0, 1/2)$ подвержены инверсным неисправностям на выходах. Найдено множество G функций, таких, что для почти всех функций ненадёжность асимптотически оптимальных по надёжности схем в базисе B, содержащем функции множества G, равна ε (при $\varepsilon \to 0$).

Ключевые слова: *ненадёжные функциональные элементы, асимптотически оптимальные по надёжности схемы, инверсные неисправности на выходах элементов.*

Рассматривается реализация булевых функций схемами [1] из ненадёжных функциональных элементов в произвольном полном конечном базисе B. Предполагаем, что все элементы схемы независимо друг от друга с вероятностью $\varepsilon \in (0, 1/2)$ подвержены инверсным неисправностям на выходах. Эти неисправности характеризуются тем, что в исправном состоянии функциональный элемент реализует приписанную ему булеву функцию ψ, а в неисправном — функцию $\overline{\psi}$. Считаем, что схема S из ненадёжных элементов реализует булеву функцию $f(x_1, x_2, \ldots, x_n)$, если при поступлении на входы схемы двоичного набора $a = (a_1, a_2, \ldots, a_n)$ при отсутствии неисправностей на выходе схемы S появляется значение $f(a)$.

Ненадёжностью $P(S)$ схемы S назовем максимальную вероятность ошибки на выходе схемы S при всевозможных входных наборах схемы. Надёжность схемы S равна $1-P(S)$. Пусть $P_\varepsilon(f) = \inf P(S)$, где инфимум берется по всем схемам S из ненадёж-

[1]Работа выполнена при финансовой поддержке РФФИ, проект № 12-01-31340.