МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

УДК 004.75

РАЗРАБОТКА БЕЗОПАСНОГО ПРОТОКОЛА РАСПРЕДЕЛЁННОЙ СИСТЕМЫ ПРОВЕДЕНИЯ СОРЕВНОВАНИЙ CTF

Н. И. Анисеня

Национальный исследовательский Томский государственный университет, г. Томск, Россия

Работа посвящена разработке математического метода проведения соревнований Capture the Flag, основанных на решении заданий при угрозе DDoS-атак на сервер организаторов. Предлагается распределённый протокол проведения соревнований, который перекладывает часть функций организаторов на участников. Участники соревнования конкурируют друг с другом и не хотят помогать командам-соперникам, поэтому к защищённости протокола предъявляются высокие требования. Получен протокол, удовлетворяющий поставленным требованиям, рассмотрены атаки, исследована устойчивость протокола к ним, предложены модификации. Описаны возможные направления дальнейших исследований в данной области.

Ключевые слова: *распределённые протоколы, защищённые вычисления, отказоустойчивые системы.*

DOI 10.17223/20710410/28/7

DEVELOPING SAFE PROTOCOL FOR DISTRIBUTED TASK-BASED CTF HOLDING SYSTEM

N. I. Anisenya

National Research Tomsk State University, Tomsk, Russia

E-mail: anisenya@gmail.com

The paper is devoted to mathematical method of holding task-based CTF contests under threat of DDoS attack against organizers' servers. For the purpose, a decentralized protocol is proposed where a part of organizer role is distributed among participants. There are some important security requirements to the protocol due to the competitive nature of CTF contests. The result protocol meets these requirements. Its stability against considered attacks is researched. Directions of further research are described.

Keywords: *distributed protocol, secure computation, fault-tolerant system.*

Введение

Соревнования по компьютерной безопасности Capture The Flag (CTF) [1] в последнее время стали очень популярны. Соревнования CTF имеют различные форматы

проведения [2]: задания на соревнованиях формата Jeopardy зачастую требуют детального ознакомления с новыми уязвимостями, а соревнования в формате Attack-Defense, помимо навыков нападения, дают практический опыт по защите компьютерных систем, что делает чрезвычайно полезным участие в подобных соревнованиях студентов, обучающихся по связанным с компьютерной безопасностью специальностям. Соревнования CTF нередко проводятся в рамках конференций, посвященных компьютерной безопасности (DEFCON [3], PHDAYS [4]), зачастую являясь их главным элементом (RuCTF [5]).

Команда по компьютерной безопасности SiBears Томского государственного университета имеет собственный сервис Blackbox [6, 7] для проведения соревнования CTF формата Jeopardy. С использованием Blackbox проведены множество соревнований, среди которых SiBCTF [8] и отборочные RuCTF 2012 Quals проводились на международном уровне.

Постоянно растущий интерес к CTF-движению привёл к тому, что подобными соревнованиями стали интересоваться злоумышленники. В 2012 г. на отборочных соревнованиях RuCTF 2012 Quals имел место инцидент, когда после дисквалификации за нарушения одной из команд на сервер организаторов была осуществлена DDoS-атака. Её результатом стала недоступность сервера соревнования на некоторое время, в течение которого организаторами настраивалась фильтрация соединений по белому списку IP-адресов, присылаемых честными участниками.

Использованный подход не обладает достаточной гибкостью: командам необходимо заранее знать IP-адреса своих участников, что в случае динамических адресов не всегда возможно, а добавление новых адресов в список осуществляется организаторами вручную. В данной работе предлагается способ борьбы с подобными ситуациями, который заключается в отказе от централизованного взаимодействия участников через сервер организаторов. Этот способ подразумевает распределение части роли организаторов между участниками соревнования, что повышает требования к защищённости вычислений. Рассматриваются атаки, предлагаются улучшения протокола, а также рассматриваются возможные направления дальнейших исследований.

1. Цель, задача и актуальность

Целью работы является предложение математического способа обеспечения доступности соревнования CTF, проводимого в формате Jeopardy, при угрозе DDoS-атаки на организаторов.

Под *централизованным сетевым взаимодействием*, или просто *централизованным взаимодействием*, участников будем понимать такое их взаимодействие, которое полагается на некоторый известный всем участникам узел сети — посредника, имеющего отличную от прочих участников и незаменимую по отношению к ним роль.

Злоумышленником назовём нечестного участника соревнования, который преследует хотя бы одну из следующих целей:

1) нарушение работоспособности системы, с высокой вероятностью приводящее к невозможности участия в соревновании всех участников;
2) нарушение работоспособности системы, с высокой вероятностью приводящее к искажению собственных результатов;
3) нарушение работоспособности системы, с высокой вероятностью приводящее к искажению результатов другого конкретного участника.

Активным участием некоторого узла назовём такое его поведение в сети, при котором он отправляет в сеть данные.

Для достижения указанной цели ставится следующая задача: разработать протокол распределённого проведения соревнований CTF, основанных на решении заданий. Требования к протоколу:

1) в результате работы протокола должна формироваться таблица результатов, позволяющая восстановить очерёдность получения ответов;
2) протокол не должен требовать активного участия организаторов во время проведения соревнования;
3) протокол не должен полагаться на централизованное взаимодействие участников во время проведения соревнования;
4) протокол должен позволять проводить соревнование даже при отключении большого количества участников;
5) протокол должен позволять проводить соревнование даже при большом количестве злоумышленников (нечестных участников).

Поскольку данные в распределённой сети распространяются с ощутимой задержкой, определим следующие отношения для значений времени.

Два значения времени t_1 и t_2 равны с учётом временного окна w, если $|t_1 - t_2| \leqslant w$; обозначим $t_1 =_w t_2$.

Значение времени t_1 предшествует значению времени t_2 с учётом временного окна w, если $t_2 - t_1 > w$; обозначим $t_1 <_w t_2$.

Если $t_1 <_w t_2$ либо $t_1 =_w t_2$, то будем записывать этот факт следующий образом: $t_1 \leqslant_w t_2$.

При разработке протокола не рассматривались следующие ситуации и проблемы:

1) проблемы подготовительного этапа (регистрации команд);
2) ситуация распада графа сети на компоненты связности;
3) недостаточная точность временных расчетов с учётом выбранного временного окна w.

Подход с использованием ресурсов пользователей для поддержания работы единой системы давно применяется в различных сферах, где требуются независимые от доверенной стороны («центра») передача данных, их хранение и обработка. Исследования подобных методов с целью применения в конкурентных средах, то есть в том случае, когда участники распределённой системы не заинтересованы в помощи друг другу, видится важной задачей, актуальной для различных областей деятельности.

2. Начальные условия

Ввиду отсутствия единого сервера участники должны иметь возможность самостоятельно проверять правильность полученного ответа на задание. После того как правильный ответ получен, участник должен передать остальным доказательство получения ответа, не раскрывая самого ответа. Помимо доказательства знания ответа, участник должен предоставить подтверждение того, что ответ получен в определённое время.

Сеть соревнования представляет собой оверлейную p2p-сеть (рис. 1), в которой каждая команда-участник (в том числе и команда организаторов) соединена с некоторыми другими командами-участниками (соседями) напрямую, например посредством TCP-соединения.

Предполагается, что каждый участник сети имеет список идентификаторов команд — открытых ключей. Соответствие идентификаторов командам знают только организаторы.

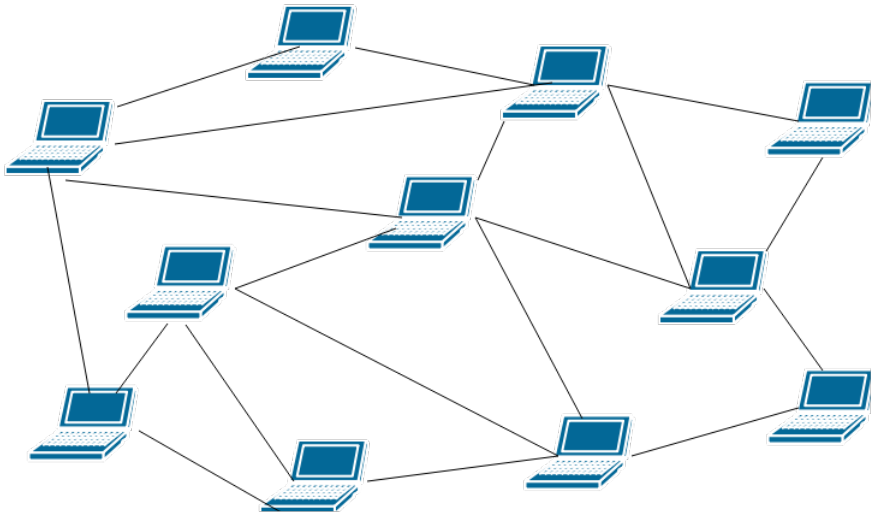

Рис. 1. Иллюстрация p2p-сети

Передача сообщения от Алисы к Бобу в общем случае осуществляется через промежуточные узлы, назовём их *посредниками*. Для отправки сообщения Бобу Алиса сначала шифрует его на открытом ключе Боба. В шифруемое сообщение включается преамбула, позволяющая определить, что процесс расшифрования прошел успешно. Затем Алиса посылает сообщение в сеть, отправляя его каждому своему соседу. Каждый сосед после получения сообщения от Алисы пытается его расшифровать. Очевидно, что успешно расшифровать сообщение способен только Боб, поэтому в случае неуспешного расшифрования текущий узел передаёт полученное сообщение всем своим соседям. Так происходит до тех пор, пока сообщение не будет доставлено Бобу. По аналогии с Freenet [9], для предотвращения бесконечной передачи сообщений узлы не передают одни и те же сообщения дважды.

Описанный способ организации сети соревнования обеспечивает анонимность участников, которая, в свою очередь, с одной стороны, не позволяет участникам блокировать передачу сообщений от или для известной команды, с другой — делает неотличимым от остальных узлов участие в сети команды организаторов.

3. Маршрутизация в условиях конкуренции

3.1. Проблема отказа от посредничества

Участники сети — команды — конкурируют друг с другом и не заинтересованы в доставке чужих сообщений. Для доставки сообщения Бобу Алисе потребуется передать его через посредников. Для защиты конфиденциальности и целостности сообщения оно шифруется на известном Алисе открытом ключе Боба. Однако нет гарантий, что сообщение будет доставлено посредниками. Далее предлагается способ сделать невозможным отказ от посредничества участника сети. Иными словами, если участник сети отказывается поддерживать её работу — выступать в роли посредника, то он не может получить к ней доступ.

3.2. Луковая маршрутизация в сети Tor

Сеть Tor [10] разрабатывалась с целью анонимного доступа к веб-ресурсам. Клиентское приложение Tor представляет собой маршрутизатор, работающий как прокси-сервер. После подключения к сети маршрутизатор получает список IP-адресов узлов сети Tor. Для отправки запроса на веб-ресурс маршрутизатор случайным образом

выбирает нескольких (по умолчанию трёх) посредников из списка узлов, выстраивая таким образом цепочку прокси-серверов. Сообщение шифруется на ключах узлов, состоящих в цепочке, в порядке, обратном порядку следования сообщения: для отправки сообщения m от Алисы к Бобу через посредников M_1, M_2, M_3 оно будет зашифровано следующим образом:

$$E_{M_1}(E_{M_2}(E_{M_3}(m, ip_{Bob}), ip_{M_3}), ip_{M_2}), \tag{1}$$

где ip_x — это IP-адрес узла x. Полученную «луковицу» Алиса отправляет первому посреднику M_1. Получив сообщение (1), посредник M_1 расшифровывает его, как бы снимая свой слой луковицы, и передаёт следующему посреднику M_2 сообщение вида

$$E_{M_2}(E_{M_3}(m, ip_{Bob}), ip_{M_3}). \tag{2}$$

Так продолжается до тех пор, пока Боб не получит исходное сообщение m. Ответ от Боба проходит по той же цепочке в обратном направлении. Разработчиками Tor описанному способу маршрутизации дано название *луковая маршрутизация* (рис. 2).

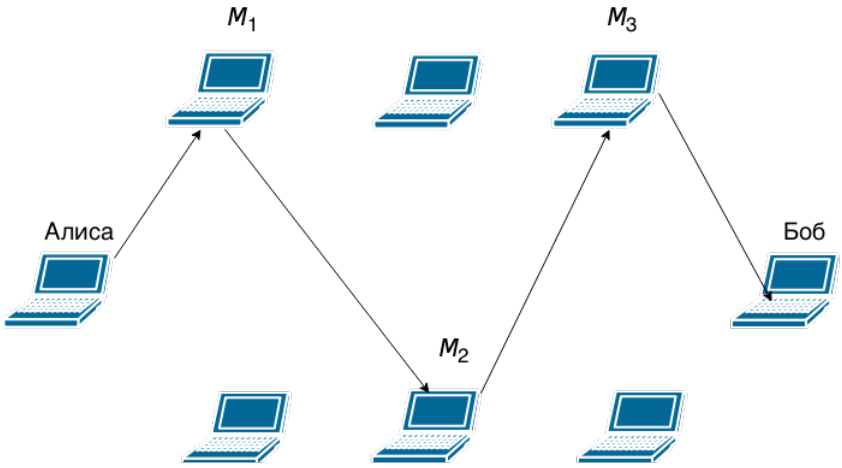

Рис. 2. Пример луковой маршрутизации сети Tor

Анонимность источника запроса обеспечивается за счёт того, что с точки зрения посредников сообщения (1) и (2) не отличаются. Если получатель, в свою очередь, тоже является участником сети Tor, то так же обеспечивается и его анонимность. Однако сеть Tor полагается на честность участников, что не подходит для конкурентной среды. Поэтому для решения поставленной задачи данный способ маршрутизации требует некоторых модификаций.

3.3. Безотказная луковая маршрутизация

Как сказано в п. 2, IP-адреса команд неизвестны, обращение к командам происходит по идентификаторам через посредников. Анонимность отправителя и получателя обеспечивается способом передачи сообщений и с помощью шифрования, которое защищает также целостность и конфиденциальность сообщения.

Для защиты доступности сообщений предлагается использовать принцип луковой маршрутизации сети Tor. Назовём этот способ *безотказной луковой маршрутизацией*.

Пусть Алиса отправляет Бобу сообщение m, M — посредник (рис. 3). Пример безотказной луковой маршрутизации проиллюстрирован на рис. 3. Для упрощения записи

будем считать, что E_X — это шифрование на открытом ключе пользователя X. Алиса шифрует сообщение следующим образом:

$$E_{Bob}(E_M(E_{Bob}(m))) \tag{3}$$

и отправляет получившуюся «луковицу» Бобу.

Получив сообщение (3), Боб расшифрует свой слой и передаст посреднику M следующее сообщение:

$$E_M(E_{Bob}(m)). \tag{4}$$

Посредник, повторив действия Боба, отправит в сеть сообщение вида

$$E_{Bob}(m). \tag{5}$$

Когда сообщение (5) дойдёт до Боба, он его прочитает и сможет ответить аналогичным образом, выбрав в общем случае другого посредника.

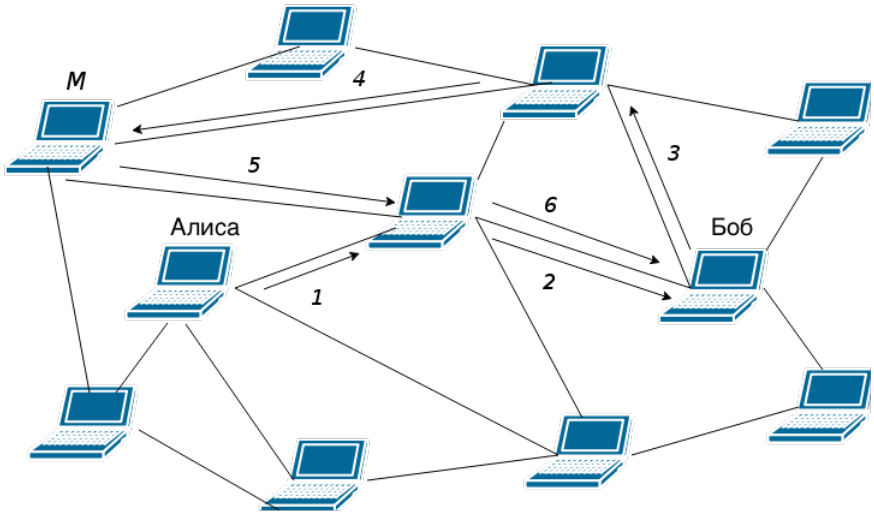

Рис. 3. Пример безотказной луковой маршрутизации

Рассмотрим, как обеспечивается невозможность отказа от посредничества. По полученному сообщению (3) Боб не сможет узнать ни источник сообщения, ни его получателя, а также не сможет определить, предназначено ли в конечном итоге это сообщение самому Бобу. Поэтому если Боб хочет получать сообщения из сети, ему придётся выступать в роли посредника для каждой полученной луковицы. С точки зрения любого посредника M, за исключением Боба, сообщения (3), (4) и (5) трактуются одинаково: «неизвестно, что внутри, но, возможно, это мне», поэтому сообщения передаются дальше. Посредники, не участвующие в образовании луковицы, вынуждены передавать сообщения по этой же причине.

Предложенный способ маршрутизации имеет существенное ограничение: он может быть применён только в тех случаях, когда каждый участник сети вынужден постоянно ожидать сообщения. Действительно, если по какой-либо причине участник сети уверен, что в данный момент не ожидает получения данных, то он может отбрасывать поступившие для передачи сообщения.

4. Протокол проведения соревнования

4.1. Обозначения и определения

Пусть sign — некоторый алгоритм цифровой подписи; verify — соответствующий ему алгоритм проверки подписи; $U\{ID_1, \ldots, ID_n\}$ — *множество идентификаторов* (открытых ключей) участников. Каждому открытому ключу ID_i соответствует закрытый ключ $\widehat{ID_i}$, $i \in \{1, \ldots, n\}$, то есть для любого сообщения x и любого $i \in \{1, \ldots, n\}$

$$\mathrm{verify}_{ID_i}(\mathrm{sign}_{\widehat{ID_i}}(x)) = 1.$$

Определим ответы на задания. Пусть $g_1, \ldots, g_m$ — открытые ключи (проверки подписи), $f_1, \ldots, f_m$ — соответствующие им закрытые ключи (подписания), то есть для любого $i \in \{1, \ldots, m\}$ и для любого сообщения x

$$\mathrm{verify}_{g_i}(\mathrm{sign}_{f_i}(x)) = 1.$$

Ключ f_i будем считать *ответом* на задание с номером i; g_i назовём *ключом проверки ответа* на задание с номером i.

Для того чтобы подтвердить получение ответа на задание в конкретное время конкретной командой, будем использовать метки времени.

Меткой времени от участника A для данных y, указывающей на момент времени t, будем считать набор значений $(t, y, \mathrm{sign}_{K_A}(t, y))$, где K_A — закрытый ключ (подписания) пользователя A.

Согласно требованиям к протоколу, нельзя полагаться на сервер выдачи меток времени, поэтому необходимо использовать распределённый подход — команды сами должны выдавать друг другу метки времени, подтверждающие получение ответа в конкретное время конкретным участником. За основу взят подход, описанный в [11]. Введём псевдослучайную функцию $G : \mathbb{N} \to U' \subset U$, $|U'| = t < n$.

Функция $G(y)$ используется в протоколе для определения множества участников, которые должны поставить метку времени под данными y. Для подтверждения получения данных y в указанное время требуется получить хотя бы k меток времени. Параметры n, t и k находятся в отношении $k < t < n$.

4.2. Протокол

Пусть на момент начала соревнования имеется сеть участников, описанная в п. 2, в которую команда организаторов входит как равноправный участник. Полагаем, что в этой сети для передачи сообщений используется безотказная луковая маршрутизация, описанная в п. 3.3. Каждый участник на момент начала соревнования имеет следующее:

1) алгоритмы sign и verify;
2) алгоритм симметричного шифрования на ключе E_x;
3) множество идентификаторов участников U;
4) ключи проверки ответов $g_1, \ldots, g_m$;
5) псевдослучайную функцию $G(y)$ с параметрами k, t;
6) зашифрованный набор заданий.

Началом соревнования считается момент рассылки организаторами ключа для расшифрования списка заданий, после чего команда организаторов перестаёт принимать активное участие.

Пусть некоторый участник u с идентификатором ID_u получил ответ f на задание. Рассмотрим протокол, согласно которому должен действовать участник u:

1) $y = \mathrm{sign}_f(t_u, ID_u)$, t_u — текущее время пользователя u.
2) Для всех пользователей с идентификатором $id \in G(y)$:
   а) $u \to id : z = E_x(y, t_u, ID_u)$, x — сеансовый ключ;
   б) $id \to u : t_{id}, \mathrm{sign}_{\widehat{id}}(z, t_{id})$, t_{id} — текущее время пользователя с идентификатором id.
3) Каждому пользователю с идентификатором $id \in U$ выслать:
   t_u, ID_u, $y = \mathrm{sign}_f(t_u, ID_u)$, $z = E_x(y, t_u, ID_u)$, x, $\{t_i, \mathrm{sign}_{\widehat{id_i}}(z, t_i) : i \in G(y)\}$.

Соревнование завершается в установленное организаторами время. Таблица результатов, имеющаяся у организаторов в этот момент, считается итоговой.

4.3. Пояснения к протоколу

Функция $G(y)$ используется с целью сделать труднопредсказуемым множество U' допустимых для запроса меток времени участников. Избыток участников в множестве U' объясняется тем, что среди команд множества U' могут быть как неактивные участники, так и злоумышленники, которые не будут выдавать метки времени для подтверждения ответа. Таким образом создаётся «запас» в виде $(t - k)$ участников для каждого ответа.

На первом шаге значение y служит сразу для двух целей. Во-первых, y является аргументом функции $G(y)$ на шаге 2, то есть определяет множество пользователей U', от которых будут набираться метки времени для подтверждения ответа. Поскольку значение y зависит от ответа f, его невозможно посчитать заранее. Во-вторых, значение y зависит от идентификатора участника, решившего задание, а значит, это значение не может быть использовано другим участником.

Необходимость шифровать запрос на выдачу метки времени для подтверждения ответа (шаг 2а) возникает для того, чтобы исключить возможность отказывать конкретному участнику в выдаче метки времени. После публикации факта получения ответа на шаге 3 все участники смогут расшифровать запрос и провести необходимые проверки.

4.4. Алгоритм проверки ответа

При получении сообщения (в шаге 3) каждый участник должен выполнить ряд проверок, прежде чем добавить в свою таблицу результатов новую запись. Пусть t — текущее время, w — условленное временное окно, g — ключ проверки ответа f. Алгоритм проверки факта получения ответа f следующий:

1) расшифровать $D_x(z)$;
2) если $t \leqslant_w t_u$ или $\exists i\ (t \leqslant_w t_i)$, то проверка не пройдена;
3) если среди множества выдавших метку времени для подтверждения ответа существует идентификатор $id \notin G(y)$, то проверка не пройдена;
4) если $\mathrm{verify}_g(y) = 0$, то проверка не пройдена;
5) $j := 0$; для всех $i \in G(y)$ выполнить:
   если $\mathrm{verify}_{ID_i}(\mathrm{sign}_{\widehat{ID_i}}(z, t_i)) = 1$, то $j = j + 1$;
6) если $j < k$, то проверка не пройдена;
7) проверка пройдена.

5. Атаки и предлагаемые улучшения

5.1. Атака отказа от посредничества

Описание атаки

Атака реализует угрозу нарушения работоспособности системы, что с высокой вероятностью приводит к невозможности участия в соревновании всех участников. Для защиты от этой атаки используется безотказная луковая маршрутизация, однако не созданы условия для её применения (см. п. 3.3). Участник может отказываться передавать не предназначенные ему данные до тех пор, пока ему самому не потребуется ожидать подтверждения факта получения ответа.

Противодействие

Предлагается наполнить сеть трафиком с помощью рассылки сообщений следующего вида:

$$(t, id_B, \operatorname{sign}_{\widehat{id_A}}(t, id_B)). \tag{6}$$

Сообщение (6) назовём *разрешением на запрос подтверждения получения ответа*, или просто *разрешением* от участника A участнику B, в котором t — это время создания разрешения, id_B — идентификатор участника B. Подразумевается, что разрешения рассылаются каждым участником каждому участнику.

К шагу 2а протокола следует добавить отправку разрешения $(t, ID_u, \operatorname{sign}_{\widehat{id}}(t, ID_u))$ от пользователя с идентификатором id пользователю u. Пользователь с идентификатором id выдаёт подтверждение только при наличии разрешения. Разрешение также должно быть в публикуемом факте получения ответа на шаге 3 протокола. Проверку наличия необходимых разрешений следует добавить в алгоритм проверки.

Таким образом, все участники постоянно будут ожидать получения разрешения от всех пользователей, так как предсказать, какие именно разрешения пригодятся, невозможно. Для того чтобы необходимость получения разрешений не прекращалась, можно ввести время жизни разрешения, то есть считать устаревшие разрешения недействительными.

5.2. Сговор с целью невыдачи метки времени для подтверждения ответа

Описание атаки

Атака реализует угрозу нарушения работоспособности системы, что с высокой вероятностью приводит к невозможности участия в соревновании конкретного участника либо всех участников. Участники не мотивированы выдавать метки времени для подтверждения чужих ответов (шаг 2б протокола). Если для некоторого $G(y)$ найдутся хотя бы $(t - k - 1)$ участников, которые отказались выдавать метку времени, то факт получения ответа не будет сформирован. Такая ситуация может сложиться не только в результате сговора злоумышленников, но и в результате отключения команд по иным причинам.

Противодействие

Протокол полагается на то, что большинство участников действует добросовестно. Пусть f — это количество злоумышленников в сети, $f < n$. Вероятность того, что

данная атака будет успешной, следующая:

$$p = 1 - \frac{\sum\limits_{i=0}^{t-k+1} \binom{n-f}{k+i}\binom{f}{t-k-i}}{\binom{n}{t}}.$$

В табл. 1 приведены вероятности для $t = [n/2]$, $k = [t/2]$, $n \in \{50, 100, 250, 500\}$.

Таблица 1

Вероятности успешной атаки п. 5.2

$n=50, t=25, k=12$		$n=100, t=50, k=25$		$n=250, t=125, k=62$		$n=500, t=250, k=125$	
f	p	f	p	f	p	f	p
14	0,0000	26	0,0000	69	0,0000	136	0,0000
15	0,0001	28	0,0000	74	0,0000	146	0,0000
16	0,0003	30	0,0000	79	0,0000	156	0,0000
17	0,0011	32	0,0000	84	0,0000	166	0,0000
18	0,0036	34	0,0001	89	0,0000	176	0,0000
19	0,0093	36	0,0008	94	0,0000	186	0,0000
20	0,0210	38	0,0035	99	0,0001	196	0,0000
21	0,0423	40	0,0121	104	0,0015	206	0,0000
22	0,0768	42	0,0338	109	0,0107	216	0,0008
23	0,1281	44	0,0791	114	0,0493	226	0,0123
24	0,1981	46	0,1579	119	0,1555	236	0,0895
25	0,2861	48	0,2742	124	0,3522	246	0,3274

5.3. Сговор с целью выдачи ложной метки времени конкретному участнику

Описание атаки

Атака может быть применена с целью нарушения работоспособности системы, что с высокой вероятностью приводит к искажению как собственных результатов, так и результатов другого конкретного участника. Если в множестве $G(y)$ окажется меньше k активных честных участников и хотя бы k участников вступят в сговор, то они смогут выдать ложную метку времени.

Противодействие

Как и в предыдущем случае, противодействие данной атаке полагается на то, что большинство участников сети — честные. Вероятность того, что атака окажется успешной по отношению к некоторому участнику, равна

$$p = \frac{\sum\limits_{i=0}^{t-k+1} \binom{f}{k+i}\binom{n-f}{t-k-i}}{\binom{n}{t}}.$$

В табл. 2 приведены вероятности для $t = [n/2]$, $k = [t/2]$, $n \in \{50, 100, 250, 500\}$.

5.4. Прочие атаки

Поскольку команды представлены лишь своим идентификатором, злоумышленник, помимо своей команды, может зарегистрировать несколько фиктивных и устроить сговор самостоятельно. Стоит заметить, что фиктивные команды могут вводиться и честными участниками с целью «уравновесить» злоумышленников. Можно пойти по пути

Таблица 2

Вероятности успешной атаки п. 5.3

$n = 50, t = 25, k = 12$		$n = 100, t = 50, k = 25$		$n = 250, t = 125, k = 62$		$n = 500, t = 250, k = 125$	
f	p	f	p	f	p	f	p
14	0,0018	26	0,0000	69	0,0000	136	0,0000
15	0,0061	28	0,0000	74	0,0000	146	0,0000
16	0,0161	30	0,0000	79	0,0000	156	0,0000
17	0,0359	32	0,0001	84	0,0000	166	0,0000
18	0,0699	34	0,0007	89	0,0000	176	0,0000
19	0,1218	36	0,0032	94	0,0001	186	0,0000
20	0,1934	38	0,0114	99	0,0009	196	0,0000
21	0,2836	40	0,0328	104	0,0073	206	0,0000
22	0,3881	42	0,0779	109	0,0370	216	0,0014
23	0,5000	44	0,1569	114	0,1265	226	0,0193
24	0,6112	46	0,2737	119	0,3063	236	0,1221
25	0,7139	48	0,4207	124	0,5503	246	0,3942

Bitcoin [12], в котором используется принцип защиты от злоупотребления услугами proof-of-work, предложенный в [13], и сделать участие в соревновании зависимым от решения сложной вычислительной задачи.

Отказ от рассылки разрешений можно приравнять к отказу от выдачи метки времени для подтверждения. Сюда же можно отнести рассылку разрешений с неверным временем. Точно так же включение неправильного времени в подтверждение ответа (без сговора) можно отнести к невыдаче метки времени для подтверждения ответа.

Заключение

В работе предложен удовлетворяющий поставленным требованиям протокол для проведения соревнований CTF, основанных на решении заданий.

В продолжение решения задачи проведения децентрализованных соревнований планируется рассмотрение проблем подготовительного этапа, построения сети, а также доработка предложенного протокола таким образом, чтобы снизить зависимость успешности проведения соревнований от добросовестности участников. Впоследствии планируется обобщить полученные результаты и описать протокол для решения абстрактной задачи.

ЛИТЕРАТУРА

1. *Колегов Д. Н., Чернушенко Ю. Н.* О соревнованиях CTF по компьютерной безопасности // Прикладная дискретная математика. 2008. № 2. С. 81–83.
2. `https://ctftime.org/ctf-wtf/` — CTFtime.org / All about CTF (Capture The Flag). 2014.
3. `https://www.defcon.org/` — DEF CON Hacking Conference. 2014.
4. `http://www.phdays.ru/` — PHDays — Positive Hack Days. 2014.
5. `http://ructf.org/` — RuCTF. 2014.
6. *Анисеня Н. И., Стефанцов Д. А., Торгаева Т. А.* Сервис BlackBox для проведения соревнований по защите компьютерной информации Capture the Flag // Прикладная дискретная математика. Приложение. 2013. № 6. С. 52–56.
7. `http://blackbox.sibears.ru/` — Blackbox. 2014.
8. `http://sibctf.ru/` — SiBCTF. 2014.
9. *Clarke I., Sandberg O., Wiley B., and Hong T. W.* Freenet: a distributed anonymous information storage and retrieval system // Intern. Workshop on Designing Privacy Enhancing

Technologies: Design Issues in Anonymity and Unobservability. N.Y.: Springer Verlag, 2001. P. 46–66.

10. `https://www.torproject.org/` — Tor Project. 2014.
11. *Haber S. and Stornetta W. S.* How to time-stamp a digital document // J. Cryptology. 1991. No. 3. P. 99–111.
12. `http://bitcoin.org/` — Bitcoin — Open source P2P money. 2014.
13. *Dwork C. and Naor M.* Pricing via processing or combatting Junk Mail // Proc. CRYPTO'92. Berlin, Heidelberg: Springer, 1993. P. 139–147.

REFERENCES

1. *Kolegov D. N., Chernushenko Yu. N.* O sorevnovaniyakh CTF po komp'yuternoy bezopasnosti [About the CTF — computer security competitions]. Prikladnaya Diskretnaya Matematika. 2008, no. 2, pp. 81–83. (in Russian)
2. `https://ctftime.org/ctf-wtf/` — CTFtime.org / All about CTF (Capture The Flag). 2014.
3. `https://www.defcon.org/` — DEF CON Hacking Conference. 2014.
4. `http://www.phdays.ru/` — PHDays — Positive Hack Days. 2014.
5. `http://ructf.org/` — RuCTF. 2014.
6. *Anisenya N. I., Stefantsov D. A., Torgaeva T. A.* Servis BlackBox dlya provedeniya sorevnovaniy po zashchite komp'yuternoy informatsii Capture the Flag [The BlackBox service for hosting Capture The Flag computer security competitions]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2013, no. 6, pp. 52–56. (in Russian)
7. `http://blackbox.sibears.ru/` — Blackbox. 2014.
8. `http://sibctf.ru/` — SiBCTF. 2014.
9. *Clarke I., Sandberg O., Wiley B., and Hong T. W.* Freenet: a distributed anonymous information storage and retrieval system. Intern. Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability, N.Y., Springer Verlag, 2001, pp. 46–66.
10. `https://www.torproject.org/` — Tor Project. 2014.
11. *Haber S. and Stornetta W. S.* How to time-stamp a digital document. J. Cryptology, 1991, no. 3, pp. 99–111.
12. `http://bitcoin.org/` — Bitcoin — Open source P2P money. 2014.
13. *Dwork C. and Naor M.* Pricing via processing or combatting Junk Mail. Proc. CRYPTO'92. Berlin, Heidelberg, Springer, 1993, pp. 139–147.