Список использованных источников

1. Алиев А.Т. Стеганографический метод синонимичных преобразований открытого текста с учетом контекста / А.Т. Алиев, А. Н. Щербакова // Материалы III Международной научно-практической конференции / под общей ред. О. Н. Жданова, В. В. Золотарева; Сиб. гос. аэрокосмич. ун-т. – Красноярск, 2009. – 144 с.
2. Барсуков В. С. Компьютерная стеганография вчера, сегодня, завтра. Технологии информационной безопасности 21 века / В. С. Барсуков // Специальная техника. – 1998. – № 4-5.
3. Грибунин В. Г. Цифровая стеганография / В. Г. Грибунин [и др.]. – М.: СОЛОН-Пресс, 2002. – С. 2.
4. Информационная безопасность [Электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/wiki.
5. Орлов В. В. Активная стеганография в сетях TCP/IP / В. В. Орлов, А. П. Алексеев // Инфокоммуникационные технологии. – 2009. – Т. 7. – № 2.
6. Стеганография [Электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/wiki.
7. Mazurczyk W. Hiding Information in Retransmissions [Электронный ресурс] / Mazurczyk W., Smolarczyk M., Szczypiorski K. – Режим доступа: http://arxiv.org/abs/0905.0363.
8. Handel T., Sandford M. Hiding Data in the OSI Network Model, Proc. 1st International Workshop. Information Hiding, 1996. – P. 23-38.
9. Krätzer C., Dittmann J., Lang A., Kühne T. WLAN Steganography: a First Practical Review, Proc. 8th ACM Multimedia and Security Workshop., September 2006.
10. RFC 791 – Протокол IP (Internet Protocol) [Электронный ресурс]. – Режим доступа: http://rfc2.ru/791.rfc.
11. Transmission Control Protocol. Программная спецификация протокола. (RFC 793) [Электронный ресурс]. – Режим доступа: http://www.protocols.ru/files/RFC/rfc-793.pdf.

УДК 004.056.5:004.057.42

О.О. Фатіков, магістрант
В.В. Соломаха, ст. викладач
Чернігівський державний технологічний університет, м. Чернігів, Україна

КОМП'ЮТЕРНА СИСТЕМА ДЕТЕКТУВАННЯ НЕЯВНИХ МЕРЕЖЕВИХ АТАК

У статті розглянуто розробки архітектури комп'ютерної системи детектування й документування мережевих атак з використанням обманних методів.

Вступ. Одним з перспективних напрямків у побудові систем захисту інформації в наш час вважається застосування в системах захисту інформації обманної тактики [1; 2].

Обманна тактика захисту інформації дозволяє відвертати увагу порушників від основних цілей, заманюючи на неправильні інформаційні об'єкти, робити збір інформації щодо приймання, тактики й мотивації зловмисників, здійснювати їхню ідентифікацію й викриття.

Для виконання цих завдань можуть бути використані обманні системи, які називають також неправильними інформаційними системами, імітаторами інформаційних систем або системами-пастками. Основними функціями таких систем є залучення й утримання уваги зловмисників на неправильних інформаційних цілях, введення зловмисників в оману, виявлення й фіксація дій порушників, їх контроль, а також збір і агрегація даних про дії порушників з різних джерел.

Обманні системи являють собою програмно-апаратні засоби забезпечення інформаційної безпеки, що реалізують функції приховання й камуфляжу, що захищаються, інформаційних ресурсів, а також дезінформації порушників [1; 2; 3].

У цей час знаходять застосування два основні способи побудови обманних систем.

Системи, побудовані першим способом, називаються системами з низьким рівнем взаємодії. Ці системи емулюють програмно комп'ютери, операційні системи й сервіси.

Системи другого типа називаються системами з високим рівнем взаємодії. Вони являють собою фізичні сервери або віртуальні машини, з установленими ОС і прикладним програмним забезпеченням, злом якого може становити інтерес для зловмисника (веб-сервіси, СКБД і т. д.).

Кожний з підходів має свої недоліки й свої переваги.

Система з низьким рівнем взаємодії більш надійна, тому що вона не містить у собі реальних сервісів. Однак з цієї ж причини рівень реалізму такої системи досить низький. З іншої сторони, системи з високим рівнем взаємодії надають найвищий рівень реалізму, однак менш безпечні [2; 3].

Таблиця 1

Порівняльні характеристики двох типів облудних систем

	Системи з низьким рівнем взаємодії	Системи з високим рівнем взаємодії
Ступінь вірогідності	Низька	Висока
Реальна ОС	Немає	Так
Інформація, що збирається	Про з'єднання	Про всі дії зловмисника
Ступінь ризику	Низька	Висока

Для підвищення ефективності захисту інформації буда розроблена архітектура системи детектування неявних мережевих атак, яка об'єднує у собі способи захисту інформації, реалізовані в обманних системах низьким рівнем взаємодії й обманних системах з високим рівнем взаємодії. Такий підхід дозволяє об'єднати якості розглянутих типів систем.

Короткий опис методу. У ході дослідження були застосовані такі методи:

1. Аналіз проводився методом системного аналізу.

2. Під час розробки системи був застосований висхідний метод проектування: на основі відомих програмно-апаратних засобів була розроблена система детектування неявних мережевих атак.

3. Для моделювання використовувався метод імітаційного моделювання, який дозволяє оцінити параметри моделей залежно від часу. Результати моделювання підтвердили працездатність розробленої системи.

У результаті моделювання були отримані порівняльні результати роботи розробленої системи.

Розробка концепції комбінованої обманної системи. Системи з низьким рівнем взаємодії надають більше можливостей щодо створення розгалужених мереж. У той же час у них є вбудовані можливості щодо маршрутизації запитів до заданих мережевих адресів. Скориставшись цим функціоналом, можна побудувати систему, яка буде включати два рівні:

а) перший рівень. Обманна система з низьким рівнем взаємодії, яка емулює велику розгалужену мережу;

б) другий рівень. Обманна система з високим рівнем взаємодії, яка обробляє запити від зловмисника й збирає інформацію про його дії.

Для реалізації запропонованого способу побудови, система детектування неявних мережевих атак повинна містити у собі такі компоненти:

а) Легкі обманні системи з низьким рівнем інтерактивності. Вони виступають у ролі фільтра для наступного рівня – набору обманних систем з високим рівнем інтерактивності.

б) Обманні системи з високим рівнем інтерактивності. Набір віртуальних машин Vmware, на яких запущені відповідні операційні системи й служби. Відфільтрований трафік з «сенсорів» перенаправляється сюди для подальшого аналізу й збору інформації про погрозу.

в) Керування й контроль. Механізм, який надає уніфікована вистава про поточний стан системи, включаючи оберт трафіку, корисне навантаження і т. д. Він також надає інформацію про продуктивність віртуальних машин.

Усі адреси з адресного простору повинні бути оснащені облудними системами з низьким рівнем інтерактивності. Вони відіграють роль фільтрів трафіку, що не цікавить нас, відповідаючи на запити, локально й не передаючи трафік далі по мережі. Трафік, що цікавить, передається на віртуальні машини для подальшого аналізу. Нецікавий трафік можна класифікувати як усі пакети, які не належать існуючому з'єднанню, Syn-Пакети, які не роблять «потрійного рукостискання»; а також навантаження, яке неодноразово спостерігалося в минулому. Незважаючи на те, що цей механізм не дозволяє розрізняти всі погрози, він служить для ефективного оповіщення про нові погрози.

Як обманні системи з високим рівнем інтерактивності, пропонується використовувати віртуальні машини з необхідним ПЗ (операційні системи, служби і т. д.). Мережні настроювання сервера, на якому розташовані віртуальні машини, не дозволяють йому зв'язуватися із зовнішнім. Однак замість того, щоб блокувати вихідні з'єднання, вони перенаправляються назад на «важкі» обманні системи.

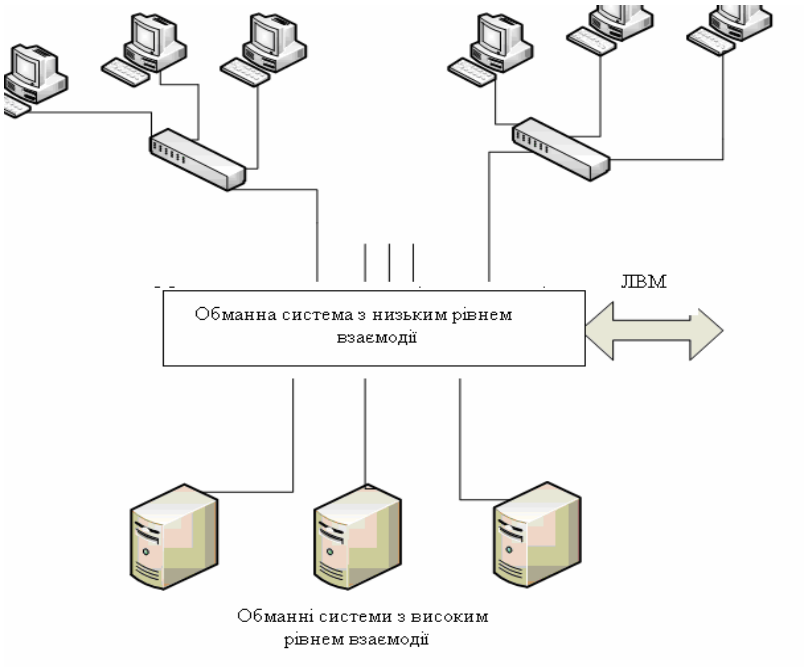

Рис. 1. Комбінована обманна система

Якщо «мережний хробак» заражає одну з машин, він може продовжувати заражати доти, поки є «чисті» ВМ. Такий механізм дозволяє спостерігати за поведінкою «хробака» і шлях його поширення в контрольованім середовищі. Як тільки буде виявлений такий тип поширення, функції роботи з контрольними крапками Vmware дозволять не тільки зберегти стани із зараженою системою й супутню інформацію, але й відкотитися до нормального стану для подальшого використання під час аналізу інформації. Для того щоб зрозуміти, що ВМ перейшла в стан відмінний від нормального, слід спостерігати за її мережевими з'єднаннями, а також змінами у файловій системі.

Останній компонент архітектури – це підсистема керування й журналювання. Вона повинна вести статистику трафіку з обманних систем з низьким рівнем взаємодії й здійснювати моніторинг навантаження на віртуальні машини. Підсистема керування відповідає за аналіз прийнятих даних на предмет аномальної поведінки, наприклад, поширення хробака. Це досягається за допомогою двох фаз аналізу, які поєднуються

наприкінці. На першій фазі перевіряється мережна активність «сенсорів». Якщо збільшується обсяг трафіку для певного порту, це швидше за все означає, що зловмисник намагається одержати доступ до сервісу, розташованого на цьому порту. На другій фазі перевіряється поведінка віртуальних машин. Якщо на віртуальних машинах створюються невідомі мережні з'єднання, це може означати інфікування мережевим хробаком.

Архітектура системи. Враховуючи всі блоки, розглянуті вище, розроблена остаточна архітектура системи захисту від мережевих атак.

Для побудови обманної мережі використовуються адреси з діапазону 192.168.0.0-255.255.255.255. Цей діапазон звичайно застосовується в локальних мережах середніх розмірів.

Як було сказано раніше, для відсівання непотрібних пакетів використовується програма Honeyd [1].

Honeyd самостійно обробляє порти, які використовуються службами, які не цікаві з погляду дослідження. Порти, на які приходять цікаві, з погляду безпеки, пакети перенаправляються на віртуальні машини, на яких установлені відповідні програми для журналювання подій в ОС.

Приклад конфігурації:

```
create default
set default default tcp action block
set default default udp action block
set default default icmp action block

create linux
set linux personality "Linux 2.4.20"
set linux ethernet "dell"
set linux default tcp action reset
add linux tcp port 80 scripts/web.sh
add linux tcp port 23 "/usr/share/honeyd/scripts/telnet.sh"
#add linux tcp port 23 "echo Welcome to telnet!"
add linux tcp port 443 proxy darkweb.portaone.com:443
add linux tcp port 22 proxy 192.168.1.8:22
add linux tcp port 3306 proxy 192.168.1.8:3306
dhcp linux on p2p1.
```

У цій конфігурації створюється віртуальний комп'ютер із установленою ОС Linux.

На ньому емулюються сервіси: веб-сервіс, сервер протоколу telnet, сервер протоколу ssh, а також СКБД mysql.

У цьому випадку веб-сервіс емулюється на рівні honeyd і не перенаправляється на відповідну віртуальну машину. За це відповідає спеціальний скрипт. Цей скрипт генерує стандартний http-відповідь просту html-сторінку:

```
add linux tcp port 80 scripts/web.sh.
```

Сервіси й ОС обрані для конфігурації, тому що є популярним розв'язком в області побудови серверів і роботи з базами даних.

При спробі підключення до сервісу Mysql з'єднання буде переспрямовано на віртуальну машину, де запущений справжній СКБД. Таким чином, можна буде відстежити подальші дії зловмисника за допомогою вбудованих засобів Mysql, таких як mysql bin log.

Також відкритий порт для ssh-з'єднання, який теж переспрямований на віртуальну машину:

```
add linux tcp port 22 proxy 192.168.1.8:22.
```

Для вивчення поведінки зловмисника в цьому випадку, використана програма відстеження натискання клавіш.

На рисунку 2 зображена схема взаємодії між компонентами.

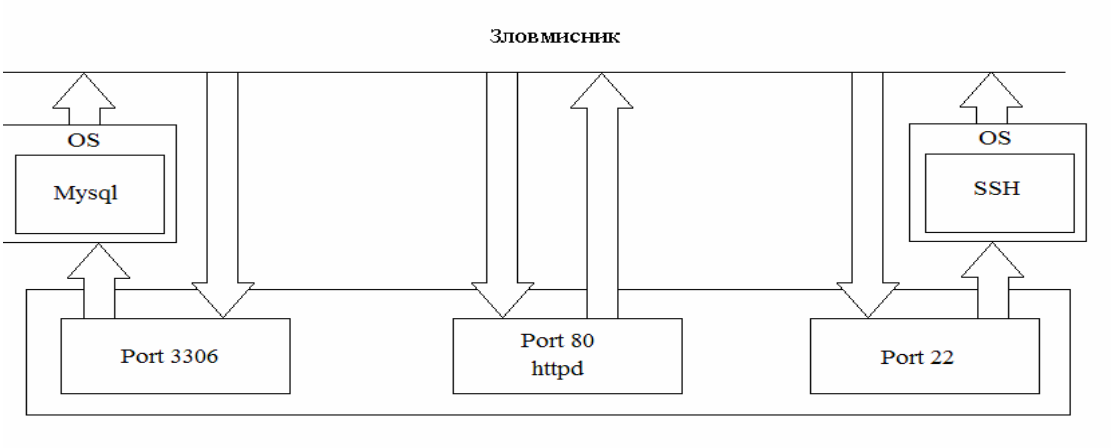

Рис. 2. Структура взаємодії між сервісами й зовнішнім світом

Таким чином, відповіді на запити по SSH і mysql зловмисник одержує від віртуальних машин. При цьому йому надається повна свобода дій, а інформація про його діяльність записується в текстовий файл.

Відповіді на запити з протоколу http генеруються прямо програмою honeyd. Тому окремої віртуальної машини для цього не потрібно.

На підставі всіх розглянутих блоків системи окремо розроблена результуюча архітектура системи – рисунок 3.

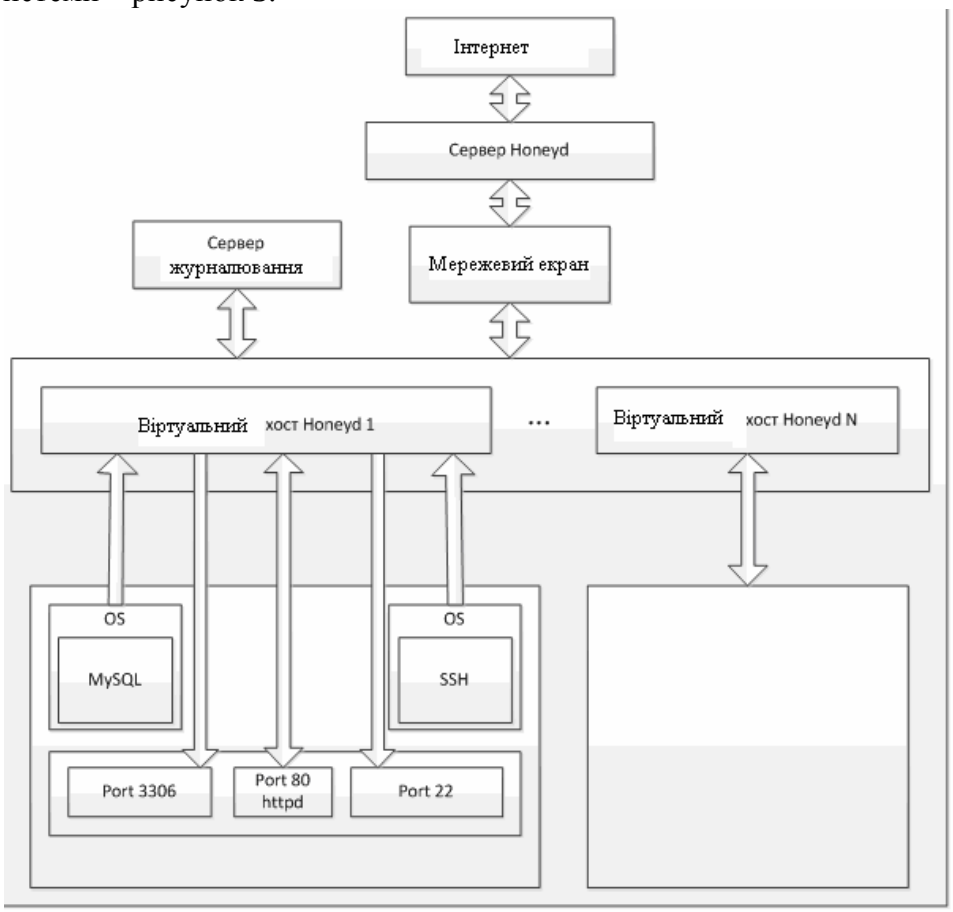

Рис. 3. Архітектура системи

Побудова моделі розробленої системи. Модель системи була створена на персональному комп'ютері з використанням віртуальних машин Vmware. Для тестування була обрана така конфігурація: 384 MB RAM, Host-Only Network Adapter. Таким чином, доступ у зовнішню мережу заблокований для обманних систем, які сконфігуровані на віртуальних машинах.

На віртуальній машині з honeyd була встановлена програма Snort.

Для перевірки працездатності була проведена проста bruteforce-атака з протоколу ssh (спроба створити більш 3-х з'єднань за одну хвилину).

Реакція snort на атаку з ssh-протоколу:

```
[**] [1:2003068:2] BLEEDING-EDGE Potential SSH Scan OUTBOUND
[**]
[Classification: Attempted Information Leak] [Priority: 2]
01/27-03:43:35.440897 192.168.17.132:40513 ->
192.168.17.131:22
TCP TTL:64 TOS:0x0 ID:42875 Iplen:20 Dgmlen:60 DF
******S* Seq: 0xe8F28737  Ack: 0x0  Win: 0x16D0  Tcplen: 40
TCP Options (5) => MSS: 1460 Sackok TS: 467020 0 NOP WS: 5.
```

Snort дає текстовий опис події, адреси комп'ютерів, між якими була спроба з'єднання, а також параметри отриманого мережевого пакета.

Зв'язок між системами з високим і низьким рівнем взаємодії:

```
set default default tcp action block
set default default udp action block
set default default icmp action block

create linux
set linux personality "Linux 2.4.20"
set linux ethernet "dell"
set linux default tcp action reset
add linux tcp port 80 scripts/web.sh
add linux tcp port 23 "/usr/share/honeyd/scripts/telnet.sh"
add linux tcp port 22 proxy 192.168.17.130:22
add linux tcp port 3306 proxy 192.168.17.130:3306

#dhcp linux on eth1
bind 192.168.1.11 linux
bind 192.168.1.10 linux.
```

Постійно створюються хости, які емулюють більшу локальну мережу. З'єднання, що приходять із зовнішньої мережі, перенаправляються на невелику кількість систем з високим рівнем взаємодії. У такий спосіб ресурси витрачаються більш ощадливо, ніж у випадку з використання тільки віртуальних машин або фізичних серверів.

Для перенапрямку використовується директива «proxy». У наведеній конфігурації використовується одна віртуальна машина, на якій установлена система з високим рівнем взаємодії.

Наведена в прикладі конфігурація може розширюватися практично нескінченно, додаванням хостів, служб і т. і.

Висновки дослідження. Ступінь вірогідності порівняна зі ступенем вірогідності систем з високим рівнем взаємодії досягнута при суттєво менших матеріальних витратах.

На рисунку 4 зображена порівняльна діаграма використання апаратних ресурсів системою з високим рівнем взаємодії й комбінованою системою.

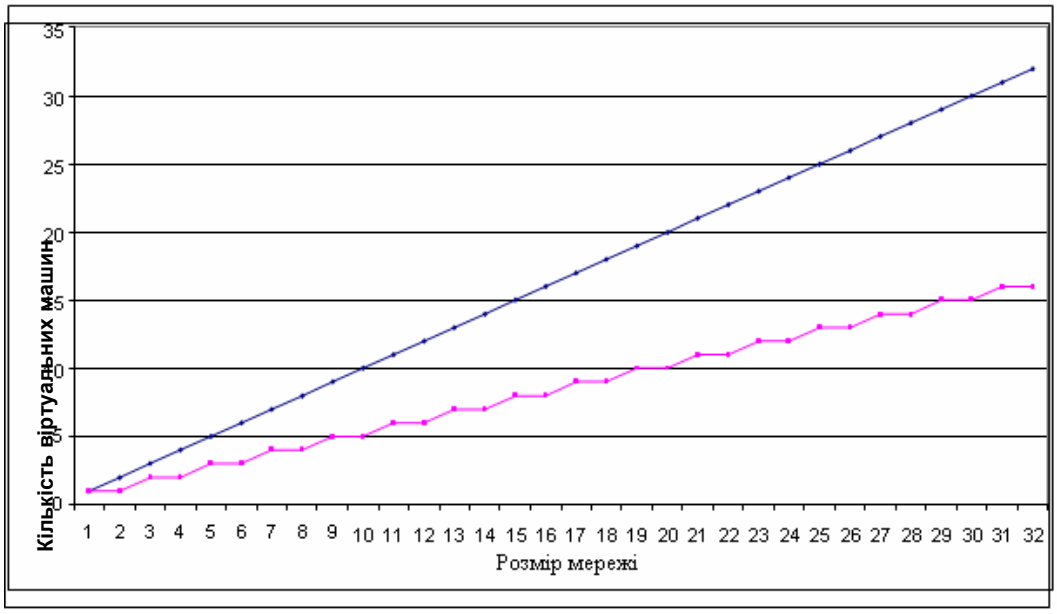

Рис. 4. Порівняльна діаграма двох систем

У цьому випадку на кожну віртуальну машину припадає по два віртуальних хоста. Можливе збільшення кількості віртуальних хостів.

За рахунок використання реальних сервісів і ОС розроблена система надає рівень вірогідності порівняний із системами з високим рівнем взаємодії. Це дає більш широкі можливості щодо відстеження дій і збору інформації про активність зловмисників.

На підставі аналізу розроблена структура облудної системи захисту інформації, яка поєднує у собі два типи захисних систем: з високим рівнем взаємодії й з низьким рівнем взаємодії. Це дозволяє зменшити витрати на установку системи захисту, тому що система з низьким рівнем взаємодії відфільтровує зайвий трафік і зменшує необхідну кількість систем з високим рівнем взаємодії. Запропонована архітектура обманної системи, на відміну від існуючих, надає новий спосіб зменшення навантаження на мережу, що емулюється й дозволяє створювати більші віртуальні мережі з високим рівнем вірогідності.

Список використаних джерел

1. Спитцнер Л. Honeynet Project: ловушка для хакеров / Л. Спитцнер // Открытые системы. – 2003. – № 7-8.

2. Лукацкий А. В. Обнаружение атак / А. В. Лукацкий. – СПб.: БХВ-Петербург, 2003.

3. Котенко И. В. Прототип ложной информационной системы / И. В. Котенко, М. В. Степашкин // Методы и технические средства обеспечения безопасности информации: тезисы докладов XI Российской научно-технической конференции (по Северо-западному региону). – СПб.: Издательство СПБГПУ, 2003.