ОБМАННЫЕ СИСТЕМЫ ДЛЯ ЗАЩИТЫ ИНФОРМАЦИОННЫХ РЕСУРСОВ В КОМПЬЮТЕРНЫХ СЕТЯХ*

И. В. Котенко, М. В. Степашкин

Санкт-Петербургский институт информатики и автоматизации РАН
199178, Санкт-Петербург, 14-я линия В.О., д. 39
<ivkote@spiiras.nw.ru>, <stepashkin@computer.edu.ru>

УДК 681.3.06
И. В. Котенко, М. В. Степашкин. **Обманные системы для защиты информационных ресурсов в компьютерных сетях** // Труды СПИИРАН, Вып. 2, т. 1. — СПб.: СПИИРАН, 2004.
Аннотация. *Одними из перспективных механизмов, дополняющих существующие механизмы защиты информационных ресурсов в компьютерных сетях, являются механизмы введения в заблуждение (обмана) нарушителей информационной безопасности. Эти механизмы предназначены для повышения защищенности целевых информационных систем за счет привлечения злоумышленников к ложным информационным целям, введения в заблуждение, идентификации их действий и разоблачения. Механизмы введения в заблуждение нарушителей реализуются посредством разработки и использования обманных систем (ОбС) или компонентов, называемых также ложными информационными системами, имитаторами информационных систем или ловушками. В статье определены назначение, функции и структура ОбС, дана их классификация, представлен предлагаемый подход к построению перспективной ОбС, предложены схемы реализации замаскированного противодействия сетевым атакам и архитектура разработанного прототипа ОбС, приведено краткое описание экспериментов, проводимых с прототипом.* — Библ. 43 назв.

UDC 681.3.06
I. V. Kotenko, M. V. Stepashkin. **Deception systems for protection of information resources in computer networks** // SPIIRAS Proceeding, Issue 2, vol. 1. — SPb.: SPIIRAS, 2004.
Abstract. *Ones of the perspective mechanisms supplementing existing mechanisms of information resources protection in computer networks are malefactors' deception mechanisms. These mechanisms are intended for increasing the security of target information systems on the base of attraction of malefactors to false information goals, deceptions, identification of their actions and disclosure. Malefactors' deception mechanisms are realized by means of development and usage of deception systems (DS) or components named also false information systems, simulators of information systems, traps or honeypots. The paper defines the destination, functions and structure of DS, presents their classification, submits the offered approach to development of perspective DS, offers the schemes of realization of disguised counteraction against network attacks and architecture of the DS prototype developed, describes the experiments spent with the prototype.* — Bibl. 43 items.

1. Введение

В настоящее время все более актуальной становится задача защиты информационных ресурсов компьютерных сетей от атак со стороны внешних и внутренних нарушителей [1]. Для решения данной задачи необходимо не только предупреждать, блокировать, обнаруживать и реагировать на действия нарушителей, но и отвлекать их от основных целей, заманивая на ложные информационные объекты, производить сбор информации о приемах, тактике и мотивации злоумышленников, осуществлять их идентификацию и разоблачение.

* Работа выполнена при частичной поддержке программы фундаментальных исследований Отделения информационных технологий и вычислительных систем РАН и РФФИ.

Для выполнения этих подзадач могут быть использованы обманные системы (ОбС), называемые также ложными информационными системами, имитаторами информационных систем или системами-ловушками [2–6]. Основными функциями таких систем являются привлечение и удержание внимания злоумышленников на ложных информационных целях, введение злоумышленников в заблуждение, обнаружение и фиксация действий нарушителей, их контроль, а также сбор и агрегация данных о действиях нарушителей из различных источников.

ОбС представляют собой программно-аппаратные средства обеспечения информационной безопасности, реализующие функции сокрытия и камуфляжа защищаемых информационных ресурсов, а также дезинформации нарушителей. С помощью фиксации и сбора данных, межсетевого экранирования, обнаружения вторжений и обмана нарушителей (на основе имитации ложных целей, уязвимых для нападения), а также других механизмов эти системы позволяют в реальном времени выявлять атаки, направлять их по ложному следу, ограничивать их распространение, идентифицировать нарушителей, исследовать их действия и определять намерения [1–8].

В статье определены место и роль механизма введения в заблуждение нарушителя, рассмотрено состояние исследований в данной области, представлен подход к построению ОбС, охарактеризованы функции и структура перспективной ОбС, дано представление о реализуемых схемах функционирования ОбС, описан реализуемый прототип ОбС и проводимые с прототипом эксперименты.

2. Жизненный цикл инцидента безопасности. Место и роль механизмов обмана нарушителя

Рассмотрим последовательность фаз реализации атаки на компьютерную сеть, отражаемых через жизненный цикл инцидента безопасности, и механизмы защиты информации, необходимые для реализации на каждой фазе выполнения атаки. Данные механизмы защиты и фазы жизненного цикла инцидента безопасности представлены на рис. 1 [9]. Основными фазами жизненного цикла любого инцидента безопасности являются: (1) предупреждение угрозы безопасности, (2) реализация угрозы и возникновение инцидента, (3) нанесение ущерба и (4) восстановление ресурсов защищаемой системы после нанесения ущерба.

Рис.1. Цикл инцидента безопасности

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Угроза — это возможная опасность (потенциальная или реально существующая) совершения какого-либо деяния (действия или бездействия), направленного против объекта защиты (информационных ресурсов), наносящего ущерб собственнику, владельцу или пользователю, проявляющегося в опасности искажения и потери информации.

Угрозы могут быть преднамеренными, являющимися следствием умышленных (преднамеренных) действий людей, и непреднамеренными, т.е. вызванными ошибками человека, сбоями и отказами в работе технических и программных средств, или стихийными бедствиями.

Реализация одной или нескольких преднамеренных угроз представляет собой атаку. При этом атака является попыткой преодоления защиты автоматизированной системы, степень успеха которой зависит от уязвимости системы и эффективности защитных мер.

Основными видами угроз являются угрозы конфиденциальности, целостности и доступности. Угрозы конфиденциальности направлены на разглашение информации, т.е. в результате реализации этих угроз информация становится известной лицу, которое не должно иметь к ней доступа. Для обозначения этого явления используется термин “несанкционированный доступ” (НСД), под которым понимается доступ к информации, нарушающий установленные правила разграничения доступа. Угрозы целостности представляют собой любое искажение или изменение неуполномоченным на это действие лицом хранящейся в вычислительной системе или передаваемой информации. Целостность информации может быть нарушена как злоумышленником, так и в результате объективных (неумышленных) воздействий со стороны среды эксплуатации системы. Угрозы нарушения доступности (отказа в обслуживании) направлены на создание ситуаций, когда в результате преднамеренных или непреднамеренных действий снижается работоспособность вычислительной системы либо ее ресурсы становятся недоступными.

Можно выделить следующие группы механизмов защиты, реализуемых на различных фазах жизненного цикла инцидента безопасности: (1) предупреждение, (2) ослабление, (3) введение в заблуждение (обман) нарушителя, (4) обнаружение, (5) реагирование, (6) нейтрализация и устранение последствий и (7) оценивание инцидента и принятых мер.

Прежде всего, реализуются механизмы предупреждения. Они препятствуют реализации угрозы, преграждая нарушителю путь к защищаемым информационным ресурсам. Примером средств, осуществляющих механизмы предупреждения, являются средства защиты периметра компьютерной сети (например, межсетевые экраны). Другой пример — средства сканирования известных уязвимостей и последующего их устранения посредством установки патчей или изменения параметров конфигурации.

Механизмы ослабления — это механизмы, которые выполняются заранее с целью уменьшения возможного ущерба от реализации угрозы. Примерами механизмов ослабления являются применение резервируемых систем, ограничение пропускной способности к защищаемой сети, регулярное осуществление резервного копирования важных ресурсов и др.

Механизмы введения в заблуждение (обмана) нарушителей представляют собой специальный тип механизмов защиты, предназначенных для навязывания нарушителям ложной информации с целью уменьшения возможности реализации угроз (вторжения), облегчения обнаружения атак, замедления дейст-

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

вий по реализации угроз и исследования намерений, стратегий и средств нарушителей.

Механизмы предупреждения, ослабления и введения в заблуждение уменьшают вероятность и степень воздействия нарушителей, но не исключают возможное возникновение вторжения. Поэтому необходима также реализация механизмов обнаружения. Вторжения должны быть обнаружены как можно раньше. Обнаружение атак на ранних стадиях их реализации позволит получить преимущество перед нарушителем, по крайней мере, во временном ресурсе. Это время может быть использовано для идентификации нарушителя и реализации более мощных механизмов предупреждения, ослабления и введения в заблуждение, что приведет к минимизации ущерба и повышению результативности реагирования на атаки.

После того, как вторжение обнаружено, должны быть реализованы механизмы реагирования на вторжение. Эти механизмы должны обеспечить блокирование повторной реализации угрозы. Они могут также включить идентификацию и трассировку действий нарушителя, сбор информации для административного и уголовного наказания нарушителя.

Когда вторжение приводит к нанесению ущерба целостности или доступности информации, следующий шаг в цикле инцидента безопасности защиты должен заключаться в нейтрализации и устранении последствий вторжения.

Состояние важных компонентов и ресурсов компьютерной сети должно быть восстановлено как можно скорее. Здесь велика роль механизмов ослабления, в частности использования резервных копий информационных ресурсов.

Заключительный шаг состоит в оценивании эффективности предпринятых мер защиты.

Следует отметить важность механизмов введения в заблуждение (обмана) нарушителей. Этому типу механизмов защиты уделялось недостаточное внимание, однако его реализация позволит существенно повысить эффективность защитных мер против внешних и внутренних вторжений.

3. Назначение, классификация, достоинства и недостатки использования обманных систем

По своему назначению выделим два класса информационных систем (ИС): *целевые ИС*, предназначенные для автоматизации необходимых функций организации, и *ложные ИС* (ОбС), служащие для имитации целевых ИС с целью введения в заблуждение (обмана) нарушителей и отвлечения их внимания от информационных ресурсов целевых ИС.

ОбС предназначены для реализации следующих о*сновных подцелей*:

(1) ограничение атак на целевые (критически важные) системы за счет отпугивания нарушителя и “принятия огня на себя” (следствием чего является снижение эффективности атак, в том числе замедление их реализации, или их полное блокирование; это может позволить вовремя среагировать на распространение вирусов, сетевых червей и т.п.);

(2) скрытное обнаружение (отслеживание) и исследование (оперативный анализ) атак и неавторизованной активности (издержки сокращаются за счет снижения числа ложных срабатываний, так как любой трафик, направленный на ОбС вероятнее всего содержит действия нарушителя);

(3) мониторинг случаев несанкционированного доступа к системе и ее использования не по назначению;

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

(4) реагирование на действия нарушителя с целью введения его в заблуждение.

ОбС могут обеспечить повышение безопасности ИС напрямую или косвенно. Непосредственное влияние ОбС на защищенность проявляется в усилении общей архитектуры защиты и конкретных механизмов защиты за счет перенесения внимания нарушителей с компонентов целевой системы на компоненты ложной, задействования межсетевого экранирования, обнаружения вторжений, реализации более эффективных механизмов реагирования на действия нарушителя и др. Косвенное влияние проявляется в раскрытии стратегий, средств и действий нарушителей для последующего усиления защитных механизмов.

Значение ОбС зависит от того, как они спроектированы, реализованы, как и где они используются. ОбС должны строиться таким образом, чтобы атака на них была наиболее привлекательна по тем или иным причинам для злоумышленника (наименее защищенная часть системы или кажущаяся привлекательность по информативности). ОбС могут имитировать отдельный протокол (SMTP, FTP, POP3, HTTP и т.п.), отдельную рабочую станцию или сервер под управлением операционной системы (ОС) и целые сети. ОбС могут быть реализованы на одном или нескольких хостах и одновременно могут использоваться системы различных производителей. ОбС могут различаться по уровню интеграции в реальные сети. Они могут существовать, как отдельно от сетей, для защиты которых созданы, параллельно с ними (для отвлечения сил атакующих и изучения их методов), так и внутри этих сетей, что является наиболее эффективным и сложным в настройке и эксплуатации средством, которое позволяет отслеживать и пресекать вторжение изнутри.

Проведем *классификацию ОбС* по назначению и уровню взаимодействия с нарушителем.

По назначению выделяют два основных типа ОбС — “производственные” и исследовательские.

Производственные ОбС применяются для защиты ресурсов отдельных компьютеров и компьютерных сетей, в том числе снижения риска их компрометации. Как правило, данные ОбС легче реализовать, так как они обладают меньшей функциональностью, чем исследовательские ОбС. Вследствие простоты их гораздо сложнее использовать для атак на другие системы, однако они могут предоставить гораздо меньше информации о нарушителе.

Исследовательские ОбС применяются для изучения действий нарушителей, используемых ими стратегий и средств с целью построения более эффективных механизмов защиты. ОбС данного типа характеризуются гораздо более высоким уровнем взаимодействия с нарушителем, чем производственные. Они более сложны и используют не эмулируемые, а реальные ОС и приложения. Это позволяет получать больше информации о нарушителях. Однако более высокая функциональность приводит к большим затратам на их сопровождение и к большему риску их компрометации и использования против других систем. Фактически ОбС данного типа могут существенно снизить защищенность АС, в которых они развернуты.

Следует отметить, что различие между производственными и исследовательскими ОбС не является принципиальным. Зачастую одни и те же ОбС могут использоваться и как производственные, и как исследовательские.

Уровень взаимодействия с нарушителем определяет, какие возможности предоставляет ОбС атакующему по реализации атак. Чем большую свободу имеет атакующий, тем больше информации можно собрать о его действиях, и

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

тем больше объем работ по установке и поддержке системы, а также выше риск ее компрометации. По уровню взаимодействия с нарушителем различают следующие типы ОбС: (1) с низким уровнем взаимодействия; (2) со средним уровнем взаимодействия; (3) с высоким уровнем взаимодействия.

Характеристика ОбС по уровню взаимодействия в представлена табл. 1.

Таблица 1. Характеристика ОбС по уровню взаимодействия

Уровень взаимодействия	Трудоемкость установки и конфигурирования	Трудоемкость развертывания и поддержки	Возможности по сбору информации	Уровень риска
Низкий	Низкая	Низкая	Ограниченные	Низкий
Средний	Средняя	Средняя	Изменяемые	Средний
Высокий	Высокая	Высокая	Значительные	Высокий

Как правило, уровень интерактивности в производственных ОбС, в отличие от исследовательских, довольно низок. Уровень интерактивности определяет, насколько активно может вести себя нарушитель, проникший в информационную систему. Чем больше уровень интерактивности, тем больше нарушитель может сделать, и тем больше можно получить информации о нарушителе, но и тем больше вреда он может нанести. Большинство систем с низким уровнем интерактивности эмулируют те или иные службы. ОбС с низким уровнем интерактивности, например, может эмулировать FTP-сервер или Web-сервер. Насколько активно нарушитель будет работать в системе, зависит от уровня эмуляции, свойственного конкретной реализации. ОбС с высоким уровнем интерактивности службы не эмулируют. Вместо этого они предоставляют реальную операционную среду и реальные сервисы.

Перечислим *основные достоинства применения ОбС* [1, 2, 6–8]*:*

- ОбС базируются на сборе данных небольшого объема, так как они ориентированы на фиксацию только действий нарушителей и любое взаимодействие с ОбС, вероятнее всего, вызвано неправомочными или злонамеренными действиями. Вследствие этого в используемых данных практически нет “шума”, что обуславливает отсутствие ложных срабатываний при обнаружении вторжений и возможность выявления новых атак, средств их реализации и стратегий злоумышленников;
- ОбС требуют минимальных ресурсов, так как они используют данные, характеризующие только неправомочные или злонамеренные действия;
- все типы ОбС основываются на простой стратегии — если кто-то взаимодействует с ОбС, отслеживай его действия и реагируй на них. Понятно, что чем проще компонент защиты, тем менее вероятны ошибки функционирования и сбои в работе;
- в отличие от большинства компонентов защиты (например, систем обнаружения вторжений (СОВ)) ОбС могут работать с зашифрованным трафиком или в сети, функционирующей по протоколу IPv6, так как не имеет значения, какая информация поступает на вход ОбС, она будет обнаружена и зафиксирована;
- так как практически любой трафик, направленный на ОбС, отражает действия нарушителя, ОбС могут обнаруживать новые атаки;
- ОбС позволяют непрерывно демонстрировать руководству организаций свою значимость, а также подтверждать роль других механизмов защиты. Всякий раз, когда на компоненты ОбС осуществляется атака, администраторы безопасности и руководство будут проинформированы об этом.

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

В качестве *основных недостатков* использования ОбС отметим следующие [1, 2, 6–8]*:*

- ОбС имеют ограниченную область применения, так как могут отслеживать только деятельность, которая непосредственно направлена на них; ОбС не могут обнаруживать и реагировать на деятельность против других систем, если нарушитель не взаимодействует с ОбС;
- ОбС обуславливают риск полной компрометации и использования для атак на другие компьютерные системы;
- наличие у ОбС “демаскирующих признаков”, т.е. отличительных характеристик и поведения, обнаруживаемых нарушителем по ответной реакции ОбС на его действия. Наличие данного недостатка может привести к тому, что нарушитель, в свою очередь, попытается обмануть систему защиты для реализации своей цели. Например, если нарушитель идентифицировал использование ОбС, он может атаковать ее от имени реального компьютера целевой системы. ОбС обнаружит эту атаку и ошибочно оповестит администратора о том, что целевая система была использована злоумышленником, порождая цепочку разбирательств, ведущих по ложному следу, а в это время нарушитель сможет сосредоточиться на реальных целях. Следует отметить, что согласно реализуемой политике безопасности ОбС может специально оповещать нарушителей о себе. Это может делаться для отпугивания злоумышленников от защищаемой сети [10]. Однако, в силу представленных выше причин, представляется, что в большинстве случаев более разумно, чтобы ОбС действовали скрытно и не были обнаружены.

4. Характеристика исследований в области построения обманных систем

В настоящее время исследования в области построения ОбС — одно из наиболее бурно развивающихся направлений в защите информации. В англоязычной литераторе для ОбС используются различные термины, в том числе honeypot (“горшочек меда”, хост-приманка), honeynet (“медовая сеть”, сеть-приманка), decoy system (система-приманка), deception system (обманная система) и др. Исследования в области ОбС были инициированы с начала 1990-х годов [10]. Одной из первых работ по использованию механизмов обмана для защиты информационных систем была работа Била Чеквика (Bill Cheswick) [11]. В ней раскрывались сценарии трассировки в реальном времени действий нарушителя на основе механизмов введения нарушителя в заблуждение.

Подход к использованию обмана для защиты компьютерных систем посредством их настройки против автоматизированных средств реализации атак и примеры его использования были изложены в работах Фреда Коэна (Fred Cohen) [12–14]. Более глубокое исследование роли методов обмана для защиты компьютерных систем представлено Фредом Коэном в работе [5]. С этого времени (с конца 90-х годов), механизмы обмана все в большей степени становились предметом исследований в области безопасности информационных систем. Примеры механизмов защиты компьютерных систем согласно [10] включают скрытые сервисы, шифрование, предоставление ложной информации, трудно подбираемые пароли, изолированные области файловых систем, зашумление информации, управление восприятием, атаки, направленные на изменение маршрутизации сетевых пакетов, ловушки и др.

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Разработка свободно распространяемого программного пакета DTK (Deception ToolKit) привела к появлению целой серии различных исследований и разработке новых технологий введения в заблуждение нарушителей для защиты информационных систем, включая создание множества коммерческих продуктов, реализацию нескольких исследовательских проектов, например, Honeynet Project [15], проект RIDLR в Naval Post Graduate School [16–18], исследования фирмы RAND [19, 20], разработку технологии D-Wall [21, 22] и др.

Наиболее продвинутым интернациональным проектом в области создания ОбС является Honeynet Project [1, 2, 15]. Honeynet Project — это научная организация, занимающаяся исследованиями в области защиты информации и специализирующаяся на изучении инструментария, используемого злоумышленниками, их тактики и мотивов. В состав организации входят специалисты по вопросам безопасности из разных стран, которые на добровольной основе предоставляют свои ресурсы для развертывания и изучения сетей-приманок, основное назначение которых — стать объектом атаки хакеров.

В настоящее время существует множество коммерческих и свободно распространяемых ОбС. Они различаются уровнем имитации реальных систем, количеством поддерживаемых протоколов, конструкцией, условиями распространения и т.п. Часть ОбС могут эмулировать только некоторые сервисы или уязвимости, причем на том компьютере, на котором они запущены. Примерами таких систем являются Decoy-режим RealSecure Server Sensor, WinDog-DTK или система The Deception Toolkit (DTK). Более развитые системы эмулируют не отдельные сервисы, а сразу целые компьютеры и даже сегменты, содержащие виртуальные узлы, функционирующие под управлением разных ОС. Примеры таких систем — CyberCop Sting или Honeyd.

Большинство существующих систем являются уникальными системами, которые имеют возможность имитации определенных сервисов, ОС и приложений, ведут лог (базу данных) действий атакующих и позволяют реализовать различные способы реагирования на атаки. Однако, злоумышленники-профессионалы выработали пути обхода этих систем.

Среди наиболее известных систем можно выделить следующие: Back Officer Friendly [23], Bait N Switch Honeypot [24], BigEye [25], Decoy Server (ManTrap) [26], FakeAP [27], HoneyD [28], HoneyWeb [29], KFSensor [30], LaBrea Tarpit [31], NetBait [32], NetFacade [33], Smoke Detector [34], Specter [35] и др.

5. Функции и структура перспективной обманной системы

В настоящем разделе сформулированы теоретические положения по созданию перспективной производственной ОбС, прототип которой разрабатывается в лаборатории интеллектуальных систем СПИИРАН [4].

Место компонентов данной ОбС в компьютерной сети представлено на рис. 2. Предполагается, что поступающие из сети Интернет сетевые пакеты (1) вначале проходят предварительную фильтрацию посредством межсетевого экрана (МЭ) (2), затем анализируются на предмет наличия атак системой обнаружения вторжений (СОВ), если пакет отнесен СОВ к категории подозрительных или обнаружена явная атака, он перенаправляется на компоненты ОбС (3). Если СОВ не удалось обнаружить атаку на сетевом уровне, но злоумышленные действия были выявлены СОВ после их реализации на хостах целевой системы (4), осуществляется перенаправление последующих пакетов злоумышленника на компоненты ОбС.

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

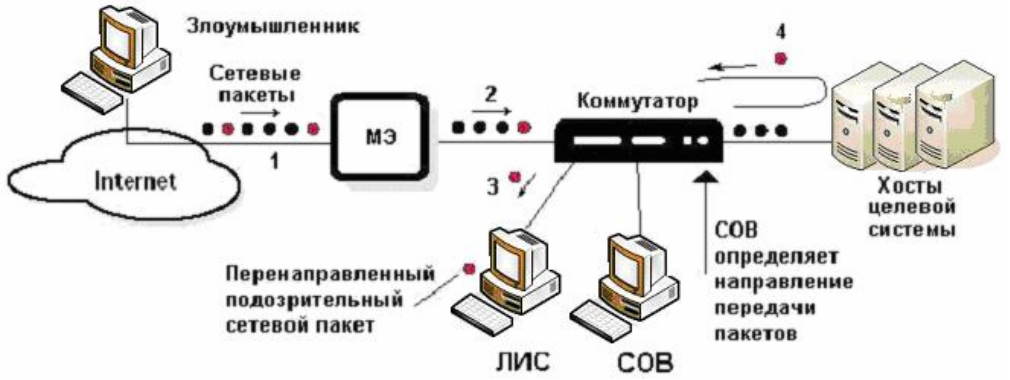

Рис.2. Место компонентов ОбС в компьютерной сети

В качестве основных функций, которые должны быть реализованы в перспективной ОбС, на основании анализа исследований в указанной области были выделим следующие:

- захват данных (“прослушивание” сетевого трафика и фиксация данных для последующего анализа);
- сбор и объединение данных от различных программных и аппаратных компонентов компьютерной сети, в частности, сенсоров, МЭ, СОВ, маршрутизаторов и др.;
- определение “свой-чужой” и переадресация несанкционированных запросов на компоненты ОбС;
- фильтрация событий (для автоматической отбраковки несущественных и фокусировки на значимых событиях);
- обнаружение вторжений (атак);
- выявление источника угроз, трассировка и идентификация нарушителя (определение типа, квалификации и др.);
- распознавание плана (стратегии) действий нарушителя;
- контроль действий нарушителя и реагирование на них, в том числе оповещение администратора о компрометации, блокирование действий нарушителя и др.;
- формирование плана действий компонентов ОбС по имитации целевой информационной системы;
- заманивание и обман нарушителя (привлечение внимания, сокрытие реальной структуры защищаемой системы и ресурсов, камуфляж, дезинформация) за счет эмуляции сетевых сегментов, серверов, рабочих станций, в том числе передаваемого трафика, и их уязвимостей, автоматическое реагирование на действия нарушителя, в том числе оповещение администратора;
- удаленное администрирование, документирование, ввод сигнатур, профилей и др. (обеспечивает централизованное управление, основанную на правилах безопасности реакцию системы, подготовку отчетов и анализ тенденций);
- обеспечение интерфейса с администратором безопасности.

Следует отметить, что кроме реализации собственно обманных действий, ОбС, в обязательном порядке, должна обеспечивать выполнение, по крайней мере, двух функций — контроль и сбор данных. Цель контроля данных состоит в обеспечении невозможности использования скомпрометированных компонентов (ресурсов) для атаки или для нанесения вреда другим системам после про-

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

никновения нарушителя в ОбС. Сбор данных гарантирует, что можно обнаружить и зарегистрировать все действия нарушителей, даже если они замаскированы или зашифрованы.

Обобщенная функциональная структура перспективной ОбС представлена на рис. 3. Жирным шрифтом выделены базовые компоненты ОбС.

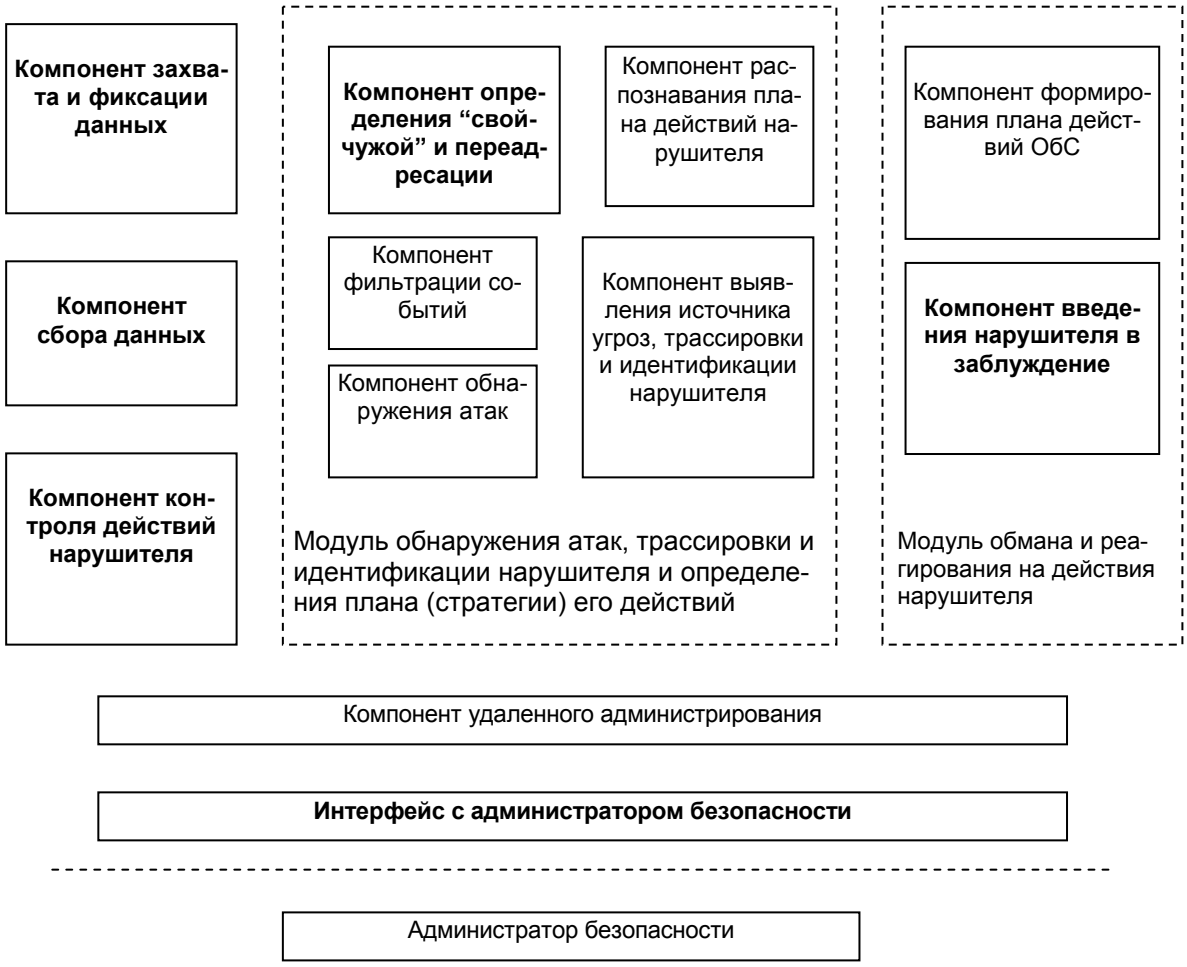

Рис.3. Обобщенная функциональная структура ОбС

Предполагается, что при реализации своих функций ОбС обеспечивает три представленных ниже уровня введения в заблуждение (обмана) (рис. 4).

1. Уровень сегмента (основных компонентов целевой системы). На данном уровне ОбС имитирует целевую систему в целом. При обнаружении атаки злоумышленник перенаправляется с целевой системы на ОбС. Этот уровень соответствует сетям ловушек ("Honeynet") [7, 8] и, в большей степени, их развитию — парадигме "ферм ловушек" ("Honeypot Farms") [36].

2. Уровень хоста. Данный уровень предполагает размещение компонентов ОбС, имитирующих отдельные хосты, в компьютерной сети целевой системы. Этот уровень соответствует парадигме "хостов-ловушек" ("Honeypot") [6].

3. Уровень сервиса/приложения. В рамках хоста целевой системы каждое приложение/сервис формируется следующим образом: целевой модуль сервиса/приложения вместе с модулем обмана вкладывается в обертку. В режиме санкционированного использования при вызове сервиса/приложения управление передается целевому модулю. При обнаружении несанкционированного

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

обращения управление передается модулю обмана. Этот уровень соответствует парадигме “программных ловушек” (“Software Decoys”) [16–18].

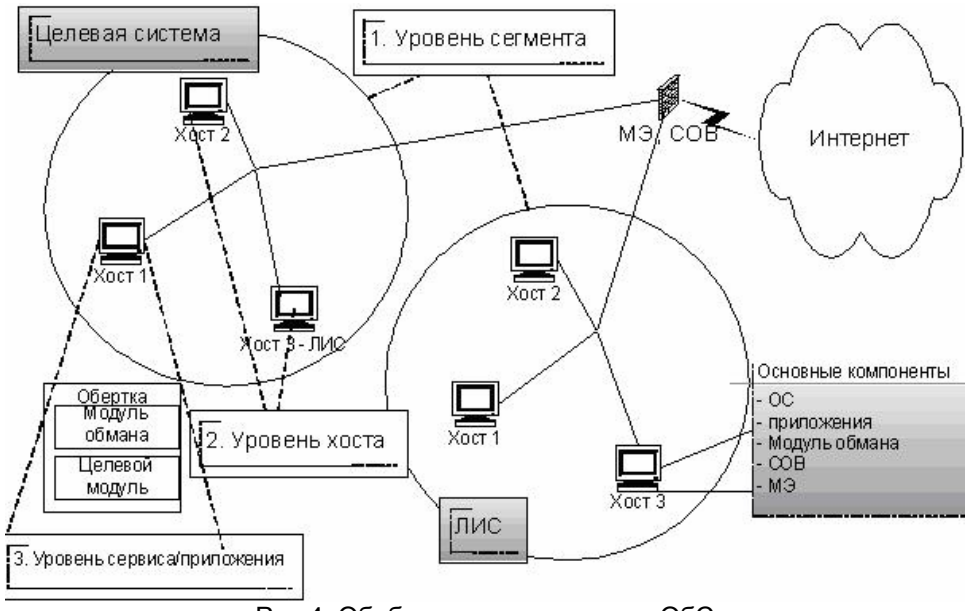

Рис.4. Обобщенная архитектура ОбС
и реализуемые ОбС уровни введения в заблуждение (обмана)

6. Обобщенное описание схем реализации отдельных функций обманной системы

Представим краткое описание используемых схем реализации некоторых базовых функций ОбС, в частности, (1) сбора и объединения данных, (2) определения “свой-чужой” и переадресации несанкционированных запросов на компоненты ОбС, (3) контроля действий нарушителя, (4) обмана нарушителя и автоматического реагирования на его действия.

Сбор и объединение данных от различных программных и аппаратных компонентов компьютерной сети (сенсоров, МЭ, СОВ, маршрутизаторов и др.) данных позволяет зарегистрировать все действия нарушителей, даже если они замаскированы или зашифрованы. Задача и сбора и объединения данных является одной из наиболее сложных в реализации ОбС. Связано это с тем, что существует множество источников данных и уровней обработки информации, кроме того, различными компонентами информационных систем используются несовместимые между собой форматы данных.

Можно выделить, по крайней мере, три основных источника данных, которые нужно уметь собирать, обрабатывать и объединять: (1) дамп сетевого трафика (из нескольких источников), (2) журналы регистрации событий ОС, (3) журналы регистрации событий приложений (СОВ, МЭ, баз данных и др.).

Для обеспечения эффективной фиксации действий нарушителей необходимо использовать несколько различных уровней сбора данных [1, 2].

Первый уровень — это шлюз на границе периметра защищаемой сети. Сетевые пакеты на данном уровне могут отслеживаться с использованием МЭ и СОВ. Здесь просматривается весь сетевой трафик, который поступает в сеть, и идентифицируются и блокируются удаленные атаки. Использование данного уровня обеспечивает запись и регистрацию всех действий нарушителей для последующего анализа. По журналу регистрации, содержащему информацию о

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

передаче файлов, можно определить, какой инструментарий применялся нарушителем. Даже при использовании зашифрованных протоколов, с помощью пассивного анализа характерных признаков пакетов, можно определить тип атакующей системы и ее возможное местонахождение.

Второй уровень сбора данных — это журнал регистрации мостового (граничного) компонента второго уровня, реализуемого, например, на основе использования МЭ или СОВ. Этот компонент должен иметь механизм фильтрации и модификации пакетов, позволяющий блокировать исходящие соединения при обнаружении определенной сигнатуры (например, достижении установленного предельного числа исходящих соединений) и (или) изменять содержимое сетевых пакетов, обезвреживая атаки.

Третий уровень предназначен для сбора информации о деятельности нарушителя в системе, в том числе о командах, инициированных нарушителем. Нарушители, чтобы скрыть свои действия, могут использовать шифрование. Например, как только нарушитель проник на хост ОбС, он может осуществлять удаленное администрирование системы с помощью SSH. Для решения этой проблемы можно использовать специальные модули ядра ОС, устанавливаемые на хостах, которые могут стать объектами атак. Эти модули накапливают информацию обо всей деятельности нарушителей. Информацию, которую собирают модули ядра, нельзя сохранять локально на хосте, поскольку нарушитель может обнаружить и удалить или изменить эту информацию. Поэтому указанную информацию необходимо удаленно собирать на защищенной системе, причем так, чтобы нарушитель об этом не знал. Это должна делать СОВ компонента второго уровня. Она действует как сетевой анализатор, накапливающий сведения и регистрирующий всю деятельность нарушителей, записывая, в том числе, все пакеты, сгенерированные модулями ядра.

Однако нарушители могут проанализировать трафик в ОбС и обнаружить, что в пересылаемых пакетах содержатся сведения об их собственной деятельности. Чтобы воспрепятствовать этому, модуль ядра должен маскировать пакеты, например, под трафик NetBIOS, передаваемый из других систем. Причем IP- и MAC-адреса отправителей и получателей могут маскироваться под адреса локального сервера Windows, а данные, содержащиеся в пакетах, — шифроваться. В этом случае даже если нарушитель осуществляет перехват и анализ пакетов, то для него они будут выглядеть как обычный трафик.

Важным аспектом в решении поставленной задачи является возможность использования стандартных средств ОС для сбора журналов регистрации событий. В качестве подобных средств можно использовать программный пакет syslog-ng, способный собирать журналы регистрации событий с ОС Linux на одном выделенном сервере. Для ОС Windows существует несколько клиентов, работающих совместно с syslog-ng. Подобная связка позволяет практически для любой конфигурации защищаемой сети использовать одинаковый механизм регистрации событий на уровне ОС. Необходимо также, чтобы такие компоненты ОбС, как антивирусное ПО, сервисы и приложения, системы контроля целостности файлов и др., также записывали события, отражающие их работу, в журналы регистрации событий. Таким образом, на одном выделенном сервере будут сосредоточены дампы сетевого трафика с нескольких хостов сети и общий для всей сети журнал регистрации событий, что позволит администратору иметь общую картину о состоянии сети на любой момент времени.

Основным механизмом *определения “свой-чужой”* является использование сетевой СОВ, функционирующей на граничном хосте. При обнаружении зло-

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

умышленного трафика СОВ должна его блокировать до тех пор, пока МЭ не будет должным образом переконфигурирован, чтобы передавать сетевые пакеты на компоненты ОбС, а не на компоненты целевой системы. Кроме блокирования трафика, СОВ должна обеспечивать уведомление администратора (который будет сам изменять конфигурацию МЭ), либо производить запуск внешней утилиты, конфигурирующей МЭ.

Цель *контроля данных* — снижение риска использования ОбС. Для этого необходимо гарантировать, что после того, как нарушитель проник в ОбС, он не сможет использовать ее ресурсы для реализации атак на другие системы. Эта задача может быть выполнена с помощью дополнительного мостового (граничного) компонента ОбС, реализуемого, например, на основе использования МЭ или СОВ. Этот компонент реализует фильтрацию и модификацию исходящих сетевых пакетов, обеспечивая блокировку исходящих соединений при обнаружении определенной сигнатуры и (или) изменение содержимого сетевых пакетов для обезвреживания атак.

Существует несколько вариантов решений по реализации механизмов контроля данных.

Наиболее простое решение состоит в создании среды, в которой разрешены все входящие соединения, но блокируются все исходящие. Это позволяет не допустить причинения вреда другим системам, но подобная ОбС будет практически бесполезной, так как определить, что делали нарушители после того, как проникли в сеть, будет невозможно. Кроме того, нарушителям не составит труда распознать ОбС, если после попадания в нее они будут лишены обратной связи.

Второе решение предполагает подсчет числа исходящих соединений и блокировку любых соединений, превышающих установленный лимит. Данное решение было положено в основу технологии, реализованной в GenI проекта Honeynet Project [1, 2, 15]. Как только число соединений, исходящих из ОбС, достигнет определенного предела, все остальные исходящие соединения блокируются. Указанный механизм предотвращает большинство DoS-атак, сканирование или другую вредоносную деятельность, но по-прежнему оставляет нарушителям достаточную свободу действий. Нарушители по-прежнему могут опознавать ОбС и инициировать атаки, не превышая ограничения на число исходящих соединений.

Третье решение состоит в модификации и обезвреживании исходящих атак. Этот подход использован в ОбС GenII проекта Honeynet Project [1, 2, 15]. Для реализации этого уровня может использоваться система Snort-Inline, представляющая собой модифицированную версию СОВ Snort. Нарушители запускают свои эксплоиты, но Snort-Inline обезвреживает такие атаки, изменяя код эксплоитов.

При *обмане нарушителя* осуществляется привлечение его внимания за счет имитации различных уязвимостей, сокрытие реальной структуры защищаемой АС и ее ресурсов, введение нарушителя в заблуждение и навязывание ему ложной информации за счет эмуляции несуществующих сетевых сегментов, серверов, рабочих станций, их уязвимостей и передаваемого трафика. В соответствии с предложенной схемой функционирования обманных компонентов ОбС, если СОВ обнаруживает сетевые пакеты, задающие несанкционированные действия или удаленную атаку, они фиксируются и перенаправляются на

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

обманные компоненты ОбС. Если тип атаки известен, то отклик на атаку формируется обманным компонентом автоматически. Например, если нарушитель пытается прочитать содержимое какого-либо файла, обманные компоненты передают содержимое подложного файла и т.п.

7. Реализация прототипа обманной системы

Для реализации прототипа ОбС предложена следующая структура компьютерной сети, представленная на рис. 5.

Основные компоненты сети: (1) граничный хост, (2) персональные компьютеры (ПК) ЛВС, (3) демилитаризованная зона (ДМЗ) с расположенными в ней рабочими серверами, а также (4) подсеть с серверами, играющая роль сети-приманки.

Граничный хост выполняет следующие задачи: обеспечение безопасного доступа ПК ЛВС и рабочих серверов в публичную сеть Internet, блокировка запросов внешних пользователей к ПК ЛВС, ограничение исходящего сетевого трафика из сети-приманки, фильтрация и маршрутизация сетевого трафика, обнаружение вторжений на сетевом уровне, сбор с ПК всей сети данных из журналов регистрации событий. На граничном хосте используется следующее программное обеспечение: операционная система (ОС) Debian GNU/Linux [37]; iptables (обеспечивает фильтрацию и маршрутизацию сетевого трафика) [38-40]; snort_inline (сетевая система обнаружения вторжений с возможностью блокирования действий злоумышленника) [38-40]; swatch (анализ журнала регистрации событий ОС) [41]; syslog-ng (регистрация событий на уровне ОС).

Предполагается, что на жестких дисках *ПК ЛВС* никакой важной информации пользователи не хранят, а вся работа производится с данными, расположенными на серверах (например, могут использоваться так называемые “тонкие клиенты”). Любая попытка произвести соединение из публичной сети Internet с ПК ЛВС блокируется граничным хостом.

В *ДМЗ* располагаются рабочие сервера, в частности web-сервер, ftp-сервер, mail-сервер (реализующий протоколы POP3 и SMTP) и telnet-сервер. В ДМЗ также развертывается *хост-приманка*, который для пользователей представляется в качестве рабочего сервера. Все запросы, приходящие на данный хост, рассматриваются как носящие заведомо злоумышленный характер. Граничный хост должен обеспечивать маршрутизацию сетевых пакетов, идущих на хост-приманку, в сеть-приманку. Для контроля целостности критически важных файлов на серверах устанавливается система “GFI Languard System Integrity Monitor”, позволяющая записывать предупреждения обо всех изменениях файлов в системный журнал регистрации событий. На серверах используется ОС “Windows 2000 Server”, поэтому для сбора журналов регистрации событий ОС в общий журнал (расположенный на граничном хосте) необходимо использовать утилиту “eventlog to syslog”. На рабочих серверах ряд приложений формируются следующим образом: целевой модуль сервиса вместе с модулем обмана вкладывается в обертку. В режиме санкционированного использования при вызове сервиса управление передается целевому модулю. При обнаружении несанкционированного обращения управление передается модулю обмана.

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

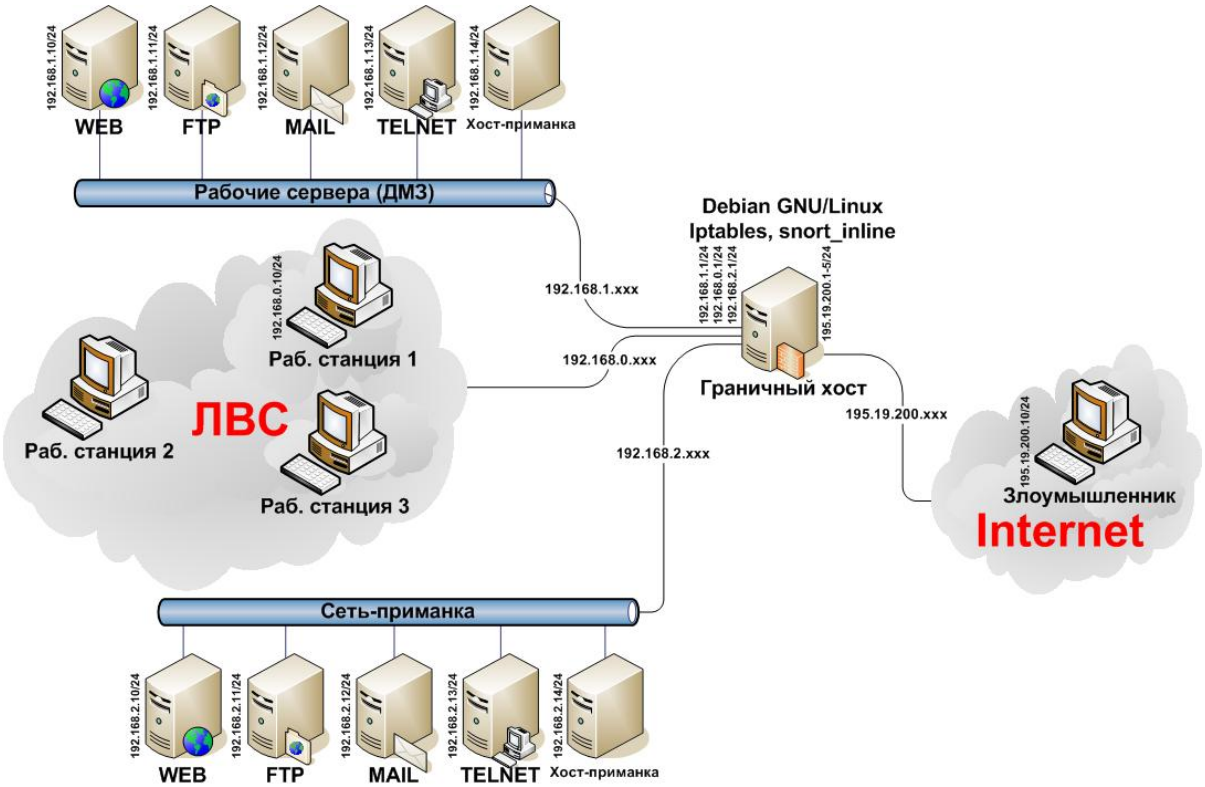

Рис.5. Архитектура компьютерной сети

Серверы *сети-приманки* имитируют функции рабочих серверов. На этих серверах производится программная эмуляция работы сервисов и приложений web, ftp, mail и telnet. Хосты-приманки могут быть как системами с эмулируемыми сервисами, так и обычными незащищенными системами. Любая попытка произвести с сервера сети-приманки соединение с рабочим сервером или с ПК из ЛВС должна блокироваться граничным хостом.

Для обнаружения изменений в критически важных файлах на всех ПК сети используется система проверки целостности файлов, например, GFI Languard System Integrity Monitor (для Windows-платформы).

Эмуляция работы таких сервисов как http, ftp, telnet, pop3, smtp на серверах сети-приманки производится с помощью программ, разработанных на языке программирования Perl. Функционирование программ описано формально при помощи конечных автоматов.

ОбС обладает интерфейсом администратора, с использованием которого в реальном времени приводится анализ общего журнала регистрации событий и отражается сетевой трафик, проходящий через граничный хост.

Для реализации экспериментов на ПК, расположенном в публичной сети Internet, эмулируются действия злоумышленника.

Каждый ПК, входящий в состав имитируемой компьютерной сети (см. рис. 5), может являться как физическим ПК, так и виртуальным. В последнем случае ПК должен эмулироваться с помощью различных программных средств. Авторами были исследованы три пакета эмуляции — Connectix Virtual PC, Microsoft Virtual PC 2004 и VMware WorkStation. На время проведения экспериментов наиболее развитыми функциональными возможностями обладал пакет VMware WorkStation [42, 43]. Кроме того, он является свободно распространяемым программным продуктом. Поэтому выбор был сделан в пользу VMware WorkStation.

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Первоначальные эксперименты по исследованию возможностей ОбС проводились на основе эмуляции всех ПК с помощью VMware Workstation на одном физическом компьютере. Для выполнения этой задачи данный компьютер должен обладать следующими ресурсами: оперативная память — 64–128Mb на каждый эмулируемый ПК, дисковое пространство — порядка 2Gb на каждый эмулируемый ПК. Пользовательский интерфейс программы VMware Workstation представлен на рис. 2. Достоинства предложенного решения: сокращение времени подготовки программных средств (операционных систем и приложений) к эксперименту; оперативность восстановления рабочего окружения в случае разрушения данных на одном из ПК (вследствие удачных деструктивных действий злоумышленника); удобство и простота изменения различных параметров ОбС, необходимая для изучения отклика ОбС на действия злоумышленника при проведении экспериментов.

Для построения отдельных компонентов ОбС использовались следующие программные средства:

(1) операционная система — Debian GNU/Linux 3.0, release 1 (на граничном хосте), Windows XP SP1 English (на ПК ЛВС и ПК злоумышленника), Windows 2000 Server SP4 English (на серверах ДМЗ и сети-приманки);

(2) WWW-сервер — Internet Information Services 5.0 (IIS), входящий в состав Windows 2000 Server, или Apache;

(3) ftp-сервер — Internet Information Services 5.0 (IIS), входящий в состав Windows 2000 Server, или Serv-U 4.1 (<http://www.serv-u.com>);

(4) telnet-сервер — сервер, входящий в состав Windows 2000 Server; (5) почтовый сервер — mdaemon for Windows;

(6) компонент маршрутизации IP-трафика и межсетевой экран — пакет iptables (Debian GNU/Linux);

(7) система обнаружения вторжений (СОВ) — snort_inline (Debian GNU/Linux), snort (Debian GNU/Linux и Microsoft Windows);

(8) средства регистрации событий операционной системы — syslog-ng (Debian GNU/Linux) и “eventlog to syslog” (Microsoft Windows);

(9) средства контроля целостности файлов — debsums (Debian GNU/Linux), GFI languard System Integrity Monitor (Microsoft Windows);

(10) компоненты регистрации сетевого трафика — tcpdump (Debian GNU/Linux) и windump (Microsoft Windows);

(11) средство анализа журнала регистрации событий ОС — swatch (Debian GNU/Linux).

Используемые программные средства, отсортированные в соответствии с основными функциями ОбС, представлены в табл. 2.

8. Сценарии проведения экспериментов

Используя данный подход к реализации ОбС, в настоящее время проводятся эксперименты по изучению возможностей по реализации основных функций введения злоумышленников в заблуждение при реализации на рабочие сервера различного рода атак. Эксперименты выполняются по нескольким сценариям, определяемым в соответствии с типом атак, направленных на рабочие серверы.

В соответствии с тем, обнаружены ли атаки на граничном хосте с использованием сетевой системы обнаружения вторжений snort_inline, их можно разделить на два типа: (1) обнаруживаемые атаки; (2) не обнаруживаемые атаки.

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Атаки первой группы должны блокироваться граничным хостом, не достигая рабочих серверов. При обнаружении этих атак ОбС способен изменять свою конфигурацию таким образом, чтобы последующие действия атакующего перенаправлялись на серверы сети-приманки.

Таблица 2. Основные функции прототипа ОбС и реализующие их программные средства

Функция	Программные средства
1. Заманивание и обман нарушителя	Программы, эмулирующие работу сервисов (www, ftp, telnet и mail).
2. Обнаружение и фиксация действий нарушителя	Программные средства обнаружения вторжений и фиксации действий на сетевом уровне: snort_inline, windump, tcpdump. Программные средства фиксации действий на уровне ОС: syslog-ng в качестве сервера (на граничном хосте) для захвата всех сообщений со всех ПК сети; eventlog to syslog для Windows в качестве клиента syslog-ng.
3. Контроль действий нарушителя	Программный пакет Iptables на граничном хосте для ограничения исходящего из сети-приманки трафика и для реализации IP-маршрутизации.
4. Сбор и агрегация данных о действиях нарушителей из различных источников	Программные компоненты фильтрации событий (из журналов регистрации событий ОС и из проходящих по сети пакетов), упорядочивания событий по шкале времени и представления данной информации администратору в удобочитаемом виде. Программные компоненты объединения данных (журналы регистрации событий со всех ПК сети объединяются в один общий журнал, расположенный на граничном хосте; сетевые пакеты записываются в текстовый файл программой tcpdump, windump или snort).

Общий алгоритм работы ОбС для атак данной группы можно описать следующим образом: (1) злоумышленник выполняет некоторые действия, которые обнаруживает СОВ snort_inline; (2) snort_inline блокирует действия злоумышленника (происходит разрыв сетевого соединения), оповещает администратора безопасности и записывает в журнал регистрации событий ОС сообщение об обнаружении злоумышленника, действующего с IP-адреса xxx.xxx.xxx.xxx; (3) анализируя в режиме реального времени поступающие в журнал регистрации событий сообщения программа swatch находит сообщение от snort_inline и выполняет команду ОС, изменяющую конфигурацию пакета iptables; (4) как только данные изменения вступают в силу, все сетевые пакеты с IP-адреса xxx.xxx.xxx.xxx направляются на серверы сети-приманки. Таким образом, если злоумышленник попытается вновь (после закрытия сетевого соединения) произвести атаку рабочего сервера, его запросы автоматически будут перенаправлены на сервер-приманку, где все его действия будут записаны для дальнейшего анализа.

К атакам второй группы можно отнести атаки, сигнатуры которых отсутствуют в базе сетевой СОВ, а также атаки внутренних злоумышленников, имеющих физический доступ к рабочим серверам. В этом случае факт проведения атакующих действий можно выявить при анализе общего журнала регистрации событий, например, по сообщениям о перезагрузке сервера или отдельного сервиса, поступившим из журналов регистрации событий операционных систем, или по сообщениям от системы контроля целостности файлов об изменении критически важных файлов. После выявления в журнале регистрации событий этих сообщений действия злоумышленника блокируются и выполняется оповещение администратора. Если атака — внешняя, программа swatch выполняет

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

команду ОС, изменяющую конфигурацию пакета iptables, и далее все сетевые пакеты с IP-адреса злоумышленника направляются на серверы сети-приманки.

Приведем несколько примеров сценариев проведения атак на рабочие серверы.

Сценарий 1. Атака злоумышленника из сети Internet на web-сервер, расположенный в ДМЗ. Предполагается, что данная атака принадлежит к первой группе. Допустим, злоумышленник пытается получить файл с паролями (/etc/passwd) с помощью изъяна phf. Он производит соединение с web-сервером и передает запрос “GET http://web-server-ip/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd”. Этот запрос обнаруживает сетевая СОВ snort_inline, которая производит блокировку дальнейших действий (сетевой пакет с запросом отбрасывается, соединение разрывается) и оповещает администратора безопасности. Злоумышленник видит web-страницу, сообщающую о невозможности отобразить ответ сервера, либо (в общем случае) сообщение “Соединение разорвано сервером” (“connection closed by remote side”).

Snort_inline записывает в журнал регистрации событий ОС сообщение об обнаружении атаки с IP-адреса «xxx.xxx.xxx.xxx». Swatch при просмотре общего журнала регистрации событий, расположенного на граничном хосте, замечает данное сообщение и выполняет команду по изменению конфигурации граничного хоста. Последующие запросы с IP-адреса злоумышленника перенаправляются к серверам, расположенным в сети-приманке. Злоумышленник обнаруживает, что его запрос не был выполнен по какой-либо причине (одной из таких причин может быть потеря сетевого пакета во время маршрутизации). Он пытается повторить его. При этом запрос идет на web-сервер из сети-приманки, который, например, (с помощью технологии “tcp-wrippers”) возвращает злоумышленнику поддельный файл с паролями. Дальнейшие действия ОбС — введение злоумышленника в заблуждение и сбор информации о злоумышленнике.

Сценарий 2. Атака внутреннего злоумышленника с использованием физического доступа к web-серверу, расположенному в ДМЗ. Данная атака относится ко второй группе атак, которые не обнаруживаются сетевой СОВ. Допустим, злоумышленник каким-либо образом проник в помещение, где расположен web-сервер. Его задача — подменить заглавную страницу на корпоративном web-сервере. Для этого ему необходимо иметь учетную запись на сервере, и, желательно, учетную запись администратора. Злоумышленник решает “сбросить” пароль администратора путем удаления базы данных SAM. Для этого он заранее создает загрузочную дискету DOS и копирует на нее программу ntfsdospro. Далее злоумышленник производит загрузку сервера с этой дискеты, запускает ntfsdospro и удаляет базу данных SAM командой “del c:\winnt\system32\config\sam” (предполагается, что установка ОС и IIS производилась в каталоги по умолчанию). После этого он загружает сервер, входит в систему под именем администратора с пустым паролем и производит подмену первой странички сайта компании.

Атака может быть обнаружена по нескольким событиям в журнале регистрации событий ОС, например, последовательность событий об успешной остановке функционирования ОС (shutdown) и об успешном рестарте ОС (startup). Кроме того, программа “GFI Languard System Integrity Monitor” сгенерирует события об изменении файлов “c:\winnt\system32\config\sam” и “c:\inetpub\wwwroot\index.html” (предполагается, что установка ОС и IIS производилась в каталоги по умолчанию). ОбС автоматически блокирует работу сервера, и оповещает об этих событиях администратора безопасности.

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Сценарий 3. Атака злоумышленника из сети Internet на ftp-сервер с использованием неверной конфигурации ftp-сервера. Предполагается, что данная атака относится ко второй группе атак (не обнаруживаемых сетевой СОВ). Допустим, что злоумышленник обнаруживает, что у пользователя webmaster пароль доступа к ftp-серверу “webmaster”. Используя эти данные, он подменяет заглавную страницу на web-сервере.

Атака может быть обнаружена по сообщению от программы “GFI languard System Integrity Monitor” в журнале регистрации событий ОС, которое оповещает об изменении пользователем webmaster файла “c:\inetpub\wwwroot\index.html” (предполагается, что установка ОС и IIS производилась в каталоги по умолчанию). После выявления в журнале регистрации событий этих сообщений действия злоумышленника блокируются и выполняется оповещение администратора, изменяется конфигурация пакета iptables, и все сетевые пакеты с IP-адреса злоумышленника направляются на Web-сервер сети-приманки.

9. Заключение

В статье охарактеризовано текущее состояние в области исследований ОбС, представлен предлагаемый подход к построению перспективной производственной ОбС, рассмотрены архитектура прототипа этой системы, а также сценарии экспериментов, проводимых с прототипом.

Рассматриваемый в статье подход основан на программной эмуляции компонентов информационных систем и на выделении трех уровней введения злоумышленников в заблуждение: (1) сегмента сети — производится эмулирование работы целого сегмента сети-приманки, дублирующего сегмент сети с рабочими серверами; (2) хоста — среди рабочих серверов используется хост-приманка; (3) сервисов и приложений — на отдельных серверах применяются программы, эмулирующие работу сервисов и приложений.

Направлениями дальнейших теоретических исследований является разработка и совершенствование моделей и алгоритмов реализации функций ОбС, в частности, по выявлению источника угроз, трассировке и профилированию нарушителя, идентификации плана действий нарушителя, формированию плана действий компонентов ОбС по имитации целевой информационной системы, сокрытию реальной структуры защищаемой системы и дезинформации злоумышленника, а также состав обрабатываемых атак и реализуемых сценариев работы.

Литература

[1] *Spitzner L.* Honeypots: Tracking Hackers. Addison Wesley, 2002.

[2] *Спитцнер Л.* Honeynet Project: ловушка для хакеров // Открытые системы, № 07–08, 2003.

[3] *Лукацкий А. В.* Обнаружение атак. СПб.: БХВ-Петербург, 2003.

[4] *Котенко И. В.*, *Степашкин М. В.* Прототип ложной информационной системы // XI Российская научно-техническая конференция (по Северо-западному региону) “Методы и технические средства обеспечения безопасности информации”: Тезисы докладов. СПб.: Издательство СПбГПУ, 2003.

[5] *Cohen F.* A Note on the Role of Deception in Information Protection // Computers and Security 1999.

[6] *Spitzner L.* Honeypots: Definitions and Values. May 2003. <http://www.tracking-hackers.com/papers/honeypots.html>

[7] *Spitzner L*. Know Your Enemy: Honeynets. The Honeynet Project, Jan 2003. <http://project.honeynet.org/papers/honeynet/>
[8] Honeynet Definitions, Requirements, and Standards. The Honeynet Project, 2003. <http://www.honeynet.org/alliance/requirements.html>
[9] Intrusion Detection: Generics and State-of-the-Art. RTO TECHNICAL REPORT, № 49. 2002.
[10] *Cohen F., Lambert D., Preston C., Berry N., Stewart C., Thomas E. A* Framework for Deception. 2001. <http://www.all.net/journal/deception/Framework/ Framework.html>
[11] *Cheswick B.* An Evening with Berferd, 1991.
[12] *Cohen F.* Operating System Protection Through Program Evolution // Computers and Security. 1992.
[13] *Cohen F.* Internet Holes - Internet Lightning Rods // Network Security Magazine, July, 1996.
[14] *Cohen F.* A Note On Distributed Coordinated Attacks // Computers and Security, 1996.
[15] Honeynet Project. <http://www.honeynet.org>
[16] *Michael J. B., Riehle R.* Intelligent software decoys // Proc. Monterey Workshop: Engineering Automation for Software-Intensive System Integration, Monterey, CA, June 2001.
[17] *Michael J. B., Auguston M., Rowe N. C., Riehle R.* Software decoys: intrusion detection and countermeasures // IEEE Information Assurance Workshop, West Point, New York, June 2002.
[18] *Rowe N. C., Michael J. B., Auguston M., Riehle R.* Software decoys for software counterintelligence // IAnewsletter, 5, 1, Spring 2002, 10-12.
[19] *Gerwehr S., Rothenberg J., Anderson R. H.* An Arsenal of Deceptions for INFOSEC (OUO) // PM-1167-NSA, RAND National Defense Research Institute Project Memorandum. October, 1999.
[20] *Gerwehr S., Weissler R., Medby J. J., Anderson R. H., Rothenberg J.* Employing Deception in Information Systems to Thwart Adversary Reconnaissance-Phase Activities (OUO) // PM-1124-NSA, RAND National Defense Research Institute. November, 2000.
[21] *Cohen F.* A Mathematical Structure of Simple Defensive Network Deceptions, 1999. <http://all.net> (InfoSec Baseline Studies).
[22] *Cohen F.* Method and Apparatus for Network Deception/Emulation // International Patent Application No PCT/US00/31295, Filed October 26, 2000.
[23] Back Officer Friendly. NFR. <http://www.nfr.com/resource/backOfficer.php>
[24] Bait N Switch Honeypot. Team Violating. <http://violating.us/projects/baitnswitch/>
[25] BigEye. Team Violating. <http://violating.us/projects/bigeye/>
[26] Decoy Server (ManTrap). Symantec. <http://enterprisesecurity.symantec.com/products/>
[27] FakeAP. Black Alchemy Enterprises. <http://www.blackalchemy.to/project/fakeap/>
[28] *HoneyD.* Niels Provos. <http://www.citi.umich.edu/u/provos/honeyd/>
[29] HoneyWeb. Kevin Timm. <http://www.var-log.com/files/>
[30] KFSensor. Keyfocus. <http://www.keyfocus.net/kfsensor/>
[31] LaBrea Tarpit. Tom Liston. <http://www.hackbusters.net/>
[32] NetBait. NetBait Inc. <http://www.netbaitinc.com/products/products.html>
[33] NetFacade. Verizon. <http://www22.verizon.com/fns/netsec/fns_netsecurity_netfacade.html>
[34] Smoke Detector. Palisade. <http://palisadesys.com/products/smokedetector/>
[35] Specter. Netsec. <http://www.specter.com/default50.htm>
[36] *Spitzner L.* Honeypot Farms. August 13, 2003. <http://www.securityfocus.com/ infocus/1720>
[37] *McCarty B.* Learning Debian GNU/Linux. O'Reilly & Associates, Inc. 2001.
[38] Snort. http://www.snort.org
[39] *Caswell B., Beale J., Foster J. C., Faircloth J.* Snort 2.0. Intrusion Detection. Syngress Publishing, 2003.
[40] *Rehman R. U.* Intrusion Detection Systems with Snort: Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID. Prentice Hall PTR. 2003.
[41] Swatch. http://www.oit.ucsb.edu/~eta/swatch
[42] *Barnett R. C.* Monitoring VMware Honeypots. Sep 2002. <http://honeypots.sourceforge.net/monitoring_vmware_honeypots.html>
[43] *Clark M.* Virtual Honeynets (using VMware). SecurityFocus InFocus Article, Nov 2001. <http://www.securityfocus.com/infocus/1506/>