2). Коливання надлишковості пов'язані із зміною тематики, професійної і стилістичної орієнтації текстів. Мовні і художні тексти показують низький рівень надлишковості (72 % і нижче), а публіцистична та науково-технічна мова має надлишковість на рівні 75 % (інколи до 85 %).

ЛІТЕРАТУРА

1. Шеннон К. Работы по теории информации и кибернетике. – М.: Наука, 1973. – С.68 – 128, 236 – 273, 441 – 483.

2. Белоногов Г.Г., Фролов Г.Д. Эмпирические данные о распределении букв в русской письменной речи // В сборнике «Проблемы передачи кибернетики». – 1963. – Вип. 9. – С. 287 – 305.

3. Пиотровский Р.Г. Информационные измерения языка. – Л. Наука, 1968. – С. 17 – 81.

4. Статистика речи. Сборник. Отв. редактор Пиотровский Р.Г. – Л.: Наука, 1968. – С.50 – 60, 228 – 230.

5. Кригін М.Ю., Широков В.А. Дослідження інформаційно-статистичних властивостей українського тексту. Математичні машини і системи, 2000, №1, с.120 -127.

6. Сушко С.О., Фомичова Л.Я., Барсуков Є.С. «Частоти повторюваності букв і біграм у відкритих текстах українською мовою» . – Захист інформації. –2010. – №3, С. 94-102.

Надійшла: 27.07.2012 р.

Рецензент: д.т.н., професор Петров О.С.

УДК 004.056.53:004.492.3 (045)

Гнатюк С.О., Волянська В.В., Карпенко С.В.

СУЧАСНІ СИСТЕМИ ВІРТУАЛЬНИХ ПРИМАНОК НА ОСНОВІ ТЕХНОЛОГІЇ HONEYPOT

У цій статті проведено аналіз існуючих систем віртуальних приманок на базі технології honeypot. Аналіз показав еволюцію honeypot-систем від Low-Interaction Honeypots до найсучасніших Gen III Honeynets і вказав на недоліки існуючих рішень. Крім того, проведено класифікацію honeypot-систем за ознаковим принципом. У подальшому ці результати можна використати для розробки honeypot-систем з метою підвищення ефективності роботи систем управління інцидентами інформаційної безпеки.

Ключові слова: віртуальна приманка, технологія honeypot, виявлення вторгнень, honeynet, ознаковий принцип класифікації.

Вступ. Достатньо довго в протистоянні «напад-захист» практикувалася своєрідна покрокова стратегія – зловмисники користувалися однією «діркою» захисту і з часом її закривали, тоді вони шукали іншу – її згодом також закривали і т.п. Такий аналог гри в шахи, де партія може тривати як завгодно довго, вимагає від захисту колосальних витрат часу і ресурсів, тим паче в нападника майже завжди є можливість адекватно відреагувати на захисні заходи. Зважаючи на це, сторона захисту повинна «грати на попередження», цим самим мінімізуючи ризик вторгнення. Саме реалізація такої ідеї лежить в основі використання віртуальних приманок – так званих, *honeypot-систем* (від англ. – «горщик з медом»). Мета їх функціонування – бути атакованими або сканованими зловмисниками для вивчення стратегії останніх, визначення кола їх засобів, за допомогою яких можуть бути нанесені удари по реальних об'єктах безпеки. Метод реалізації віртуальної приманки не принциповий – це може бути як спеціально розгорнута цілісна мережа так і один єдиний емульваний мережевий сервіс, основним і першочерговим завданням якого є зацікавлення (привернення уваги) порушника [1].

Концепція віртуальних приманок бере свій початок з робіт К. Столла і Б. Чесвіка [2, 3]. Ця концепція була реалізована в ряді ранніх продуктів (Deception Toolkit, CyberCop Sting, BackOfficer Friendly [4, 5]). Подальше удосконалення і розширення сфери застосування даної технології пов'язане з оформленням у 1999 р. проекту Honeynet Project [6]. Завдяки роботам таких фахівців як Л. Спітцнер [4], Н. Провос [5], Ф. Коен [7], Е. Балас [8], М. Рош [9] концепція віртуальних приманок оформилася в конкретну технологію з власною сферою застосування архітектурою і інструментарієм. Протягом

останніх років пропонуються все ширші галузі застосування honeypot-приманок. Зокрема деякі з них розгортаються просто для марнування часу і ресурсів нападників [10], інші – для зменшення активності спаму [11] чи обману зловмисників [12], ще інші – для аналізу діяльності хакерів при зламуванні систем [13] і виявленні образів (сигнатур) атак [14]. Л. Спітцнер у роботі [4] дав таке визначення *honeypot-концепції:* «**Honeypot –** це ресурс інформаційної системи, значення якого полягає в несанкціонованому та незаконному доступі до нього». Honeypot – це передусім концепція побудови системи для взлому. Таку систему визначає не конкретне програмне забезпечення (ПЗ) чи особливості конфігурації, а сама мета її побудови і розгортання. Як вище зазначалося ця мета – виявлення вторгнень і подальший аналіз атаки. Honeypot-приманка може приймати різні конкретні форми – від імітації окремої служби до симуляції систем і мереж. Через свою вразливість ці системи не повинні бути виробничими інформаційно-комунікаційними системами (ІКС), проте своєю схожістю приваблювати зловмисників. З огляду на актуальність цього напряму наукових досліджень та відсутність у науковій літературі класифікацій систем віртуальних приманок, *метою даної статті* є пошук існуючих систем на базі технології honeypot та спроба їх класифікації за ознаковим принципом.

Пошук ознак для класифікації систем honeypot. Віртуальна пастка має бути видимою і доступною для нападника – в цьому її користь, у іншому випадку не буде отримано корисної інформації. З іншого боку – ніяка атака на honeypot не відобразиться на захищеній ІКС. Для виділення необхідної інформації використовуються різні технології *пасивного прослуховування (sniffing)* і *реєстрації (logging)* [15]. Варто виділити такі дві групи honeypot-систем *(за метою функціонування)* [4]:

— *виробничі пастки-приманки (production honeypots)* – прості у використанні, фіксують лише обмежену інформацію і застосовуються, переважно, великими компаніями і комерційними організаціями;

— *дослідні віртуальні пастки (research honeypots)* – набагато складніші в розгортанні і обслуговуванні, детально фіксують усю інформацію і використовуються, переважно, дослідними військовими чи урядовими організаціями.

Те, яким чином буде спроектована система honeypot, залежить від завдань, які вона має вирішувати. Якщо необхідно вивчити мотивації поводження зловмисників, методи їхніх атак і засобів – тоді потрібно побудувати складну honeypot, що надає зловмисникові повноцінну операційну систему (ОС), з якою він буде взаємодіяти, що забезпечить високий рівень протоколювання. Якщо необхідно виявити несанкціоновану активність, таку як сканування системи, то для цих цілей можна побудувати просту honeypot, що буде імітувати мінімальні можливості й операції сервісів, записуючи лише команди взаємодії зі зловмисником [16-18]. Якщо стоїть завдання виявити мережного черв'яка для аналізу, тоді необхідно побудувати гнучкий honeypot, що буде взаємодіяти із черв'яком, збираючи елементи його активності. Наведені приклади демонструють диференційовану функціональність honeypot-систем, а також підкреслюють ті ознаки (рис. 1), за якими можна їх класифікувати.

За рівнем взаємодії зі зловмисником системи honeypot можна класифікувати таким чином, як це представлено у табл. 1. Рівень взаємодії представляє собою своєрідну міру, яка дозволяє визначати і порівнювати різні honeypot-системи. Чим більше може приманка, і чим більше поле діяльності зловмисника, тим серйозніша (критичніша) інформація може бути виділена. З іншої сторони, чим більше дозволено нападнику, тим більшу шкоду він може заподіяти [5]. *Низькорівневі* віртуальні приманки є найлегшими в установці, налаштуванні, розгортанні і обслуговуванні через свою простоту і базову функціональність. Зазвичай така технологія полягає в емуляції декількох сервісів.

Зловмисник, у свою чергу, обмежений у своїх діях взаємодією лише з даними сервісами [19]. Першочерговим завданням таких низькорівневих систем є виявлення прихованого сканування і спроб несанкціонованого підключення. Через свою обмежену функціональність, більшість з них емулюються програмно.

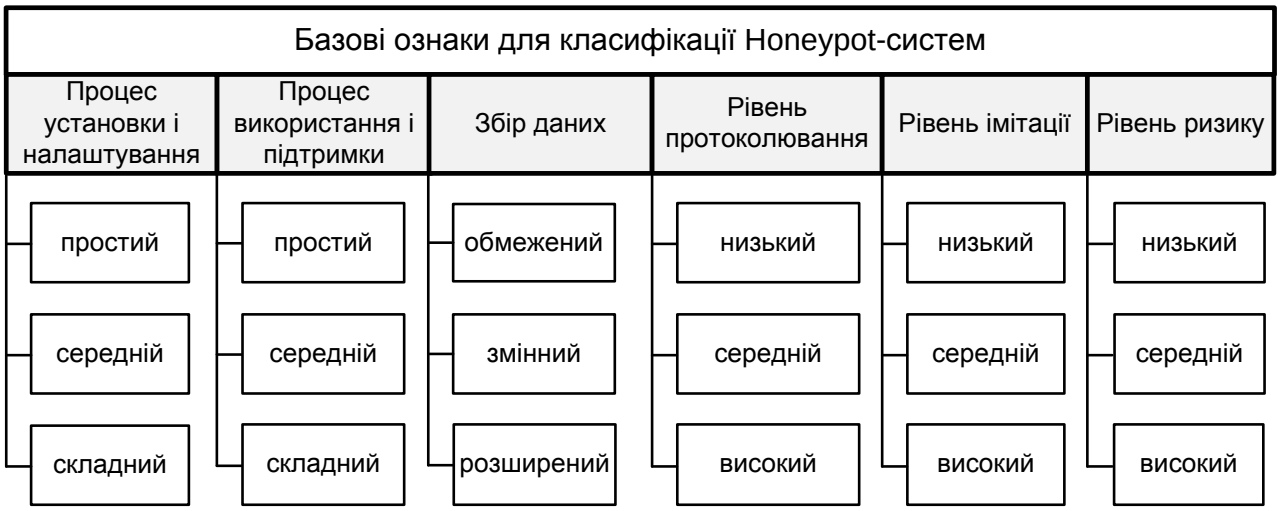


Рис. 1. Базові ознаки класифікації honeypot-систем

З огляду на простоту, низькорівневі приманки мають найменший рівень ризиків. За відсутності функціональної ОС, віртуальна приманка не може бути використана для моніторингу чи атаки інших систем. Прикладами honeypot-приманок низького рівня взаємодії є програми BackOfficer Friendly (відкрита програма) і Specter (комерційне рішення).

Таблиця 1

Класифікація honeypot-систем за рівнем взаємодії

Тип приманки	**Приклади реалізації**	**Визначаючі характеристики**
Low-Interaction Honeypots	— Deception Toolkit — LaBrea — Tiny Honeypot — BackOfficer Friendly — Specter — Honeytrap	— імітація окремих служб і сервісів; — обмежена вихідна інформація;
Medium-Interaction Honeypots	— Mwcollectd — Multipot — Nepenthes — Honeyd	— імітація роботи ОС разом з властивими їй службами; — реалізація середовища-в'язниці; — необхідність застосування додаткових захисних механізмів;
High-Interaction Honeypots	— Argos — ManTrap — Віртуалізовані системи	— повноцінна емуляція ОС (включаючи застосування реальних систем); — детальна інформація про атаку і нападника; — застосування громіздкого механізму контролю, що включає фільтрацію трафіку і застосування СВВ (IDS, систем виявлення вторгнень);
Gen I Honeynets *Gen II Honeynets* *Gen III Honeynets*	— SCADA Honeynets — The Georgia Tech Honeynet — CADHo Project Leurre.com	— повноцінна імітація роботи мережі; — наявність внутрішньої структури і архітектури; — багаторівнева система фіксації інформації, — централізоване накопичення, аналіз і кореляція даних; — повноцінний механізм мережевого захисту;

Back Officer Friendly (надалі, BOF) – один з найпростіших варіантів honeypot, перша версія якого була розроблена в 1998 році М. Ранумом. Система BOF є функціонально простою й зрозумілою для недосвідчених користувачів [20].

Засіб BOF може бути запущений як на UNIX, так і на Windows платформах з можливістю імітації таких служб: FTP, SMTP, IMAP, POP3, HTTP, TELNET, а також троянського сервісу Back Orifice – для дистанційного адміністрування комп'ютера. В системі BOF немає якого-небудь детального настроювання її роботи, саме тому вона досить проста у використанні.

Одна з головних переваг засобу BOF – його ціна, а саме він є безкоштовним. Система BOF не здійснює детальну імітацію сервісів – як тільки відбувається з'єднання, BOF

виводить повідомлення про недоступність служби й через невеликий тайм-аут робить розрив з'єднання. Ціль BOF – моніторинг подій і збереження протоколу взаємодії [19]. У табл. 2 представлено основні переваги і недоліки даного рішення:

Таблиця 2

Характеристики honeypot-системи BackOfficer Friendly

Переваги	**Недоліки**
1) Простота в установці, налаштуванні і обслуговуванні; 2) Працює на всіх Windows- і Unix-платформах, включаючи більшість desktop- та laptop-систем; 3) Незначні ризики завдячуючи простоті.	1) Обмежений сімома портами для виявлення атак; 2) Порти не можуть бути спеціальним чином налаштовані, що підвищує ймовірність зняття «відбитку» системи (fingerprinting); 3) Відсутня можливість віддаленої реєстрації подій, попередження і налаштування.

Specter **–** комерційний засіб honeypot, створений й підтримуваний компанією NetSec. Концепція Specter схожа з BOF – зловмисникові не надається доступ до реальної ОС. Програма встановлюється в систему й імітує набір сервісів, з якими зловмисник може взаємодіяти. Зловмисник обмежений функціональністю, наданою Specter. Можливості Specter більш широкі: моніторинг ведеться на 13-ти визначених і одному вибірковому портах. Покриваючи більшу кількість портів, Specter має можливість виявити більше число різних атак. Друга відмінність полягає в тім, що Specter, як і інші honeypot, імітує сервіси, однак, відповіді цих сервісів мають куди більший вбудований реалізм, ніж у BOF. Наприклад, при підключенні до імітованого Specter сервісу HTTP можна бачити справжній Web-сервер з Web-сторінками, з якими може взаємодіяти зловмисник. Таким чином, існує можливість зміни даних сторінок, додаванням потрібного змісту, створюючи при цьому більш реалістичні дані для зловмисника. Ще однією особливістю даного засобу є не тільки імітація сервісів, але й надання додаткової взаємодії зі зловмисником. Наприклад, зловмисник має можливість скачати файл паролів і бути впевненим, що заволодів конфіденційною інформацією, тоді як даний файл буде спеціально підготовленим [21]. Крім того, Specter має можливість імітації великої кількості (більше десяти) ОС. Дана імітація виражається в системних відповідях обраних служб. Таким чином, коло атак, що виявляються, істотно розширюється. Такий honeypot може записувати взаємодію на одному, обумовленому користувачем, порту.

Ця можливість корисна для виявлення нових атак. Наприклад, якщо стали відомі деталі нової атаки або активності мережного черв'яка, то існує можливість визначити порт взаємодії honeypot для виявлення заданої активності. Ще одна унікальна особливість засобу Specter – розширені можливості налагодження. Для того щоб ускладнити процес ідентифікації зловмисником honeypot Specter, при установці можна призначити власне доменне ім'я, адресу, а також інші специфічні характеристики. Задаючи безліч опцій, що налаштовуються, можна побудувати унікальний засіб honeypot [19]. Переваги та недоліки у Specter практично такі ж самі як і у BOF (див. табл. 2), окрім можливості налаштовувати один порт для взаємодії з зловмисником.

Honeypot *середнього рівня взаємодії* надають зловмиснику більше можливостей взаємодії ніж низькорівневі приманки, але й мають меншу функціональність, як високорівневі. Вони розраховані на конкретно визначену активність, що викликає конкретно визначену реакцію, яка в той же час виходить далеко за межі низькорівневих віртуальних приманок. Середньорівневі приманки зазвичай вимагають більше часу і зусиль для установки і налаштування, ніж низькорівневі. Розгортання і обслуговування такої віртуальної приманки – це також набагато складніший процес. Так як зловмиснику надається більша можливість взаємодії, то розгортання приманки вимагає дотримання правил захищеності. Не зважаючи на це, середньо-рівневі віртуальні приманки збирають набагато більший об'єм інформації. На відміну від простого перегляду портів, вони дозволяють захоплювати шкідливий код, фіксувати активність зловмисника, вивчати його дії після отримання доступу до системи і навіть отримувати засоби злому. Ширший рівень взаємодії вимагає ускладнення роботи і збільшення ризику, проте це винагороджується отриманням

набагато детальнішої інформації. Прикладами honeypot-приманок середнього рівня взаємодії є програми Mwcollectd, Multipot, Nepenthes, Honeyd [22].

Honeyd **–** засіб, розроблений і підтримується Н. Провасом. Уперше випущений у квітні 2002 року, Honeyd є Open Source Honeypot для UNIX платформ. Honeyd був розроблений як виробничий Honeypot, використовуваний для виявлення атак або несанкціонованої активності. У зв'язку з тим, що даний honeypot надає відкриті вихідні тексти, то існує можливість власного внутрішнього настроювання, наприклад, додавання імітованих сервісів. Це означає, що даний Honeypot може взаємодіяти через будь-який порт. Honeyd виявляє активність на всіх TCP-портах; а імітовані сервіси спроектовані тільки для введення зловмисника в оману й збору його активності. Honeyd представляє кілька нових концепцій Honeypot. По-перше, не виявляються атаки, що виходять із IP-адреси самого Honeypot (на відміну від BOF і Specter). По-друге, Honeyd дозволяє зробити імітацію цілої ІКС: існує можливість налаштувати IP-адреси й ОС, які їм будуть зіставлятися. Підтримується більше число ОС – від всіх Windows-подібних реалізацій до Unix-Систем маршрутизаторів [23]. Honeyd імітує систему не тільки на прикладному рівні, а також на рівні IP-стека. При цьому ймовірність успішної ідентифікації Honeypot різко зменшується. Таким чином, Honeyd надає ще більший рівень обміну інформації зі зловмисником, завдяки чому його можна віднести до honeypot середнього рівня взаємодії [4, 19].

Віртуальні приманки *високого рівня взаємодії* – крайній випадок реалізації honeypot-технологій. Вони надають обширну інформацію про атаку, але, в той же час, є надзвичайно вимогливими в плані будівництва і обслуговування, а також приносять найвищий рівень ризику. Завдання високорівневих honeypot-приманок – надати зловмиснику доступ до справжньої ОС, де нічого не емулюється і не обмежується. Вони дозволяють детально дослідити нові засоби злому, визначити нові вразливості ОС чи ПЗ і вивчити способи зв'язку зловмисників між собою [4]. Найчастіше високорівневі приманки розміщуються всередині контрольованого середовища, наприклад за мережевим екраном. Така архітектура дуже складна в розгортанні і обслуговуванні, особливо за умови, що зловмисник не повинен здогадатися про спостереження і контроль.

Тому, вимагається великий об'єм робіт для побудови такого мережевого екрану з необхідною базою правил [5]. Через надзвичайно громіздкий механізм контролю, високорівневі honeypot-приманки дуже складні, вимагають багато часу та зусиль для установки і налаштування. Їх обслуговування, в свою чергу, також вимагає значних затрат часу і ресурсів, що включає в себе оновлення бази даних правил фільтрації і сигнатур, а також постійний моніторинг активності на віртуальній приманці [24].

Така складність приносить колосальний рівень ризику, проте слід зазначити, що правильно реалізована honeypot-приманка високого рівня взаємодії дозволяє проникнути в суть атаки як не одна інша honeypot-система. Прикладом високорівневої віртуальної приманки є комерційне рішення ManTrap, створене компанією Recourse Technologies [25].

ManTrap **–** комерційний засіб honeypot високорівневої взаємодії, створений компанією Recourse Technologies. ManTrap унікальний тим, що спроектований не тільки стати метою для зловмисників – створює високо контрольовану ОС, з якої може взаємодіяти зловмисник. Однак Recourse Technologies зробила значний крок уперед і створила логічно контрольоване оточення (так звану пастку), з якого неможливо вийти атакуючому для нападу на реальну систему. При цьому ManTrap, використовуючи відомі концепції, не створює порожню «пастку», заповнюючи її різною функціональністю. ManTrap створює «пастки», які є дзеркалами, що копіюють безпосередньо ОС. Кожна пастка – повноцінна функціональна ОС, що має всі ті ж можливості, що й справжня система. Дане наближення до реальних систем створює дуже потужне й гнучке рішення.

Кожна пастка – це свій власний віртуальний світ з невеликими обмеженнями. Існує можливість налаштування кожної пастки як реальної фізичної ОС. Можна створювати користувачів, установлювати застосунки, запускати процеси або компілювати власні бінарні файли. Коли зловмисник здійснює атаку й одержує доступ до пастки, то вона виглядає для зловмисника як справжня ОС. Він не буде впевнений, що перебуває в підробленому

оточенні, де кожна дія записується. Таким чином, зловмисник може робити всі перераховані раніше дії, взаємодіяти із пристроями й системними бібліотеками тощо [26, 27]. Інша корисна можливість полягає в тому, що ManTrap створює віртуальні оточення на одній фізичній системі. Використовуючи один комп'ютер, можуть бути створені до чотирьох пасток, тобто – чотири різних honeypot високорівневої взаємодії. Основні переваги та недоліки рішення ManTrap представлені в табл. 3.

Таблиця 3

Основні характеристики приманки ManTrap

Переваги	**Недоліки**
1) Виявляє активність на кожному з портів, використовуючи вбудований аналізатор (sniffer); 2) Дає зловмиснику повністю функціональну ОС для взаємодії; 3) Фіксує всю активність зловмисника через простір ядра, включаючи і шифрований трафік SSH; 4) Першокласні можливості реєстрації подій і ведення журналу; 5) Можливість віддаленого адміністрування і попередження по E-mail.	1) Високо-інтерактивна функціональність потенційно дозволяє зловмиснику використовувати приманку для пошко-дження інших систем; 2) Зловмисник може зняти зліпок системи чи вирватися з контрольованого середо-вища клітки; 3) Обмеженість можливістю запуску лише на системі Solaris.

Проте, ManTrap має деякі обмеження. По-перше, у силу того, що цей засіб не імітує системи, а використовує технологію «пасток», то основа всіх ОС – одна. Також, ManTrap підтримує лише деякі ОС – сам honeypot може функціонувати тільки на комп'ютері з ОС Solaris. Крім того, ManTrap використовує особливу установку Solaris з необхідними параметрами, тому поки немає можливості встановити даний засіб на комп'ютери із системами NT, BSD або Linux. У табл. 4 наведена узагальнена класифікація наведених рішень honeypot:

Таблиця 4

Узагальнена класифікація існуючих засобів Honeypot

Назва honeypot	**Процес установки і налагодження**	**Процес використання і підтримки**	**Збір даних**	**Рівень протоколювання**	**Рівень імітації**	**Рівень ризику**	**Рівень взаємодії**
BOF	простий	простий	обмежений	низький	низький	низький	низький
Specter	простий	середній	обмежений	середній	середній	середній	низький
Honeyd	середній	середній	змінний	середній	високий	середній	середній
ManTrap	складний	складний	розширений	високий	високий	високий	високий

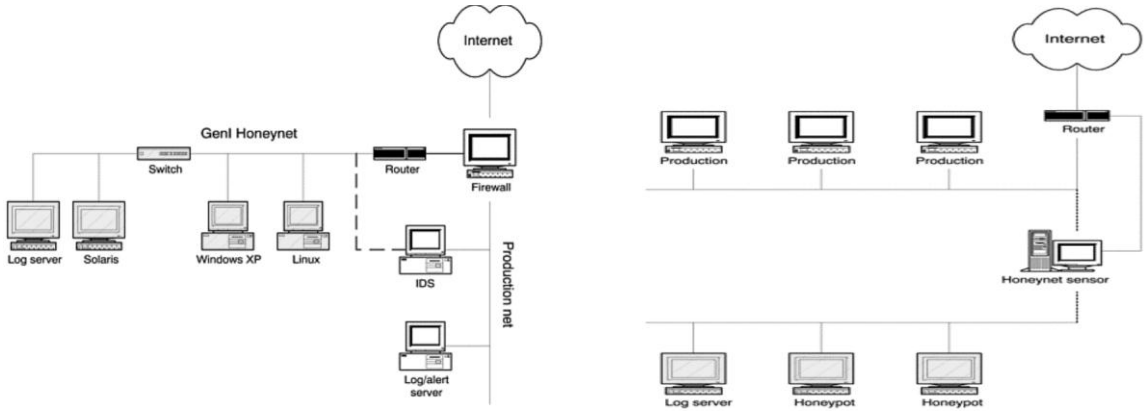
Honeynets. Логічним продовженням розширення можливостей віртуальних honeypot-приманок став подальший розвиток *концепції High-Interaction Honeypots,* що призвело до появи цілих віртуальних мереж-приманок під назвою Honeynets. Власне сама концепція була розроблена дослідною групою з 30 фахівців, що в червні 2000 р. сформували некомерційний проект Honeynet Project [6]. Дворічні дослідження цієї організації були оформлені у вигляді ряду статей під загальною назвою «Знай свого ворога». Відповідно це джерело [1] дає таке визначення: «**Honeynet –** це мережа, розміщена за реверсивним мереженим екраном, що фіксує усі вхідні і вихідні дані. Реверсивний файрвол обмежує об'єм шкідливого трафіку, що може покинути Honeynet-мережу. Ці дані зберігаються, фіксуються і контролюються. У середовищі Honeynet може бути розміщена будь-яка система, включаючи такі системи, які уже функціонують у виробничій мережі, яку покликана захищати Honeynet. Honeynet – це мережа, призначена бути атакованою і скомпрометованою для отримання відомостей про наявні та потенційні вразливості і загрози в мережі». Сьогодні існує три основні архітектури Honeynet-мереж: I-ого покоління (Gen I Honeynets); II-ого покоління (Gen II Honeynets) та III-ого покоління (Gen III Honeynets).

Gen I Honeynets. Honeynet-мережі I-ого покоління обмежені в можливостях контролю та приборкування зловмисників, проте вони демонструють достатню ефективність у виявленні автоматизованих атак і атак початківців. Передусім Gen I Honeynets фокусуються на атаках відповідно можливостей. Такі мережі-приманки достатньо легко ідентифікуються.

Архітектура Honeynet-мереж І-ого покоління досить проста – ізольована мережа розміщується за пристроєм контролю доступу до мережі, найчастіше таким служить мережевий екран (рис. 2*а*). Мета такого розміщення – забезпечити неможливість атаки не honeypot-систем. Часто поряд з Honeynet-мережею знаходиться виробнича ІКС для адміністрування і накопичення зафіксованих даних. Також, можливим є розміщення інших контролюючих пристроїв (наприклад, маршрутизатора) для додаткового контролю [4]. Фіксація активності шляхом комбінації можливостей файрволу, IDS-сенсорів і системних логів забезпечує перехоплення інформації на таких чотирьох рівнях [5]: активність в мережі, системна активність, активність програм та активність користувача.

Gen II Honeynets. Технологія Gen II була розроблена в 2002 р. і направлена на усунення недоліків попередньої. Honeynet-мережі II-ого покоління простіші в розгортанні і складніші у виявленні [1]. Як описувалося вище, технологія Gen I виконувала контроль даних за допомогою мережевого екрану, що обмежував кількість можливих вихідних підключень. Незважаючи на свою відносну ефективність таке рішення є недостатньо гнучким і забезпечує легке «зняття зліпку».

Honeynet-мережі II-ого покоління вирішують цю проблему шляхом модифікації загальної архітектури (рис. 2*б*). Перша основна розбіжність – використання єдиного Honeynet-сенсора, що об'єднує функціонал файрвола та IDS [4]. Друга основна відмінність – сама реалізація Honeynet-сенсора, що представляє собою пристрій другого рівня OSI (схожий на міст). Така реалізація значно ускладнює виявлення, так як відсутня маршрутизація пакетів, зменшення TTL і MAC-адреси пристроїв [28]. За рахунок описаних принципів Honeynet-мережа II-ого покоління може бути частиною основної виробничої мережі, а не ізольованою як в технології Gen I.



а) *б)*

Рис. 2. Типова Honeynet-мережа: а) І-го покоління; б) II-го покоління

Gen III Honeynets. Технологія Gen III реалізує подальше удосконалення і розширення можливостей контролю і аналізу даних. Модель аналізу даних базується на таки абстракціях: хости, процеси, мережеві потоки і файли (рис. 3) [8]. Такий підхід реалізується на основі використання системи Honeywall [29], розроблений фахівцями проекту Honeynet Project. Для контролю підключень і даних застосовується підхід IP Performance Measerment Working Group, що полягає в моніторингу потоків. У випадку використання Honeywall для цього застосовується система Argus [30]. Іншим удосконаленням є використання засобу пасивного зняття зліпку системи (passive fingerprinting), що ініціює TCP-підключення [31]. Для об'єднання цих двох типів даних (активність в ІКС і процесів на хості) навколо суцільної картини концепції потоків мережі використовують додаткову зв'язуючу ланку.

Для цього застосовують систему Sebek, що проводить моніторинг активності в мережі з перспективи хоста [32]. У роботі [33] виконано моделювання Honeynet Gen III у віртуальному середовищі UML, а праця [34] містить варіант віртуалізації повно-інтерактивних приманок.

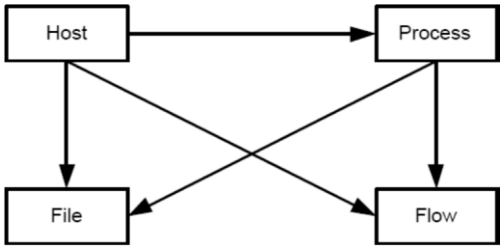


Рис. 3. Модель взаємозв'язків даних у системі Gen III

Висновки. Таким чином, у даній статті автори провели аналіз систем на базі технології honeypot від Low-Interaction Honeypots до Gen III Honeynets, результатом якого стала їх класифікації за ознаковим принципом. Подальші дослідження можуть бути пов'язані з розробкою засобу на базі технології honeypot для підвищення ефективності роботи систем управління інцидентами інформаційної безпеки.

ЛІТЕРАТУРА

1. Know Your Enemy: Learning about Security Threats / Honeynet Project. — NY.: Addison-Wesley Professional, 2004. — 800 p.

2. Stoll C. Cuckoo's Egg / C. Stoll. — NY : Pocket, 1990. — 356 p.

3. Cheswick B. An Evening with Berferd In Which a Cracker is Lured, Endured, and Studied / B. Cheswick. — NY : Management Analytics and Others, 1995. — 147 p.

4. Spitzner L. Honeypots: Tracking Hackers / L. Spitzner. — NY : Addison-Wesley Professional, 2002. — 480 p.

5. Provos N. Virtual Honeypots: From Botnet Tracking to Intrusion Detection. — NY : Addison-Wesley Professional, 2007. — 440 p.

6. Honeynet Project Blog [Електр. ресурс]: (Blog) // The Honeynet Project. — Режим доступу: http://www.honeynet.org (04.09.2012).

7. A Framework for Deception / Cohen F., Lambert D., Preston C., Berry N., Stewart C., Thomas E. — Tech. Report, 2001.

8. Balas E., Viecco C. Towards a Third Generation Data Capture Architecture for Honeynets // Workshop on Information Assurance and Security US Military Academy, West Point, NY. — IEEE, 2005.

9. Roesch M. Snort – lightweight instrusion detection for networks / M. Roesch. — LISA'99 Systems Admistration Conference, 1999.

10. LaBrea: «Sticky» Honeypot and IDS [Електр. ресурс]: (Labrea Tarpit Project) // Labrea. — Режим доступу: http://labrea.sourceforge.net (04.09.2012).

11. Hammer R. Enhancing IDS using, Tiny Honeypot / R. Hammer. — SANS Institute, 2006.

12. The Deception Toolkit [Електр. ресурс]: (The Deception Toolkit Home Page and Mailing List) // Fred Cohen & Associates. — Режим доступу: http://www.all.net/dtk/dtk.html (04.09.2012).

13. Diebold P. A Honeypot Architecture for Detecting and Analyzing Unknown Network Attacks / P. Diebold, A. Hess, G. Schafer // In Proc. Of 14th Kommunikation in Verteilen Systemen 2005. —Kaiserslautern: Technische Universitat Berlin, 2005.

14. Thakar U., Varma S., Ramani A. HoneyAnalyzer – Analysis and Extraction of Intrusion Detection Patterns & Signatures Using Honeypot // The Second International Conference on Innovations in Information Technology (IIT'05). — Indore: Institute of Technology and Science, 2005.

15. Noordin M. Honeypots Revealed / M. Noordin // SecurityDocs.com. — 2004.

16. Технология Honeypot, Часть 1: Назначение Honeypot. [Електр. ресурс] — Режим доступу: http://www.securitylab.ru/analytics/275420.php (04.09.2012).

17. Технология Honeypot. Часть 2: Классификация Honeypot. [Електр. ресурс] — Режим доступу: http://www.securitylab.ru/analytics/275775.php (04.09.2012).

18. Технология Honeypot. Часть 3: Обзор существующих Honeypot. [Електр. ресурс] — Режим доступу: http://www.securitylab.ru/contest/283103.php (04.09.2012).

19. Grimes R. Honeypots for Windows / R. Grimes. — W. : APress, 2005. — p. 424.

20. Построение виртуальных ловушек. [Електр. ресурс] — Режим доступу: http://www.securitylab.ru/analytics/216223.php (04.09.2012).

21. Specter: a Commercial Honeypot Solution for Windows, by Lance Spitzner. [Електр. ресурс] — Режим доступу: http://www.securityfocus.com/infocus/1683 (04.09.2012).

22. Wichersky G. Medium Interaction Honeypots / G. Wichersky // Workshop on Information Assurance and Security United States Military Academy, West Point, NY. — IEEE, 2006.

23. Honeyd – Network Rhapsody for You, honeyd devoted site. [Електр. ресурс] — Режим доступу: http://www.citi.umich.edu/u/provos/honeyd (04.09.2012).

24. De Andrade M. Mechanism for Automatic Digital Evidence Collection on High-Interaction // Workshop on Inf. Assurance and Security US Military Academy, West Point, NY. — IEEE, 2004.

25. Recourse Tech. ManTrap 3.0 [Електр. ресурс]: (Tech. Industry) // Software Magazine. — Режим доступу: http://findarticles.com/p/articles/mi_m0SMG/is_2_22/ai_91087662 (04.09.2012).

26. Лаурент О. Борьба со спамом с помощью системы Honeypot: Часть 1 / О. Лаурент [Електр. ресурс] — Режим доступу: http://www.securitylab.ru/analytics/216335.php (04.09.2012).

27. Лаурент О. Борьба со спамом с помощью системы Honeypot: Часть 2 / О. Лаурент [Електр. ресурс] — Режим доступу: http://www.securitylab.ru/analytics/216343.php (04.09.2012).

28. Deal R. Router Firewall Security / R. Deal. — SF. : Cisco Press, 2004. — p. 912.

29. Honeywall project site [Електр. ресурс]: (Honeywall – Trac) // The Honeynet Project — Режим доступу: https://projects.honeynet.org/honey-wall (04.09.2012).

30. Argus and Infiniband [Електр. ресурс]: (ARGUS – Auditing Network Activity) // QoSient — Режим доступу: http://www.qosient.com/argus (04.09.2012).

31. What is p0f [Електр. ресурс]: (the new p0f) // lcamtuf.coredump.cx — Режим доступу: http://lcamtuf.coredump.cx/p0f.shtml (04.09.2012).

32. Balas E. Honeynet data analysis: A technique for correlating sebek and network data / E. Balas // Workshop on Information Assurance and Security US Military Academy, West Point, NY. — IEEE, 2004.

33. Хусни. Метод разработки средств автоматизации и проектирования сетей приманок : автореф. дис. на соискание науч. степени канд. техн. наук : спец. 05.13.19 «Методы и системы защиты информации, информационная безопасность» / Хусни. — СПб., 2010. — 17 с.

34. Тимошик Н.П. Вдосконалення принципів побудови та функціонування приманок в задачах захисту комп'ютерних систем та мереж : автореф. дис. на здобуття наук. ступеня канд. техн. наук : спец. 05.13.05 «Комп'ютерні системи та компоненти» / Н.П. Тимошик. — Львів, 2010. — 22 с.

Надійшла: 17.07.2012 р.

Рецензент: д.т.н., професор Корченко О.Г.

УДК 004.9:517.978.2

Грищук Р.В., Корченко О.Г.

МЕТОДОЛОГІЯ СИНТЕЗУ ТА АНАЛІЗУ ДИФЕРЕНЦІАЛЬНО-ІГРОВИХ МОДЕЛЕЙ ТА МЕТОДІВ МОДЕЛЮВАННЯ ПРОЦЕСІВ КІБЕРНАПАДУ НА ДЕРЖАВНІ ІНФОРМАЦІЙНІ РЕСУРСИ

У статті подано методологію синтезу та аналізу диференціально-ігрових моделей та методів моделювання процесів кібернападу. Створена методологія з єдиних системних позицій дозволяє здійснювати синтез усіх диференціально-ігрових методів моделювання процесів кібернападу, які передбачають застосування комплексів відповідних моделей різного ступеню точності, від моделей оцінювання рівня захищеності – до моделей прогнозування розвитку динаміки процесу кібернападу. Застосування методології сприяє процесу інтеграції прогресивних систем інформаційної безпеки в новостворювані ІТ- технології, що, поряд з вирішенням основних завдань за призначенням, вирішують завдання інформаційної безпеки та є стійкими до прогнозованого класу кібератак і параметрів, які їх характеризують. Результати методології відображаються як у кількісній, так і якісній формі, що не суперечить основним положенням теорії складних систем.

Ключові слова: методологія синтезу та аналізу, диференціально-ігрові моделі та методи моделювання, інформаційний ресурс, процес кібернападу, рівень захищеності, кібератака, стратегія кіберзахисту, стратегія кібернападу.

Постановка проблеми. Стрімкий розвиток науково-технічного прогресу на початку XXI сторіччя в галузі інформаційних технологій (ІТ- технологій) пов'язаний з повсюдним впровадженням їх у всі сфери діяльності сучасного суспільства будь-якої розвиненої держави світу. Високі темпи інформатизації українського суспільства та державних інститутів сприяють подальшому зростанню ролі й місця кіберпростору в питаннях забезпечення національної безпеки в інформаційній сфері. Кіберпростір на сьогодні виступає системоутворюючим чинником, безпека якого не в останню чергу визначає рівень інформаційної безпеки (ІБ) держави. Масова доступність ІТ- технологій відкриває широкі можливості щодо здійснення несанкціонованого доступу (НСД) до державних інформаційних ресурсів (ІР) як неавторизованим користувачам, так і злочинним угрупованням, чим створює передумови для виникнення загроз безпеці інформації у національному сегменті кіберпростору в інформаційній сфері [1]. Протидія таким загрозам є принциповим аспектом укріплення стратегічної стабільності держави та її ІБ [2]. Безпрецедентний у світовій практиці за своїми аналогами й наслідками для органів державної влади інцидент з ІБ, пов'язаний з масованим кібернападом (КБн) на державні ІР, що відбувся в лютому 2012 року в національному сегменті кіберпростору, спонукає до