КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (10), 2020
ISSN 2663 - 4023
DOI: 10.28925/2663-4023.2020.10.8897
УДК 654.071

Опірський Іван Романович
Д.т.н., доц., професор кафедри захисту інформації
Національний Університет "Львівська Політехніка", Львів, Україна
ORCID: 0000-0002-8461-8996
Ivan.r.opirskyi@lpnu.ua

Василишин Святослав Ігорович
Аспірант кафедри захисту інформації
Національний Університет "Львівська Політехніка", Львів, Україна
ORCID: 0000-0003-1944-2979
Sviatoslav.I.Vasylyshyn@lpnu.ua

Піскозуб Андріян Збігнєвич
К.т.н.,доц., доцент кафедри захисту інформації
Національний Університет "Львівська Політехніка", Львів, Україна
ORCID: 0000-0002-3582-2835
Azpiskozub@gmail.com

АНАЛІЗ ВИКОРИСТАННЯ ПРОГРАМНИХ ПРИМАНОК ЯК ЗАСОБУ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Анотація. У цій статті проаналізовано використання програмних приманок як активу захисту інформації. Проведено ретельне дослідження типів приманок, їх переваг та недоліків, можливих порушень безпеки, конфігурації та загальної ефективності системи. Часто на карту поставлений весь електронний бізнес організації, і навіть при найнадійнішій системі захисту стовідсоткова гарантія невразливості внутрішніх даних компанії не буде надана в принципі. Залежно від цілей, які переслідує програмна приманка, вона може мати різні параметри конфігурації, починаючи від програмних рівнів, які не потребують великих налаштувань, і закінчуючи складними апаратними комплексами. Залежно від рівня складності приманки та її можливостей їх можна класифікувати на три групи: слабкі, середні та сильні рівні взаємодії. Окрім суто практичного застосування Honeypot, описаного вище, не менш важливою є й інша сторона питання - дослідження. На жаль, однією з найбільш актуальних проблем професіоналів безпеки є брак інформації. Хто загрожує, чому вони атакують, як і якими засобами вони користуються - ці питання дуже часто не мають чіткої відповіді. Поінформовані засоби озброєні, але у світі безпеки такої інформації не вистачає - джерел даних немає. Це дуже рідкісний сценарій, оскільки ніхто не може навіть теоретично допустити можливість використання пастки як стартової точки для нападу на інші об'єкти. Якщо дозволено використовувати Honeypot для підключення до віддалених хостів, зловмисник зможе атакувати інші системи, використовуючи IP-адресу пастки як джерело атаки, що з юридичної точки зору спричинить серйозні проблеми. Така можливість може бути заборонена або контрольована, але якщо вона заборонена, вона може здатися зловмисникові підозрілою, а якщо вона існує, але контролюється, зловмисник може оцінити обмеження або заборонені запити на основі інформації отримавши, зробіть висновок, що об'єкт, що атакується, є пасткою.

Ключові слова: програмні приманки; безпека; аналіз рівнів; інформація; приманки.

1. ВСТУП

Honeypot - це мережевий ресурс, метою якого є «злом» зловмисника, його ідентифікація та відстеження дій, які він здійснює для злому сайту [9]. Це одна з

© І.Р. Опірський, С.І. Василишин, А.З. Піскозуб, 2020

технологій, яка призначена для забезпечення мережевої безпеки. Інструменти Honeypot відрізняються від класичних засобів безпеки тим, що вони не призначені для виконання якихось конкретних завдань. Навпаки, honeypot - це гнучкий інструмент, який можна використовувати в різних ситуаціях. Назва походить від англійської ідеї, згідно з якою, якщо залишити горщик з медом (вразливий вузол), на ньому обов'язково будуть літати бджоли (хакери). Протистояти приманкам досить складно, але цілком реально. Мета цієї роботи - дати огляд структури програмних приманок та деяких способів протистояння з ними.

Постановка проблеми.

Пастки можна використовувати для виявлення несанкціонованої діяльності, де традиційні рішення безпеки здатні генерувати величезну кількість записів журналу, тоді як лише деякі з них відображають реальні спроби проникнення або розслідування. Крім того, не всі сучасні інформаційні технології мають інтелектуальні здібності і не завжди можуть визначити незнайомі досі напади. Приманка Honeypot успішно вирішує такі проблеми, оскільки завдяки невеликій кількості сформованої корисної інформації ви можете бути впевнені, що є атака чи дослідження. Пастки також використовуються для реагування на спробу зловмисника вторгнутися в мережеву інфраструктуру. Якщо зловмисник потрапляє в мережу, і одна з атакованих систем виявляється пасткою, корисна інформація, отримана з цієї пастки, використовується для реагування на атаку. Описані вище переваги можуть викликати у користувача ілюзію, що Honeypot - ідеальний інструмент для максимальної безпеки.

Ідея Honeypot представлена в більш широкому розумінні - на рівні всієї мережі - Honeynet. Однак така система складається не з одного комп'ютера або активного мережевого пристрою, а з цілої мережі. Він знаходиться за брандмауером і перехоплює всі вхідні та вихідні з'єднання, потім інформація про активність зловмисника вивчається та аналізується. Honeynet, звичайно, створює для зловмисника більш точну картину, ніж окремий Honeypot. Крім того, за допомогою декількох машин з різним програмним забезпеченням можна дізнатись набагато більше про дії хакера, ніж за допомогою однієї пастки.

Аналіз останніх досліджень і публікацій.

Проблеми захисту систем з використанням програмних приманок досліджуються багатьма науковцями та спеціалістами цієї галузі, однак в сучасному світі на надто багато компаній готова надавати інформацію з приводу того чи використовують вони дану систему захисту в своїх компаніях. Останнє дослідження по використанню програмних приманок серед світових компаній було опубліковано у статтях Andrea Dominguez, "The State of Honeypots: Understanding the Use of Honey Technologies Today", SANS Reading Room, 2020 [3] та Khan, Z.A.; Abbasi, U. "Reputation Management Using Honeypots for Intrusion Detection in the Internet of Things" [4].

Технологія Honeypot забезпечує аналітикам такі переваги: збір інформації про хакерів, вимоги до системних ресурсів, простота управління та чіткість використання. Зазвичай системи IDS (виявлення вторгнень) реєструють десятки або навіть сотні мегабайт інформації на день. Знайти потрібну інформацію в цій горі журналів непросто, оскільки більшість запитів є законними запитами користувачів. Honeypot реєструє незрівнянно менші обсяги інформації, які на 100% містять інформацію, необхідну для аналізу (якщо вона правильно налаштована). З цього випливає, що honeypot не є ресурсоємним завдяки своєму призначенню; йому не потрібні оновлення або постійна технічна підтримка; досить налаштувати і очікувати. Крім того, приманки є яскравим прикладом того, що компанії не даремно вклали гроші. Якщо компанія

ISSN 2663 - 4023

інвестує в IDS та подібну безпеку, а потім ці кошти надійно захищають мережу, може здатися, що це була даремна трата грошей, оскільки ніхто не зламає мережу. У такому разі Honeypot буде гарним доказом того, що мережа все ще ламається, а гроші не даремні. Причому, в умовах вразливості Української держави, а саме інформаційної війні – питання побудови систем захисту інформаційних мереж є актуальним [5].

Мета статті.

Метою статті є аналіз використання програмних приманок як засобу забезпечення інформаційної безпеки, визначення її переваг та недоліків, постановка даного засобу захисту інформації в ієрархії інструментів захисту та визначення її привабливості й ефективності.

2. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Залежно від цілей, які переслідує програмна приманка, вона може мати різні параметри конфігурації, починаючи від програмних рівнів, які не потребують великих налаштувань, і закінчуючи складними апаратними комплексами. Залежно від рівня складності приманки та її можливостей їх можна класифікувати на три групи: слабкі, середні та сильні рівні взаємодії. Давайте розберемося з кожним з них по порядку [9].

Honeypot з низьким рівнем простоти у використанні та дуже надійні. Вони імітують лише частину служб, і зловмисник буде обмежений у взаємодії з ними. Наприклад, вони можуть імітувати систему UNIX, на якій запущено telnet. При підключенні до такого сайту зловмисник отримає запит на вхід і спробує забрати паролі до системи. Або FTP-сервер з анонімним обліковим записом і нібито файлом із паролями (номери кредитних карток тощо). Будь-яка спроба отримати доступ до цього файлу буде спробою злому. Система буде вести журнали часу спроби злому, IP-адресу та порт зловмисника, а також порт, до якого він намагався отримати доступ. Завданням таких програмних приманок є мінімальний рівень реєстрації [1]. Ризик використання приманок рівня 1 мінімальний, але він є. Це пов’язано з тим, що саме програмне забезпечення теж є програмою; отже, воно може бути вразливим. Якщо його можна обійти, зловмисник отримає доступ до решти вузлів мережі. Сила цих найпростіших приманок у тому, що вони самі по собі прості. Відомо, що чим простіші, тим надійніші, тому ці програми мінімізують ризик, пов’язаний з можливою поломкою самої приманки і наступним поломкою системи.

Приманки середнього рівня надають більше можливостей для реконструкції зловмисника, більш складного і, отже, більш вразливого. Наприклад, така система може моделювати більш складні веб-сервери, які можуть реагувати на нестандартні команди та мати більш досконалу систему реєстрації. Таким чином, для розширення середовища зловмисника (тобто він зможе взаємодіяти не тільки з «підробленими» службами, але і з «підробленою» ОС) це дасть більше можливостей для реєстрації. Але такий підхід також створить більше проблем. По-перше, це рішення є досить складним, тому помилки можуть виникати на етапі роботи або конфігурації, що в подальшому призведе до більшої вразливості системи. По-друге, надати віртуальному середовищу функціональність реальної системи - це трудомістке завдання. Чим більше функціональності та реалістичності забезпечує віртуальне середовище, тим простіше зловмиснику обійти це середовище та отримати контроль над реальною операційною системою. Крім того, шкідливий код може вийти з-під контролю віртуальної машини, якщо він спеціально для нього розроблений.

Приманки з високим рівнем взаємодії дають максимум інформації про зловмисника і є максимально складними та небезпечними. Вони дають зловмиснику доступ до реальної системи, яка нічого не робить і не підключена до інших систем. Структура такої приманки найчастіше така: вузол приманки, мережевий датчик та сховище інформації. Такий вузол може бути розташований у мережі за брандмауером, і тоді фактичний контроль лежить на брандмауері. Якщо вузол приманки неправильно налаштований або трапляються якісь інші непередбачені ситуації, зловмисник зможе отримати доступ до мережі. Одним з недоліків такого рішення може бути складність його реалізації та відносна вартість підтримки [2].

Розглянемо недоліки (не враховуючи тих, які вже були названі) різних способів реалізації приманки. Сама ідея приманки передбачає наявність ресурсу, який приверне увагу потенційних зловмисників. До нього не повинно надходити юридичних звернень, а будь-яке звернення до нього має бути підозрілим до адміністратора [3]. Однак ізоляція вузла від інших мережевих вузлів вже має бути підозрілою з гострим бажанням швидше залишити ресурс. Якщо у "приманки" немає іншого трафіку, крім злому то вам слід піти. Але навіть якщо вузол взаємодіє з деякими іншими, це може бути просто віртуальна локальна мережа, яка створює «видимість» взаємодій, або інші вузли «приманки», які налаштовані на різну ступінь «складності злому»: від найпростіших, що містять дистрибутив за замовчуванням, до найбільш безпечних, які спрямовані на виявлення невідомих атак. Якщо в мережі видно чітко вразливий вузол (і особливо з конфігурацією за замовчуванням; дірки таких конфігурацій добре відомі), це або відсутність знань адміністратора, або пастка. Наслідки другого можуть бути набагато скорботнішими, ніж переваги першого. Honeypot рівня 1 можна розпізнати, надіславши йому нестандартну команду, або якщо це популярна система, що містить помилки. У такому випадку він сам може бути об’єктом нападу. Як уже зазначалося, шкідливий код може вийти з-під контролю віртуальної машини (vmware не є досконалим, а також містить помилки), тоді зловмисник отримає контроль над батьківською системою (стосовно приманок рівня 2). Якщо зловмисник має справу з приманкою на трьох рівнях взаємодії, можуть бути різні варіанти. Наприклад, неправильно налаштований мережевий датчик (який зберігає базу даних «серцем» всієї системи) може або подавати помилкові попередження після кожного сканування порту, або не реагувати на дещо змінені атаки [11].

Загалом, розбиваючи дуже серйозні ресурси, такі як державні сайти, банки тощо, зловмисники можуть використовувати різні стратегії. IP-адреса зловмисника ще не є показником. Це може бути IP-адреса проксі-сервера. У цьому випадку залишається лише покластися на проксі-сервер для ведення журналу. Тому деякі зловмисники використовують мережу з декількох зламаних комп’ютерів і отримують доступ до мережі через модем GPRS / EDGE у мобільному телефоні (який при необхідності може бути швидко знищений), віддаляючись від місця свого проживання. Крім того, якщо машина зловмисника є вразливою, приманка може контратакувати й занести вірус на чужому комп’ютері або зібрати інформацію, таку як файли cookie [5].

ISSN 2663 - 4023

Таблиця 1

Порівняльна таблиця різних приманок та їх властивостей

Специфікації	Illusive Networks платформа	Trapx платформа	Xello платформа
Можливість створювати оманливу ОС		Windows Linux	Windows Linux
C&C Detection	Ні	Так	Ні
MITM Detection	Ні	Так	Ні
Земульовані пастки	Так	Так	Так
Приманки	Так	Так	Ні
NAC інтеграція	Так	Так	Ні
Повні OS пастки	Так	Так	Так
SIEM інтеграція	Так	Так	Так
Інтеграція кінцевої точки	Так	Так	Так
EDR	Так	Так	Так
Активна директорія	Так	Так	Так
Лінійна кореляція	Так	Так	Так
Інтеграція сендбоксу	Ні	Так	Ні
Бази даних	Ні	Так	Так
POS	Ні	Так	Ні
ATM	Ні	Так	Ні
SCADA	Так	Так	Так
Iot	Так	Так	Так
Хмари	Невідомо	AWS Azure Openstack	Ні
Використання зображень	Так	Так	Так
Відкрита API для додаткових маніпуляцій	Так	Так	Так
Виявлення Ботнету	Ні	Так	Так
Автоматичний аналіз коду	Ні	Так	Ні
Конструктор пасток	Ні	Так	Так

Вибираючи жертву, зловмиснику слід якомога більше вивчити топологію її мережі. Він повинен переконатися, що вузол обслуговує зовнішній трафік, що конфігурація відрізняється від конфігурації за замовчуванням, що вузол використовується іншими членами мережі і т. Д. Потім він може ескалювати ситуацію протягом декількох днів, скануючи порти та надсилаючи рядки що імітують переповнення буфера. Він також може атакувати саму медовуху, щоб вимкнути її (ddos, SYN, ECHO-death та інші атаки, які приховують IP-адресу зловмисника).

Насправді Honeypot - це наживка, на яку у випадку удачі та фактору високої надійності потрапляє зловмисник. Завданням Honeypot є пройти атаку або несанкціоноване розслідування, що дозволить вивчити стратегію зловмисника та визначити діапазон засобів, за допомогою яких можуть бути здійснені атаки на реальні об'єкти безпеки [7]. Реалізація Honeypot не є принциповою, і це може бути як спеціальний виділений сервер, так і одна мережева служба, завдання якої - привернути увагу хакерів. Розроблена організація мережі зображена на Рисунку 1.

ISSN 2663 - 4023

Honeypot принципово відрізняється від усіх подій у галузі безпеки. Як правило, усі товари на цьому ринку розроблені для вирішення суворо визначеної функції (неважливо, чи йдеться про апаратне чи програмне забезпечення): брандмауер вирішує завдання обмеження доступу з однієї мережі в іншу на різних рівнях, SSH Послуга призначена для зашифрованого доступу до ресурсів операційної системи тощо. Технологія Honeypot не призначена для вирішення конкретної проблеми, а представляє цілу філософію - гнучка, настроювана відповідно до мети. Honeypot надає фахівцям із безпеки значні переваги. Перш за все, це збір необхідної інформації, часто містить цінну інформацію. Розгортання та експлуатація живих приманок не представляє особливих труднощів, а інструменти Honeypot, як правило, не вимагають системних ресурсів [5].

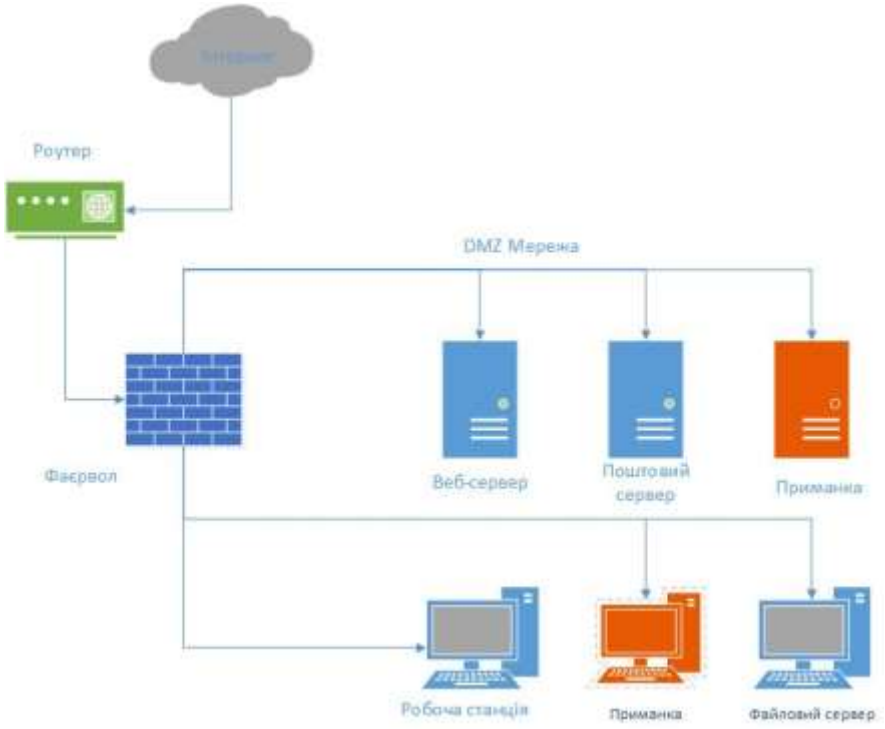

Рис. 1. Схема організованої робочої мережі з використанням приманки

За бажанням адміністратор мережі може відстежувати події в хронологічному порядку та з'ясувати, що сталося на певному ділянці інфраструктури за X годин Y хвилин Z секунд. Однак не можна заперечувати, що виділення необхідної інформації часто є досить складним, оскільки доводиться переглядати величезні файли журналів, щоб з'ясувати, коли, де і як було виявлено підозрілу несанкціоновану діяльність. З цієї точки зору інструменти Honeypot виглядають майже ідеально: зібраної інформації небагато, але вся вона має велику цінність, оскільки така інформація розкриває суть спроби зламати, сканувати чи дослідити. Оскільки Honeypot спочатку «кинули» для нападу та досліджень, можна припустити, що майже вся інформація, взята з пастки, відображає дії зловмисників. На його основі ви можете аналізувати, будувати статистичні дані щодо методів, які використовують хакери, а також визначати наявність будь-яких нових рішень, що використовуються хакерами.

Щоб зрозуміти цінність пасток, розглянемо модель безпеки Брюса Шнайера, яка має три рівні: запобігання, виявлення та реагування. Пастки для приманок можна

активувати на всіх трьох рівнях; наприклад, на рівні профілактики Honeypot використовується для уповільнення або повного припинення автоматичних вторгнень. Honeypot вирішує проблеми такого роду - завдяки невеликій кількості корисної інформації, що генерується, ви можете бути майже впевнені, що має місце напад чи розслідування.

Особливу увагу слід приділити встановленню та експлуатації Honeypot. Як правило, весь спектр заходів зводиться до "встановлення та очікування". Найпоширеніший випадок - із виділеним сервером під контролем фахівців. Сьогодні існує багато підроблених програм, які створюють враження справжніх їх головне завдання - записати весь обмін. Перевага Honeypot полягає в тому, що копію програмного забезпечення можна зробити на морально застарілому сервері, який не може впоратися з типовими обчислювальними завданнями електронного бізнесу.

Пастки також використовуються для реагування на вторгнення. Якщо зловмисник потрапив у мережу, і одна з атакованих систем виявилася пасткою, корисна інформація, отримана з цієї пастки, використовується для реагування на атаку. Описані переваги можуть створити ілюзію, що Honeypot - ідеальний інструмент для максимальної безпеки. На жаль, через низку недоліків це не зовсім так, і Honeypot може служити доповненням до існуючого асортименту захисних засобів. Перш за все, слід відзначити вузьку спрямованість конкретної пастки. Також існує ймовірність виявлення та небезпека повного злому Honeypot [8].

Honeypot потенційно не може покрити всі проблеми безпеки, тому доведеться або дослідити рівень безпеки окремої частини інфраструктури, або використовувати кілька приманок. Неможливо виключити ризик того, що зловмисник зрозуміє, що перед ними не справжній "фронт роботи", а лише фіктивна пастка. Найчастіше це трапляється через неправильну або недостатньо ретельну постановку пастки, тобто у переважній більшості випадків винен людський фактор.

Окрім суто практичного застосування Honeypot, описаного вище, не менш важливою є й інша сторона питання - дослідження. На жаль, однією з найбільш актуальних проблем професіоналів безпеки є брак інформації. Хто загрожує, чому вони атакують, як і якими засобами вони користуються - ці питання дуже часто не мають чіткої відповіді. Поінформовані засоби озброєні, але у світі безпеки такої інформації не вистачає - джерел даних немає. Як правило, фахівці з безпеки дізнавались про зловмисників, вивчаючи використані ними засоби та аналізуючи ознаки нападу. Коли система була скомпрометована, адміністратори часто знаходили засоби зловмисників, залишені ними на зламаній системі. Таким чином, багато припущень було зроблено на основі фіксованих інструментів та технік.

3. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Отже, пастки виготовляються людиною, а тому вони також вразливі. До того ж вони є лише частиною "оборонної системи". Покладати всю свою надію на приманки безглуздо. Необхідно використовувати комплексний підхід, при якому програмна приманка має свою нішу. Вони повинні запускати добре налаштований брандмауер і бути ізольованими від мережі. Користувачі повинні вести бесіду про те, які паролі вибрати та якими вони повинні бути. Системний адміністратор повинен вдосконалювати свої навички, регулярно оновлювати систему та контролювати її стан. І керівництво не повинно економити на безпеці даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Milov, O., Voitko, A., Husarova, I., Opirskyy, I., Fraze-Frazenko, O., et.al., "Development of methodology for modeling the interaction of antagonistic agents in cybersecurity systems" Eastern-European Journal of Enterprise Technologies, 2019. DOI: 10.15587/1729-4061.2019.164730

[2] Дудикевич В. Б. Забезпечення інформаційної безпеки держави: навчальний посібник / В. Б. Дудикевич, І. Р. Опірський, П. І. Гаранюк, В. С. Зачепило, А. І. Партика. – Львів: Видавництво Національного університету «Львівська політехніка», 2017. – 204 с. (IBSN 978-966-941-091-7).

[3] Andrea Dominguez, "The State of Honeypots: Understanding the Use of Honey Technologies Today", SANS Reading Room, 2020.

[4] Khan, Z.A.; Abbasi, U. "Reputation Management Using Honeypots for Intrusion Detection in the Internet of Things". Electronics 2020, 9, 415.

[5] Z. Brzhevska, N. Dovzhenko, R. Kyrychok, G. Gaidur, і A. Anosov, «Інформаційні війни: проблеми, загрози та протидія», *Кібербезпека: освіта, наука, техніка*, вип. 3, вип. 3, с. 88-96, бер 2019.

[6] Akiyama, M., Yagi, T., Hariu, T., & Kadobayashi, Y. (2017). Honeycirculator: distributing credential honeytoken for introspection of web-based attack cycle. International Journal of Information Security. DOI:10.1007/s10207-017-0361-5

[7] Gandotra, V., Singhal, A., & Bedi, P. (2012). Threat-Oriented Security Framework: A Proactive Approach in Threat Management. Procedia Technology, 4, 487-494. DOI:10.1016/j.protcy.2012.05.078

[8] Onaolapo, J., Mariconti, E., & Stringhini, G. (2016). What Happens After You Are Pwnd: Understanding The Use Of Leaked Account Credentials In The Wild. Proceedings of the 2016 ACM on Internet Measurement Conference - IMC 16. DOI:10.1145/2987443.2987475

[9] Martin, W.W. "Honeypots and Honeynets–Security through Deception." [Електронний ресурс] Available: http://www.sans.org/reading_room/whitepapers/attacking/41.php, SANS Institute, 2001, As Part of the Information Security Reading Room.

[10] Wikipedia. Honeypot (computing) [Електронний ресурс] Available: http://en.wikipedia.org/wiki/Honeypot_(computing).

[11] Norton. [Електронний ресурс] Available: https://us.norton.com/internetsecurity-iot-what-is-a-honeypot.html

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (10), 2020
ISSN 2663 - 4023

Ivan R. Opirskyy
Dc.S., Associate Professor, Professor of Information Security Department
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0000-0002-8461-8996
Ivan.r.opirskyi@lpnu.ua

Sviatoslav I. Vasylyshyn
Postgraduate student of Information Security Department
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0000-0003-1944-2979
Sviatoslav.I.Vasylyshyn@lpnu.ua

Andrian Z. Piskozub
Ph.D, Associate Professor, Associate Professor of Information Security Department
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0000-0002-3582-2835
Azpiskozub@gmail.com

ANALYSIS OF THE USE OF SOFTWARE BAITS (HONEYPOTS) AS A MEANS OF ENSURING INFORMATION SECURITY

Abstract. This article analyses the usage of software baits as an information security asset. They provided close research about honeypot types, their advantages and disadvantages, possible security breaches, configuration and overall system effectiveness. Often, the entire electronic business of the organization is at stake, and even with the most reliable system of protection, a one-hundred-per cent guarantee of invulnerability of internal company data will not be given in principle. Depending on the goals pursued by the software lure, it can have various configuration parameters, ranging from software levels that do not require large settings and ending with complex hardware complexes. Depending on the level of complexity of the bait and its capabilities, they can be classified into three groups: weak, medium, and strong levels of interaction. In addition to the purely practical application of Honeypot, described above, no less important is the other side of the issue - research. Unfortunately, one of the most pressing problems for security professionals is the lack of information. Who threatens, why they attack, how and by what means they use - these questions very often do not have a clear answer. Informed means are armed, but in the world of security such information is not enough - there are no data sources. This is a very rare scenario, as no one can even theoretically allow the possibility of using a trap as a starting point to attack other objects. If you allow Honeypot to connect to remote hosts, an attacker could attack other systems using the trap's IP address as the source of the attack, which would cause serious legal issues. This possibility may be prohibited or controlled, but if it is prohibited, it may seem suspicious to the attacker, and if it exists but is controlled, the attacker may assess the restrictions or prohibited requests based on the information received, conclude that the attacked object is a trap.

Keywords: honeypots; security; levels; analysis; information; baits.

REFERENCES

[1] Milov, O., Voitko, A., Husarova, I., Opirskyy, I., Fraze-Frazenko, O., et.al., “Development of methodology for modeling the interaction of antagonistic agents in cybersecurity systems” Eastern-European Journal of Enterprise Technologies, 2019. DOI: 10.15587/1729-4061.2019.164730

[2] Dudykevych V.B. Provision of information security of the state: a textbook / V.B. Dudykevych, I.R. Opirskyy, P.I. Garanyuk, V.S. Zachepilo, A.I. Partyka. - Lviv: Publisher of Lviv Polytechnic National University, 2017. - 204 p. (IBSN 978-966-941-091-7).

[3] Andrea Dominguez, “The State of Honeypots: Understanding the Use of Honey Technologies Today”, SANS Reading Room, 2020.

[4] Khan, Z.A.; Abbasi, U. “Reputation Management Using Honeypots for Intrusion Detection in the Internet of Things”. Electronics 2020, 9, 415.
[5] Z. Brzhevska, N. Dovzhenko, R. Kyrychok, G. Gaidur, and A. Anosov, "Information Wars: Problems, Threats and Counteraction", Cybersecurity: Education, Science, Technology, Vol. 3, issue 3, p. 88-96, Mar 2019.
[6] Akiyama, M., Yagi, T., Hariu, T., & Kadobayashi, Y. (2017). Honeycirculator: distributing credential honeytoken for introspection of web-based attack cycle. International Journal of Information Security. DOI:10.1007/s10207-017-0361-5
[7] Gandotra, V., Singhal, A., & Bedi, P. (2012). Threat-Oriented Security Framework: A Proactive Approach in Threat Management. Procedia Technology, 4, 487-494. DOI:10.1016/j.protcy.2012.05.078
[8] Onaolapo, J., Mariconti, E., & Stringhini, G. (2016). What Happens After You Are Pwnd: Understanding The Use Of Leaked Account Credentials In The Wild. Proceedings of the 2016 ACM on Internet Measurement Conference - IMC 16. DOI:10.1145/2987443.2987475
[9] Martin, W.W. “Honeypots and Honeynets – Security through Deception.” [Online] Available: http://www.sans.org/reading_room/whitepapers/attacking/41.php, SANS Institute, 2001, As Part of the Information Security Reading Room.
[10] Wikipedia. [Online] Available: http://en.wikipedia.org/wiki/Honeypot_(computing).
[11] Norton. [Online] Available: https://us.norton.com/internetsecurity-iot-what-is-a-honeypot.html

(cc) BY-NC-SA
This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.