А. В. Гаген, 014 гр.
2004 г.

Системы обнаружения атак на сетевом уровне (Network Intrusion Detection Systems)

Системы обнаружения атак на сетевом уровне (Network IDS, NIDS) являются программными продуктами, контролирующими сетевой трафик на определенных уровнях OSI, как правило, на третьем, “сетевом” (Network Layer), с целью обнаружения попыток проникновений злоумышленников в защищенную локальную систему. В качестве примера можно отметить NIDS, следящую за числом TCP-запросов на соединение с различными портами и пресекающую попытку сканирования открытых портов. Поскольку сервером NIDS просматривается трафик, адресатом которого он не является, то зачастую требуется установка специальной платы сетевого интерфейса (NIC), либо специальных драйверов, позволяющих пропускать “чужие” данные через первый и второй уровни OSI до третьего. Очевидно, что сервер NIDS способен контролировать трафик только того сегмента сети, которому он принадлежит (хотя это правило можно в некоторой мере обойти, как описано ниже). Сетевые IDS могут обслуживать сразу несколько компьютеров в сегменте, но имеются аргументы и в пользу установки NIDS для единственного хоста, например, важной базы данных. Здесь основным “за” является объем трафика, обрабатываемого сервером NIDS, в противовес которому ставится качество.

Что касается полноты защиты компьютеров от атак извне, то правильным решением будет использование NIDS в сочетании с традиционными средствами, такими как файрвол, шифрование, VPN, honeypots и т.д.

Основными игроками рынка коммерческих NIDS на сегодняшний день являются компании Cisco (www.cisco.com), ISS (http://www.iss.net), CyberCafe (www.cybercafe.com) и Shadow (www.nswc.navy.mil/ISSEC/CID).

1. Каким образом обнаруживаются атаки?

Существуют две основные концепции, применяющиеся для распознавания атак: использование знаний и использование шаблонов поведения.

1.1. Системы, основанные на базах знаний (knowledge based IDS)

В литературе NIDS, основанные на знаниях, также называются системами обнаружения злоупотреблений (misuse detection systems). Этот термин появился, наверное, потому, что любое действие внутри системы, тем или иным образом обнаруженное сервером IDS, считается атакой. Кстати, подавляющее большинство сетевых IDS сегодня основано на базах знаний, и ниже мы увидим, почему.

Итак, NIDS, использующие в своей работе знания, фактически используют сведения об атаках на систему, на которую они рассчитаны, и о ее уязвимостях. Сервер NIDS хранит внутри себя сведения о таких уязвимостях и следит за попытками хакеров использовать их. Когда обнаруживаются подобные попытки, то IDS выполняет некоторое заданное действие, например, перенастраивает файрвол. Иными словами, любое действие, не распознанное системой, как атака, считается разрешенным, что заставляет администратора NIDS регулярно обновлять базы знаний, тщательно анализировать предупреждения, выдаваемые сервером, а это отбирает много времени. Еще одним недостатком является серьезная специализация систем, использующих знания, которые

фактически являются актуальными лишь для конкретных операционных систем, версий, платформ и приложений. И еще один недостаток – такие системы не в состоянии обнаружить атаки от авторизованных пользователей, имеющих большие привилегии в сети, потому что уязвимости ими, фактически, не используются. Но, с другой стороны, число ложных срабатываний сведено к минимуму, что является основным достоинством подхода, основанного на знаниях. Кроме того, такие NIDS способны предоставлять очень детализированные отчеты о потенциальных попытках взлома, что дает возможность персоналу предпринимать адекватные предупреждающие или корректирующие действия.

Накопленные сведения об атаках и уязвимостях реализуются системами путем использования так называемых сигнатур – стандартных шаблонов, с которыми сравниваются сетевые пакеты. Сигнатуры бывают трех типов: строковые (string signatures), сигнатуры портов (port signatures) и сигнатуры заголовков (header condition signatures).

Строковые сигнатуры (например, регулярные выражения) сравниваются с содержимым пакетов для отыскания в них определенных строк, указывающих на потенциальную атаку. Здесь таится основная опасность ложных срабатываний системы в связи с тем, что если используются слишком простые сигнатуры, то они рано или поздно совпадут с содержимым совершенно безобидного пакета. Поэтому строковые сигнатуры необходимо тщательно продумывать и логически объединять их.

Сигнатуры портов следят за соединениями с определенными критическими портами хостов для того, чтобы затем, вероятно, эффективно применить строковые сигнатуры. Примером таких портов можно назвать Telnet (TCP-порт 23), FTP (TCP-порты 20 и 21), SUNRPC (TCP/UDP порты 111). Особое внимание уделяется неиспользуемым портам, и любая попытка соединения с ними тщательно проверятся.

В отличие от строковых сигнатур, сигнатуры заголовка следят не за данными пакета, а за его служебными полями и проверяют их на допустимость. Например, очевидно, что TCP-пакет не должен одновременно запрашивать установление и разрыв соединения (установлены флаги SYN и FIN). Еще один пример атаки, за которой следят сигнатуры портов, –WinNuke, посылающая в сеть пакеты на порт NetBIOS (139) с установленными указателями Urgent или Out-of-Band, что приводит к "синему экрану" для некоторых Windows.

1.2. Системы, основанные на шаблонах поведения (behavior based ids)

Подобные системы в своей работе основываются на предположении, что атаки на вычислительную машину вызывают заметные изменения в ее поведении, следовательно, их можно обнаружить, сравнивая ее текущее поведение с некоторыми шаблонами. Критериями поведения можно назвать загруженность CPU, памяти, дискового пространства, число сетевых соединений и открытых файловых дескрипторов и многое другое. Как видно из перечисленных выше параметров, шаблоны поведения лишь ограниченно могут быть применены к сетевой среде, они лучше подходят для контроля на уровне хоста.

Модели нормального поведения определяются весьма разнообразными методами, от ручного “вбивания” допустимых границ тех или иных параметров в систему до так называемого режима обучения, когда все текущие параметры системы запоминаются как допустимые. Впоследствии IDS следит за поведением подконтрольной системы и в случае каких-либо отклонений выполняет заранее заданное действие. Иными словами, любое

состояние параметров системы, не вписывающееся в рамки шаблонов IDS, считается критическим, что обуславливает хорошую полноту системы (т.е. процент обнаруженных атак), но и является причиной основного недостатка поведенческого подхода – большой процент ложных срабатываний.

Полнота является не единственным аргументов “за”. Есть еще и возможность распознавания неизвестных атак, и высокая межплатформенная переносимость, и возможность распознавания атак от авторизованных пользователей, вовсе не использующих уязвимости системы. Однако, по-прежнему, поведенческие принципы делают NIDS “параноидальной”, что сильно затрудняет работу с ней. Притом, если в режиме обучения произошла какая-нибудь атака, она будет зафиксирована как допустимая.

Наконец отметим, что лишь очень немногие современные NIDS используют этот подход, несмотря на то, что один из идеологов IDS D. Denning в своей статье “An Intrusion Detection Model” (IEEE transactions on software engineering) позиционировал его как обязательное требование к IDS.

2. Ограничения у систем обнаружения атак

Рассмотрим следующие основные ограничения:

2.1. Ограничения по ресурсам

NIDS страдают от того факта, что они требует немалые ресурсы для анализа трафика – мощности центрального процессора, оперативной памяти и дискового пространства для ведения журнала. К примеру, хакеры могут растянуть процесс сканирования сети на месяцы, а из-за ограниченного размера журнала NIDS не сможет сравнивать прошедшие события с настоящими и прийти к нужным выводам. В итоге, в настоящее время NIDS не могут поддерживать сильно загруженные сегменты, например, 100-Мбит/сек, и являются подходящими лишь для незначительно загруженных сегментов сетей или WAN-связей.

2.2. IP-фрагментация также значительно усложняет задачу

Для оптимальной передачи данных через сеть, на четвертом уровне модели OSI источника они фрагментируются в пакеты IP, которые снова собираются воедино на том же уровне приемника. Для разных пакетов сетевые устройства могут выбрать разные физические маршруты, поэтому приемник получает пакеты в произвольном порядке. Все это приводит к очень серьезным последствиям для серверов NIDS, которым было бы значительно проще ограничиться работой с трафиком на сетевом уровне, экономя тем самым вычислительные ресурсы. Вместо этого приходится работать и на четвертом, транспортном уровне OSI, или же сохранять все фрагменты, что требует больших ресурсов. Последний подход провоцирует атаки типа "отказ в обслуживании", когда в результате резкого возрастания объема трафика через сервер NIDS система вынуждена пропускать пакеты без анализа.

2.3. Трудность применения IDS в коммутируемых средах

Она основана на свойстве коммутаторов (switches) распределять трафик в зависимости от его направления. В то время как концентратор (hub) просто разослал бы пришедший пакет во все свои активные порты, в момент прихода IP-пакета на порт коммутатора, он поднимает его вверх по стеку OSI до третьего уровня (что, кстати, дает теоретическую

возможность встраивать в подобные устройства NIDS) и анализирует сведения о приемнике. В итоге делается вывод, на какой порт необходимо отправить данный пакет, после чего пакет вновь спускается вниз по стеку OSI и пересылается только на нужный порт. Этим приемом эффективно уменьшается загрузка сети и размер коллизионных доменов, однако создаются проблемы с установкой серверов NIDS – ведь теперь они способны захватывать и анализировать трафик только в своем сегменте.

Рассмотрим на простом, но распространенном примере, как решаются такие проблемы. Пусть мы желаем контролировать трафик, поступающий от коммутатора на машину с ресурсом (см. рис. 1).

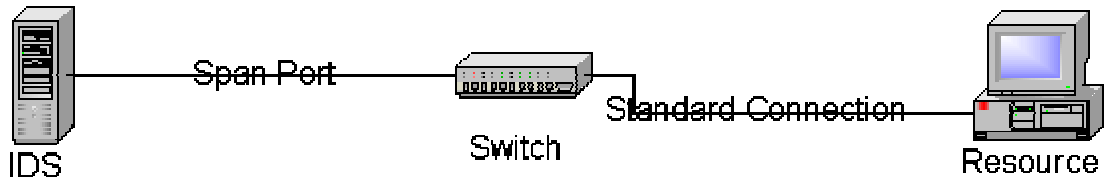

Figure-1

Простейший вариант – локальное использование концентратора, дублирующего трафик с коммутатора на NIDS (рис. 2). Это не создаст дополнительной нагрузки на сегмент, поскольку маловероятно, что сервер NIDS будет активно посылать в нее данные и провоцировать коллизии.

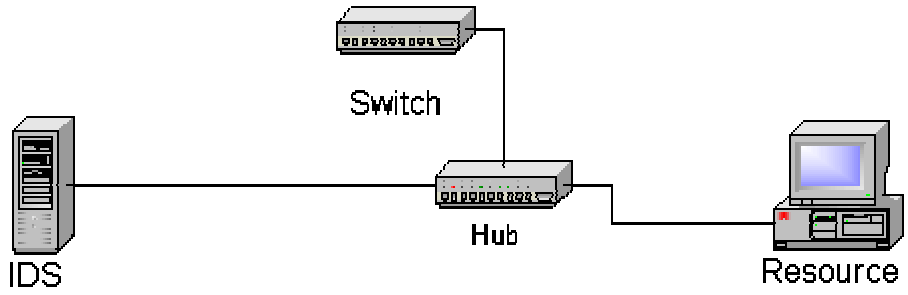

Figure-2

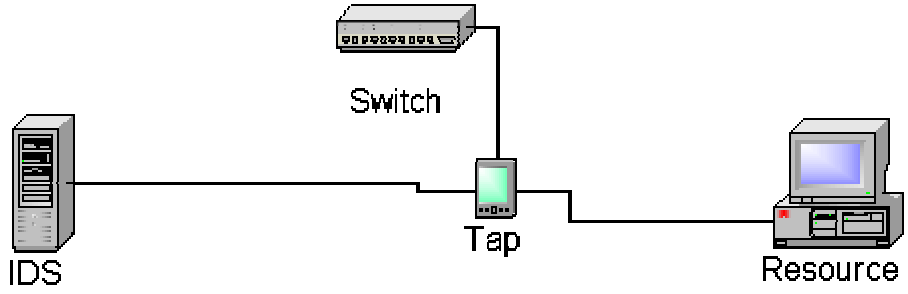

Figure-3

Вместо концентраторов могут использоваться разветвители (taps). Будучи однонаправленными устройствами, они пропускают трафик между коммутатором и ресурсом к серверу IDS, однако обратно – нет (рис. 3).

Разветвители в сочетании с концентраторами могут использоваться для создания более сложных архитектур, когда один сервер NIDS обслуживает несколько машин с ресурсами (рис. 4).

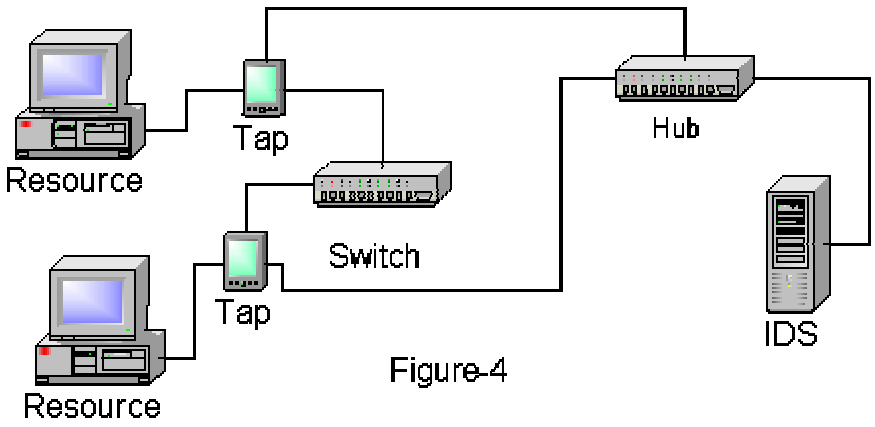

Figure-4

Как уже было упомянуто выше, некоторые продукты NIDS могут быть внедрены непосредственно в сами сетевые устройства типа коммутаторов. Однако в случае сильно загруженных сегментов это решение нежизнеспособно, ибо объединительные платы на задней панели коммутатора работают на скоростях в несколько Гбит/сек. В таких условиях можно проводить только поверхностный анализ пакетов, что легко обходится изощренными хакерами.

2.4. Распределенное по времени сканирование

Из-за большого объема трафика NIDS с трудом могут сопоставлять события, прошедшие, скажем, сутки назад и текущие. В итоге подконтрольные хосты по-прежнему остаются уязвимыми к распределенным по времени сканированиям (ping sweeps или port scans), при которых нарушители сканируют один порт/адрес через большие промежутки времени. Решением этой проблемы может стать приобретение носителей информации большого объема, однако часто информация, полученная хакерами таким способом, является неактуальной, что позволяет не заботиться о распределенных по времени сканированиях.

3. Типичные точки расположения NIDS в сети

Два простых примера расположения серверов NIDS:

3.1. У важного узла

Иногда NIDS устанавливают перед критичными хостами, например, серверами баз данных, для контроля трафика только с этим хостом. Причина, по которой так делают – максимально возможное снижение трафика через систему. Это решение вообще граничит с системами обнаружения атак на уровне хоста, которые устанавливаются на каждый защищаемый сервер и обнаруживают атаки именно на него.

3.2. Магистраль локальной сети

Некоторые производители интегрируют IDS с коммутаторами и маршрутизаторами (например, ODS Network и CISCO). Однако, как было упомянуто выше, для детального разбора пакета нужны значительные ресурсы, которые проблематично совместить с сетевыми устройствами. Поэтому наиболее эффективным выбором будет упрощенный вариант NIDS, который позволяет обнаруживать только базовые виды атак.

Источники:

1. “FAQ: Network Intrusion Detection Systems”, Robert Graham, March 2000
   http://www.robertgraham.com/pubs/network-intrusion-detection.html
2. “CERIAS Hotlist”, Perdue University
   http://www.cs.purdue.edu/coast/hotlist/intrusion_detection
3. “An Analysis Of Security Incidents On The Internet”, Carnegie Mellon University
   http://www.cert.org/research/JHThesis/Start.html
4. “Intrusion Detection FAQ”, The SANS Institute
   http://www.sans.org/resources/idfaq/
5. “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection”, T. Ptacek and T. Newsham, Secure Networks, January 1998.