МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

БАБИЧ Сергій Васильович

ТЕХНОЛОГІЇ ТА АЛГОРИТМИ ОБМАНУ ДЛЯ БОРОТЬБИ З КІБЕРАТАКАМИ

спеціальність: 125 – Кібербезпека

освітньо-професійна програма –Кібербезпека

Кваліфікаційна робота

Виконав студент групи КБм -21
С.В. Бабич

Науковий керівник
д.т.н., професор В.В.Яцків

Кваліфікаційну роботу допущено
до захисту:

«____» _____ ___2022 р.

Завідувач кафедри

_____ В.В.Яцків

ТЕРНОПІЛЬ - 2022

Факультет комп'ютерних інформаційних технологій

Кафедра кібербезпеки

Освітній ступінь «магістр»

спеціальність: 125 - Кібербезпека

освітньо-професійна програма –Кібербезпека

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ В.В.Яцків

«____» _____ 2021 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

БАБИЧ Сергій Васильович

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:

Технології та алгоритми обману для боротьби з кібератаками /
Deception technologies and algorithms to combat cyber attacks

керівник роботи д.т.н., професор В.В. Яцків

затверджені наказом по університету від 31 грудня 2021 року № 606

2. Строк подання студентом закінченої випускної кваліфікаційної роботи 16 грудня 2022 року.

3. Вихідні дані до кваліфікаційної роботи: завдання на випускну кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

– провести аналіз методів обману для протидії кібератак;
– дати визначення обману у інформаційному середовищі, моделі обману;
– розглянути технологію обману;
– провести аналіз існуючих програмних продуктів;
– продемонструвати практичну реалізацію сервісу технології обману;
– дослідити ефективність розглянутих рішень.

5. Перелік графічного матеріалу у роботі:

Огляд відповідних методів безпеки та примітивів на різних рівнях, сірий: технології на основі обману

Схема організації зв'язків з використанням Honeypot.

Схема організації зв'язків з використанням Deception.

Схематичне зображення агента.

Інцидент кібератаки з використанням пастки

Схематичне зображення фіктивної бази користувачів.

Емуляція пастки для слухача трафіку.

Моніторинг версії ПЗ на робочих станціях.

Атака PasswordSpray SMB.

6. Консультанти розділів кваліфікаційної роботи

	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 11 жовтня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	АНАЛІЗ МЕТОДІВ ОБМАНУ ДЛЯ ПРОТИДІЇ КІБЕРАТАК	12.2021 р. – 03.2022 р.	
2	ТЕХНОЛОГІЇ ОБМАНУ. ЕВОЛЮЦІЯ ІС ТА ВЕКТОРИ РОЗВИТКУ	03.2022 р. – 05.2022 р.	
3	ІМПЛЕМЕНТАЦІЯ ТЕХНОЛОГІЇ ОБМАНУ DECEPTION	05.2022 р. – 11.2022 р.	

Студент _____ Бабич С.В.
(підпис)

Керівник роботи _____ д.т.н., професор В.В.Яцків
(підпис)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Технології та алгоритми обману для боротьби з кібератаками» на здобуття освітнього ступеня «Магістр» зі спеціальності 125 «Кібербезпека» освітньо-професійної програми «Кібербезпека» написана обсягом 82 сторінку і містить 23 ілюстрації, 6 таблиць та 175 джерел за переліком посилань.

Метою кваліфікаційної роботи є збір аналітичної інформації щодо використання технології обману та отримання навичок її імплементації.

Методи досліджень. Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: методи аналізу інформації та даних, методи автентифікації користувачів, методи контролю доступу, методи проектування.

Результати дослідження: проведено аналіз досліджень та технології обману зокрема, виокремлено актуальні форми до імплементації, отримано навички розгортання системи.

Результати роботи можуть успішно застосовуватися при реалізації систем протидії кібератакам.

Ключові слова: інформаційна безпека, мережева безпека, обман, технологія обману, кіберобман, honeypot, honeytoken, приманки, пастки, deception, виявлення атак, кібератака, цільові атаки, направлені атаки.

ABSTRACT

The qualification work on the topic "Technologies and algorithms of deception to combat cyberattacks" for obtaining the Master's degree in the specialty 125 "Cybersecurity" of the educational and professional program "Cybersecurity" is written in the volume of 82 pages and contains 23 illustrations, 6 tables and 175 source’s by reference list.

The purpose of the qualification work is to collect analytical information on the use of deception technology and to acquire skills in its implementation.

Research methods. To solve the tasks in this qualification work, the following methods of information and data analysis, user authentication methods, access control methods, and design methods were used.

Research results: an analysis of research and deception technology in particular was carried out, relevant forms for implementation were identified, skills of system deployment were obtained.

The results of the work can be successfully applied in the implementation of systems for countering cyberattacks.

Key words: information security, network security, deception, deception technology, cyber fraud, honeytokens, lures, traps, deception, honeypot, detection of attacks, cyberattack, target attack, focus attack.

ЗМІСТ

ВСТУП

Сунь Цзи одного разу написав, що «всі війни базуються на обмані» [1]. Це було задовго до перших цифрових пристроїв. З тих пір, 2500 років тому, обман був невід'ємним аспектом багатьох сфер діяльності, наприклад, військових. Це достатньо актуально, з точки зору на трагічну війну в Україні (що слугувало причиною формуванню додаткового розділу, який присвячено військовому аспекту технології обману та тренування протидії кібератакам).

Протягом багатьох років обман був важливим аспектом військових операцій. В інформаційній безпеці (ІБ) соціальна інженерія, як детально описано K.D. Mitnick [2] було першим використанням обману. У 1986 і 1991 роках C.Stoll [3], відповідно B.Cheswick [4], переніс концепцію обману на захисні програми. Ці програми отримали назву honeypots (HP). Пізніше цю концепцію було узагальнено до технології обману deception (DT), яка є надмножиною HP та всіх інших технологій, що спираються на моделювання та дисимуляцію. Deception Toolkit, опублікований F.Cohen, був першим публічним програмним забезпеченням для обману [5].

За останні три десятиліття концепції обману набули зростаючої популярності в інформаційній безпеці. Заходи безпеки на основі периметра, такі як брандмауери та механізми автентифікації, не забезпечують належного рівня безпеки в контексті внутрішніх загроз і соціальної інженерії. Стратегії поглибленого захисту, такі як виявлення та запобігання вторгненням на основі сигнатур, часто страждають від великої кількості хибнопозитивних виявлень, що призводить до втоми тривоги безпеки інформації та систем управління подіями. DT мають кілька основних переваг, таких як низький рівень хибно-позитивних сповіщень і можливості виявлення за 0 днів, що робить його перспективним рішенням для боротьби з розширеними загрозами безпеці, такими як виявлення вторгнень [6] і атрибуція [7, 8].

Мета і завдання дослідження. Метою роботи є дослідження технологій та алгоритмів обману для боротьби з кібератаками.

Досягнення визначеної мети передбачає вирішення таких завдань:

- дослідити методи реалізації обману як контрміри кібератакам;
- описати типову модель обману зі структуризацією пасток;
- провести аналіз відомих технологій обману;
- проаналізувати сценарій ефективного використання методів обману для боротьби із кібератаками;
- дати оцінку ефективності їх реалізації та навести статистичні дані щодо наслідків імплементації.

Об'єкт дослідження – процеси функціонування систем обману та протидії кібератакам та кіберзлочинам.

Предмет дослідження – технології та алгоритми обману (HP, DT); інформаційні системи, сервіси та продукти даного напрямку.

Методи досліджень. Для розв'язання поставлених задач у даній кваліфікаційній роботі використано: методи та техніки обману (HP, DT), обмеження доступу, контроль трафіку та версій даних, процеси автентифікації та авторизації.

Наукова новизна одержаних результатів. Проведено аналіз технічної реалізації технологій обману (HP, DT) в галузі боротьби із кібератаками.

Практичне значення отриманих результатів. Запропоновано модель протидії можливим кібератакам.

Публікації та апробація КР.

1. Бабич С.В. Визначення технології обману. Формування моделі Deception. Матеріали наукової конференції «Автоматизація та комп'ютерно-інтегровані технології» (АКІТ - 2022), Тернопіль, 2022. – С.

2. Бабич С.В. Технологія обману на основі файлів. Матеріали наукової конференції «Кібербезпека та комп'ютерно-інтегровані технології» (КБКІТ - 2022), Тернопіль, 2022. – С.

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ – програмне забезпечення

ІБ – інформаційна безпека

MILDEC – «маскування», вид бойового ПЗ

EW – електронна війна

PSYOPS – психологічні операції

CNO – робота комп’ютерної мережі

OPSEC – операційна безпека

ROWE – орієнтоване на результат робоче середовище

HT - honeytoken

HP – honeypot

DT – технологія обману

MTD – техніка обману підробленими об’єктами

HIHP – промислова приманка високої взаємодії

LIHP – приманки з низьким рівнем взаємодії

AD – Active Directory

ОЗ – облікові записи

HTTP, HTTPS, FTP, POP3, SMTP, SSH і Telnet – протоколи передачі даних

1 АНАЛІЗ МЕТОДІВ ОБМАНУ ДЛЯ ПРОТИДІЇ КІБЕРАТАК

1.1 Передумови, теорія та визначення

Після багатьох років досліджень і розробок комп’ютерна безпека залишається завданням, схильним до помилок, а хронічні проблеми комп’ютерної безпеки вимагають нових підходів. Одним із компонентів інструментів і методів досягнення безпеки може бути обман. У повсякденній безпеці обман відіграє відносно помітну роль, наприклад, залишаючи увімкненим світло у вітальні, щоб зломщики подумали, що хтось є вдома. Однак в інформаційних технологіях обман часто не використовується, або він відіграє приховану, а не явну роль. Обман працює принципово інакше, ніж звичайні методи безпеки. Традиційна безпека, як правило, працює безпосередньо на або проти дій хакера, наприклад, щоб виявити їх або запобігти їм. Обман працює шляхом маніпулювання мисленням хакера, щоб змусити його діяти так, як це вигідно захиснику. Будучи принципово відмінним, обман може бути сильним там, де звичайна безпека слабка (і навпаки). Хоча обман не завжди корисний, він може бути важливим і ефективним способом компенсації властивих традиційній безпеці вразливостей, і може бути вигідно поєднати обидва явно.

У комп’ютерній безпеці перевага обману полягає в тому, що він може протиставити сильні сторони захисника проти слабких сторін хакера. Хакери часто покладаються в значній мірі, якщо не виключно, на одне джерело інформації — дані, отримані через мережу. Часто мережевими даними можна маніпулювати на користь захисника. Крім того, коли хакер вперше потрапляє в мережу, він повинен дізнатися про мережу, досліджуючи її. Розслідування включає сканування та вивчення самої мережі та підключених до неї пристроїв. Процес розслідування хакера в поєднанні з цією початковою наївністю може створити неминучий і передбачуваний канал для обману. Як правило, мережевий захисник має фізичний контроль над своєю мережею, і він це добре знає. Захисник може використати процес розслідування хакера,

щоб надати йому неправду і таким чином атакувати його процес прийняття рішень.

Обман є невід'ємною частиною людської природи та досвіду. Існують законні, навіть необхідні причини обманювати інших, як у спорті та іграх. Обман є частою темою історії, літератури, драматургії та маркетингу. Споживачі регулярно вдаються до протидії обману. Однак небагато людей використовують обман так, як це необхідно для безпеки комп'ютера. Як показує література про військовий обман, ефективний обман супротивника – це професійна навичка, яка потребує розуміння процесів, принципів і методів обману. Обман може бути використаний для атаки на процеси прийняття рішень хакерами; таким чином обман забезпечує образливий захід безпеки — те, чого так бракує захисникам комп'ютерної безпеки.

B.Whaley [9] визначає обман як неправильне сприйняття, яке навмисно викликане іншими суб'єктами. У його типології сприйняття обман має три вимоги:

- не бути множинним сприйняттям;
- не бути спровокованим власним бажанням;
- не бути індукованим ненавмисним.

Приблизно через десять років після цієї публікації J.Bell і B.Whaley [10] опублікували таксономію обману. Ця класифікація класифікує обман на дві основні категорії: прикриття та симуляція.

Симуляція складається з трьох класів:

- маскування;
- перепакування;
- засліплення.

Імітація, винахід та обман входять до категорії симуляції. Ця таксономія є найбільш часто використовуваною таксономією для обману в області інформаційної безпеки. У цій роботі використовується таксономія J.Bell і B.Whaley. J.F.Dunnigan і A.A.Nofi [11] класифікував обман на основі шести принципів обману, представлених C.A.Fowler і R.F.Nesbit [12].

Обман повинен зміцнювати очікування ворога, мати реалістичний час і тривалість, бути інтегрованим з операцією, координуватися з приховуванням справжніх намірів, бути пристосованим до потреб обстановки та бути творчим і творчим. Їх таксономія розрізняє дев'ять класів обману:

- приховування;
- маскування;
- неправдива та підсаджена інформація;
- брехня;
- демонстрація;
- хитрощі;
- демонстрації;
- фінти;
- прозорливість.

N.C.Rowe і H.S.Rothstein [13] запропонували таксономію на основі робіт C.Fillmore [14], T.Copeck [15] і J.L.Austin [16] про лінгвістичні справи. Їхня систематика складається з 32 випадків у 7 групах. F.Stech та інші [17] виявили, що обман визначається в літературі різними способами. Вони опублікували наукометричний аналіз поняття обману в кіберпросторі. J.D.Monroe [18] порівняли кілька конкуруючих визначень обману.

1.2 Військові доктринальні концепції

Таксономія структурована в три шари, де дві основні категорії (активний обман і прикриття) містять 13 атомарних особливостей обману. У 2012 році в армії США [19] опублікували документ, який згрупував MILDEC як частину інформаційних операцій і пов'язаних з EW, PSYOPS, CNO, OPSEC. J.Shim і C.Arkin [20] порівняли кілька таксономій щодо обману щодо домену. У сфері інформаційної безпеки була обрана раніше введена таксономія від ROWE на основі лінгвістичних випадків.

Інші області: філософія, психологія, економіка, військова (таксономія J.Bell і B.Whaley) і біологія. M.H.Almeshekah і E.H.Spafford [21] опублікували роботу про планування та інтеграцію обману. Вони запропонували податкову таксономію з акцентом на цілях обману. Пізніше, M.H.Almeshekah [22] розширив попередню роботу, згрупувавши обман та інші механізми безпеки в чотири категорії. Потім він визначив перетини категорій і зіставив механізми зі знаменитим ланцюгом убивств, запропонованим E.M.Hutchins та інші [23] . J.Pawlick та інші [24] створив таксономію, згрупувавши існуючі теоретико-ігрові підходи до обману, як обговорюється далі в цьому розділі. Їх таксономія поділяє обман на:

- збурення;
- захист рухомої цілі (MTD);
- обфускацію (заплутування, знечитування коду);
- змішування;
- HP-технології;
- провокацію зловмисників.

Більш конкретні таксономії, наприклад, для HP, наведені C.Seifert [25] або F.Pouget [26] . Термін « Технологія обману» (наративом якої стає DT) нещодавно став відомим і наразі використовується для безпеки на основі обману, яка перевершує технологію HP 30-річної давності. Його часто пов’язують із цілісними структурами, які включають: адаптивну HT та генерацію приманок, автоматизоване розгортання та моніторинг, а також інтегровані можливості візуалізації та звітності.

1.3 Обман у середовищі безпеки інформаційних технологій

Термін IT-безпека підсумовує широкий спектр технічних та організаційних заходів, спрямованих на захист попередньо визначеного набору активів. Ці активи можна класифікувати відповідно до так званих цілей

безпеки CIAA: **конфіденційність**, **цілісність**, **автентифікація** та **доступність** і описує цілі захисту звичайних IT - систем. DT не можна відобразити на ці цілі безпеки, але зазвичай використовується для більш абстрактних цілей, наприклад виявлення вторгнень або аналіз поведінки зловмисників. Ще одна фундаментальна відмінність між класичною IT-безпекою та DT полягає у використанні Security by Obscurity (SbO). Цей термін описує навмисне використання методів, більш складних, ніж необхідно, або не опублікованих, щоб перешкодити зловмиснику отримати знання про систему. Хоча це не рекомендовано й неодноразово давало негативні наслідки в класичній IT-безпеці, це дійсний метод уповільнення зловмисників або відволікання їх від можливих цілей. DT не обов'язково залежить від неясності, але є значно ефективнішим, якщо наявність невідома. Відношення класичної IT-безпеки та DT показано на рисунку 1.1.

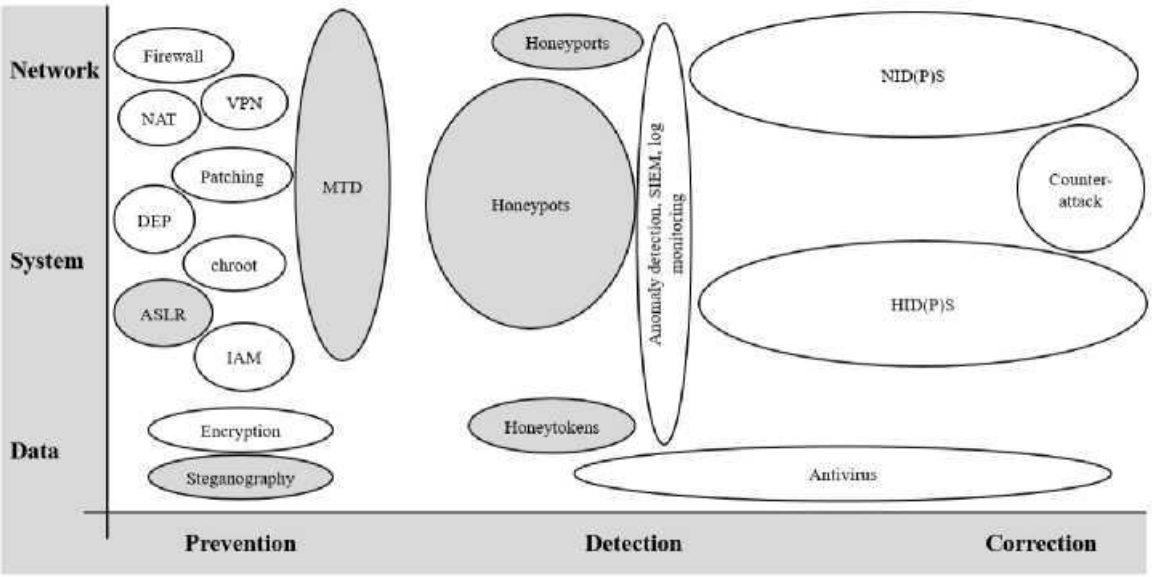

Рисунок 1.1 - Огляд відповідних методів безпеки та примітивів на різних рівнях, сірий: технології на основі обману

На цьому рисунку показано три різні рівні ресурсів для захисту, а також три різні механізми захисту. Ресурси можна згрупувати в мережеві, системні та дані. Захист можна здійснити шляхом запобігання атаку, виявляючи її та реагуючи на неї таким чином, що перешкоджає її результату, так звана

корекція. Існує багато класичних будівельних блоків IT-безпеки, які в основному стосуються запобігання, наприклад віртуальні приватні мережі (VPN), брандмауери, запобігання виконанню даних (DEP), керування ідентифікацією та доступом (IAM) і шифрування. Кілька добре налагоджених механізмів, таких як рандомізація макета адресного простору ((K)ASLR) або стеганографія, можна вважати DT відповідно до таксономій, представлених у попередньому підрозділі, оскільки вони зміцнюють системи та захищають ресурси ускладнюючи зловмиснику пошук поверхні атаки. Ще кілька нових DT адресують запобігання, наприклад MTD, Random Host Mutation (RHM), Dynamic Instruction Set (DIS) та Єдина архітектура (UA). Більш відомі DT, такі як HP і HT, виявлення адреси. Крім того, ця сфера стосується лише кількох класичних механізмів IT-безпеки, таких як системи виявлення (і запобігання) вторгнень у мережу та хост (N/HID(P)S), системи безпеки та керування подіями (SIEM) і антивірусне ¬програмне забезпечення. Більш нові технології, такі як алгоритми виявлення аномалій, також вирішують цю мету захисту за допомогою методологій, подібних до DT. За словами J.Corey, HP, за своєю суттю, діють як система виявлення вторгнень на основі аномалій [27]. Незважаючи на те, що існує збіг, це твердження вірно лише для механізму виявлення, оскільки весь вхідний трафік можна вважати аномальним і автоматично класифікувати як атаку. Загалом характеристика HP, як і DT загалом, полягає у надсиланні оманливих і неправильних даних, тоді як виявлення аномалій – це лише отримання та аналіз інформації.

1.4 Когнітивні вразливості

Обман за визначенням B.Whaley [9], як і інші автори, сильно спирається на психологію людини. У 1994 році M.C.Libicki [28] ввів термін семантичні атаки. Він відрізняв це від фізичних і синтаксичних атак і обговорював це в контексті ІБ. Пізніше D.B.Buller і J.K.Burgoon ідентифікували та описали 18

релевантних пропозицій щодо обману в контексті загального спілкування. [29]. Вони також досліджували відповідні фактори обману. Через 6 років G.Cybenko та інші [30] ввів термін «когнітивний хакерство» та запропонував кілька заходів протидії, таких як автентифікація джерела, моделювання траєкторії та ігри Ulam.

Термін когнітивний хакерство в основному згадується в контексті обману в новинах або соціальних мережах. K.D.Mitnick [2] опублікував велику роботу про соціальну інженерію, яку він визначив як використання впливу та переконання для обману людей. До сьогодні його роботу часто називають остаточною книгою для обману, хоча вона зосереджена лише на соціальній інженерії.

Автоматизоване навчання методам виявлення обману було досліджено J.Cao та інші [31]. L.Cranor і S.Garfinkel [32] стверджує, що безпека будь-якої соціотехнічної системи базується на трьох елементах: продукт, процес і панорама. У своїй роботі вони проаналізували співвідношення зручності використання та безпеки. Вплив дизайну на обман був додатково проаналізований J.Yuill [33].

D.Ramsbrock та інші [34] провели емпіричне дослідження, щоб дослідити поведінку зловмисників після того, як їм було надано доступ до приманки. Пізніше ця ідея була використана B.Sobesto [35], а також M.Howell [36], щоб дослідити вплив характеристик дизайну, таких як вітальний банер або системні ресурси, на поведінку зловмисників в емпіричних дослідженнях. Більш систематичний підхід до визначення набору відповідних когнітивних упереджень був запропонований D.Fraunholz та інші [37]. Вони проаналізували когнітивні упередження, такі як ті, які представив D. Kahneman [38], щоб узгодити упередження та характеристики дизайну, щоб остаточно вивести основні принципи дизайну.

1.5 Формальні моделі обману

Ряд дослідників розробляють та використовують формальний опис обману в іграх і моделях. Як було зазначено в попередньому підрозділі, J.Pawlick [24] нещодавно опублікував теоретико-ігрову таксономію обману. У своїй роботі вони також розглянули та класифікували 24 публікації обманних ігор. У цьому підрозділі розглядаються дві роботи, опубліковані після рецензування. Wang та інші [39] опублікував гру в обман для конкретного випадку використання. Вони зосереджені на додатках розумної мережі та перевагах безпеки від обману проти розподілених атак на відмову в обслуговуванні.

У 2018 році D. Fraunholz і H.D. Schotten [40] опублікували гру, яка дозволяє моделювати зонди в грі. Як зловмисник, так і той, хто захищається, можуть вибрати зусилля, які вони бажають вкласти в обфускацію систем обману або перевірку систем невідомої природи. Після вибору стратегії зловмисник вирішує, атакувати, досліджувати чи ігнорувати систему. Вони надали евристичне рішення для своєї гри.

2 ТЕХНОЛОГІЇ ТА АЛГОРИТМИ ОБМАНУ

2.1 Оманливе програмне забезпечення та Honeytokens

Термін Honeytoken (HT) був введений у 2007 році A. de Barros [41]. Ідея походить від L.Spitzner [42], який вперше визначив HT як еквівалент HP, з обмеженням представлення сутності, відмінної від комп'ютера. З моменту виникнення обману в інформаційній безпеці було опубліковано величезну кількість концепцій. Вони охоплюють різні типи об'єктів і зосереджуються на створенні оманливих близнюків та їхньому розгортанні. Багато з цих об'єктів отримали конкретні назви, які посилаються на термін honeypot, наприклад, медові слова для фальшивих паролів у базі даних. Діапазон доменів для HT можна розділити за категоріями «Сервер», «База даних», «Автентифікація» та «Файл». Як подальшу абстракцію, HT можуть мати мітки на основі хоста (H) або на основі мережі (N). Обидві відмінності було застосовано для таблиці 1, яка дає огляд останніх методів обману та HT. Ефективним обманом для захисту файлів є надання файлам підроблених метаданих, таких як фальшивий автор, дата створення та модифікації та розмір файлу. До останнього звернувся E.H.Spafford [43], використовуючи структуру Unix Sparse File для обману програм копіювання. N.C.Rowe [44] представив концепцію створення вигляду цілей як HP, користуючись страхом зловмисників перед тим, що їхні методи будуть викриті та ресурси будуть витрачені даремно. Оманливими елементами, які він пропонує для підроблених HP, є використання інструментів HP, таємничі процеси, нестандартні системні виклики в ключових підпрограмах безпеки, видалення та модифікація пакетів і маніпуляції з метаданими, щоб вони виглядали покинутими. Він також передбачає, що це призведе до ще більшого рівня обману, оскільки HP будуть виглядати як підроблені HP. S.Lauren та інші [45] пропонують маніпуляції системними викликами шляхом зміни номерів системних викликів, одночасно відстежуючи оригінальні, які зловмисник, можливо, спробує використати. M.Bercovitch [46] розробила техніку автоматичного генерування записів бази

даних за допомогою алгоритму машинного навчання, який вивчає існуючі записи. Цей генератор НТ захищає конфіденційні дані, позначені вручну, змінюючи їх перед програмою навчання. A.Juels і R.Rivest [47] ввели термін «медове слово» (honey word) для ідеї призначення кількох фальшивих паролів разом із справжнім паролем для облікового запису, щоб зменшити цінність витоку даних. Для розрізнення між медовими словами та справжнім паролем була запропонована спеціальна система автентифікації під назвою honeychecker. Ідею пересилати підозрілі спроби автентифікації на обліковий запис honey запропонували M.H.Almeshekah [22]. M.Lazarov [48] проаналізували зловмисну та звичайну поведінку, публікуючи та відстежуючи електронні таблиці Google, які містили підроблені банківські рахунки та URL -адреси на веб-сайтах. Ідея виправити вразливість системи безпеки, але зробити так, щоб вона виглядала невиправленою, була запропонована F.Araujo та інші [49] і названа honeypatch. Під час спроби використати виправлену вразливість зловмисник перенаправляється на HP. Вони припускають, що широкомасштабне впровадження зменшить зондування вразливостей та інші атакуючі дії. Подібний підхід під назвою «примарні плями» був розроблений J.Avery та E.H.Spafford [50]. Вони пропонують запобігти розкриттю вразливостей у виправленнях безпеки шляхом вставлення підроблених виправлень у справжні виправлення безпеки.

Для захисту веб-серверів D.Fraunholz та інші [51] запропонували оманливі відповідні повідомлення для пом'якшення розвідувальної діяльності. Їхній оманливий веб-сервер надає фальшиві банери з фальшивими версіями сервера та операційної системи, фальшиві записи в robot.txt файлу, динамічно впроваджуваних honeylinks і хибних відповідей на помилки під час доступу до файлу, щоб пом’якшити спроби грубого форсування файлової системи. Техніка обману під назвою MTD не складається з підроблених об’єктів, а наближається до рандомізації та зміни топології мережі та конфігурації хоста. H.Okhravi та інші [52] відзначив MTD на категорії динамічні мережі, платформи, середовища виконання, програмне забезпечення та дані. Зазвичай

використовується MTD методики: ASLR, рандомізація набору інструкцій і рандомізація послідовності коду. E.Al-Shaer та інші [53] запропонував MTD техніку, яку вони називають RHM, що дозволяє рандомізувати хост за допомогою DNS не вимагаючи змін в мережевій інфраструктурі. K.Park та інші [54] поєднали принципи MTD для динамічної генерації мережевих топологій і впровадження приманок (табл. 2.1).

Таблиця 2.1– Огляд методів оманливого програмного забезпечення

Техніка	**Оманлива сутність**	**Домен**	**X**	**H**	**Посилання**
False Honeypot	Honeypot	Сервер	**X**	**c**	[44]
Honeyentries	Таблиця, набір даних	База даних	**c**	**X**	[46], [55],
MTD	topo., net. interf., memory,	Універсальний	**c**	**c**	[52 -54]
Honeyword	Пароль	Аутентифікація	**c**	**X**	[47]
Honeyaccount	Обліковий запис	Аутентифікація	**c**	**X**	[22, 55]
Honeyfile	(Хмарний) файл	Файлова	**c**	**c**	[48, 55]
Honeypatch	Вразливість	Сервер	**c**	**c**	[49, 50]
-	Пам'ять	Сервер	**c**	**X**	[57]
-	Метадані	Файл	**c**	**X**	[44]
HoneyURL	URL	Файл	**X**	**c**	[48]
Honeymail	Електронна пошта	Файл	**X**	**c**	[55, 58]
Honeypeople	Профіль в соціальній	Файл	**X**	**X**	[59]
Honeyport	Мережевий порт	Сервер	**X**	**c**	[55]
Decep. вебсервер	Коди помилок, Robot.txt	Сервер	**X**	**c**	[51]
Інтерфейс ОС	Системний виклик	Сервер	**c**	**X**	[44]

2.2 Еволюція Honeypot

M.Nawrocki та інші [60] нещодавно переглянули системи honeypot та аналіз даних. Тому цей підрозділ зосереджений на забутих, але, на думку авторів, важливих темах їхньої роботи. Спочатку коротко описано промислові HP. Після цього, самоаналіз віртуальної машини Переглянуто системи на основі VMI. Інтелектуальні приманки та автоматизоване розгортання також є важливими темами, але вони детально розглянуті H.Mohammadzadeh та інші [61] і A.Zakaria [62, 63] у 2012 та 2013 роках, тому в даній роботі не

розглядаються. Остання робота в цій галузі включає адаптивні стратегії розгортання [64, 65] та аналіз на основі машинного навчання [66].

Цікава сфера застосування HP - промислове середовище. Промислові системи та критична інфраструктура є важливими для сучасних суспільств [67]. У 2004 році Critical Infrastructure Assurance Group від Cisco Systems Inc. опублікувала HP, спеціально розроблений для промислових мереж [68]. Їхня система підтримувала Modbus, FTP, telnet, HTTP і базувалася на honeyd [69]. Через чотири роки компанія Digital Bond представила першу промислову приманку високої взаємодії (HIHP) [70]. Найвидатнішим промисловим HP є Conpot [71], опублікований Rist у 2013 році. Conpot класифікується як HP із низьким рівнем взаємодії, але підтримує велику кількість промислових протоколів зв'язку, таких як Modbus, IPMI, SNMP, S7 і Bacnet. Багато промислових HP базуються на Honeyd або Conpot . Системи на основі Conpot розширюються до HIHP шляхом їх поєднання з моделюванням системи кондиціювання повітря на основі Matlab/Simulink [72], моделюванням електромережі gridlabd [73, 74] та симулятор мережі IMUNES [75]. K.Wilhoit і Hilt [76] провели експеримент із HP, що імітує програмне забезпечення для моніторингу АЗС. Вони зафіксували кілька атак на систему, наприклад DDoS-атаку, яку вони приписали Сирійській електронній армії (SEA). На це звернули увагу M.Winn та інші [77], що економічно ефективне розгортання HP є вирішальним для їх створення як механізмів безпеки. Вони експериментували з honeyd для економічно ефективного розгортання різних реальних промислових систем управління. Інший метод для цього був запропонований у 2017 році J.D.Guarnizo та інші [78] . Вони перенаправляли вхідний трафік із глобально-розгорнутих так званих червоточних серверів на обмежену кількість пристроїв IoT. Ця техніка створює величезну кількість розгорнутих систем лише з невеликою кількістю реальних апаратних пристроїв.

Таблиця 2.2– Порівняння різних досліджень промислових ТН

		HP		
		Conpot	Honeyd	Інші
Взаємодія	Низький	[71, 75, 79]	[68, 77, 80]	[76, 81 -85]
	Високий	[73, 74, 86]	-	[70, 72, 78, 87 -91]

A.Jicha та інші [92] провели поглиблений аналіз Conpot. Шестимісячне довгострокове дослідження з Conpot було проведено J.Cao та інші [79]. A.V.Serbanescu та інші [93], навпаки, використовував широкомасштабну honeynet, яка пропонувала різноманітні промислові протоколи для 28-денного експерименту.

High-interaction Honeypot на основі VMI: автори вважають, що HP на основі VMI є найновішою, а також найбільш перспективною технологією для HIHP . У 2017 році S.Sentanoe та інші [94] порівнювали різні (HIHP) технології. Вони також вирішили використовувати VMI як технологію для своїх експериментів із SSH HP. M.Vrable та інші [95] представив Potemkin virtual honeyfarm, що базується на Xen і може клонувати еталонну віртуальну машину для кожної атаки. Argos був опублікований G.Portokalidis та інші [96] у 2006 році. Він здатний проводити аналіз шкідливих програм для відбитків пальців. Пізніше проект Honeynet [97] модифікував Sebek, включивши VMscope [98]. VMwatcher [99] може клонувати віртуальну машину. Клонована віртуальна машина контролюється AV, IDS або криміналістичними методами високого рівня. B.Hay і K.Nance [100] запропонували іншу ідею для криміналістичних методів високого рівня, запускаючи криміналістичні інструменти безпосередньо на віртуальній машині, коли вона призупинена. B.Dolan-Gavitt та інші [101] автоматизували генерацію сценаріїв самоаналізу шляхом перекладу гостьових додатків у вихідний код самоаналізу. Робота J.Pfoh та інші [102] зосереджена на продуктивності та гнучкості для трасування системних викликів на основі QEMU/KVM . Timescope [103] був запропонований у 2011 році. Він здатний записувати вторгнення та

відтворювати його кілька разів для спостереження за різними аспектами. S.Biedermann та інші [104] опублікував фреймворк, який здатний клонувати віртуальну машину в реальному часі у разі атаки. Потім зловмисник перенаправляється на клонований екземпляр, і системні виклики, а також мережева активність і стан пам'яті відстежуються. Гібридна архітектура була запропонована T.Lengyel та інші [105], вони зосереджені на виявленні зловмисного програмного забезпечення за допомогою комбінації приманок з низьким рівнем взаємодії (LIHP) і HIHP. Пізніше вони опублікували вдосконалену версію своєї архітектури з механізмом клонування з potemkin virtual honeyfarm [95] та механізмом моніторингу з їх попередньої роботи. У 2014 році T.K.Lengyel та інші [106], знову ж таки, запропонував розширену версію свого фреймворку з додатковими функціями для автоматизованого розгортання та аналізу зловмисного програмного забезпечення. Переваги HP на основі VMI для систем обману на основі MTD були відзначені V.E.Urias та інші [107] . J.Shi та інші [108] представив структуру з інтегрованим модулем для аналізу трасування системних викликів.

Нещодавно ідея використання контейнерів Linux як альтернативи для віртуальних машин була досліджена Кедровітчем [112] . Вони дійшли висновку, що контейнери добре підходять для розгортання на малопотужних пристроях, але їх легко виявити.

Анти-Honeypot як і будь-яке інше програмне забезпечення, honeypots схильні до програмних помилок. У 2004 році N.Krawetz [113] запропонував ідею ідентифікувати приманки шляхом дослідження функціональності змодельованих служб. У своїй роботі він запропонував відправляти електронні листи з SMTP-HP собі. Якщо листи не надходять, запропоновані функції недоступні, а середовище системи є підозрілим. X.Fu та інші [114] пропонують кілька методів виявлення віртуальних HP, таких як honeyd, на основі тимчасової поведінки. Вони використовують підходи на основі Ping, TCP і UDP для визначення часу проходження пакета туди й назад. S.Mukkamala та інші [115] інтегрували машинне навчання в виявлення приманок на основі

тимчасової поведінки. Заходи проти цього типу зняття відбитків пальців розроблені Shi та Cao в 2008. Вони пропонують honeyanole для пом’якшення відбитків пальців на основі тимчасової поведінки шляхом перенаправлення трафіку. S.Bahram та інші [116] досліджували VMI і виявили, що він схильний до змін у розташуванні пам'яті ядра. Зміни в макеті збільшують семантичний розрив і роблять VMI неможливим. Питання відсутності налаштування обговорювали D.Sysman та інші [117]. Вони використовували Shodan, щоб визначити адреси понад 1000 розгортань conpot на основі вигаданої назви компанії за замовчуванням. Таксономію методів антивізуалізації та антиналагодження опублікували X.Chen та інші [118]. Вони поділяють ці техніки на: абстракцію, артефакт, точність, рівень доступу, складність, ухилення та імітацію. Крім того, вони опублікували метод дистанційного зняття відбитків пальців для віртуалізованих хостів. Іншу таксономію методів виявлення HP опублікували J.Uitto та інші [119]. Вони визначають часовий, операційний, апаратне забезпечення та середовище як фундаментальні класи.

У 2016 році R.Dahbul та інші [120] представив модель загрози для HP. Ця модель об’єднує атаки в три групи: отруєння, компрометація та навчання. Вони також пропонують низку виправлень для honeyd, Dianaea та Kippo.

O.Hayatle та інші [126] запропонували метод, заснований на доказах Демпстера-Шейфера, який поєднує використання багатофакторного прийняття рішень для виявлення HP. Пізніше вони опублікували роботу про виявлення HIHP на основі моделей Маркова [127]. Wang розслідує зловмисників, які знають Honeypot, або ботнети [128] і Costarella та інші [129]. Про ризики відображених атак зловживанням говорив M.Husak [130] (таблиця 2.3).

Таблиця 2.3 – Порівняння досліджень для Honeypot на основі VMI

Автор	**рік**	**Об'єкт досл.**	**Вірт. середовище**	**Моніторинг**
Vrable [95]	2005	Generation	QEMU / KVM	Недоступний
Portokalidis [96]	2006	Signatures	QEMU / KVM	Volatile-пам'ять,
Jiang and Wang [98]	2007	Stealthniness	QEMU / KVM	Системні виклики
Jiang [99]	2007	Semantic gap	VMware, Xen, QEMU / KVM, UML	Вся система
Hay and Nance [100]	2008	High-level forensics	Xen	Емульовані утиліти Unix
Tymoshyk [109]	2009	Semantic gap	QEMU / KVM	Системні виклики
Honeynet Project [97]	2010	WMI-Sebek	QEMU / KVM	Системні виклики
Dolan-Gavitt [101]	2011	Automation	QEMU / KVM	Системні виклики
Pfoh [102]	2011	Perfomance	QEMU / KVM	Системні виклики
Srinivasan and Jiang [103]	2011	Record and replay	QEMU / KVM	Системні виклики
Biedermann [104]	2012	Generation	Xen	Системні виклики, моніторинг мережі, стан пам'яті
Lengyel [105]	2012	Generation, file capturing	Xen	Стан пам'яті
Lengyel [110]	2013	Routing	Xen	Стан пам'яті
Beham [111]	2013	Visualization, perfomance	KVM, Xen	VMI-honeymon [105]
Lengyel [106]	2014	Automation	Xen	Системні виклики, файлова система
Urias [107]	2015	Generation	KVM	Системні виклики, процеси, файлова система
Shi [108]	2015	System call analysis	KVM	Системні виклики
Sentanoe [94]	2017	SSH	Unknown	Системні виклики

2.3 Технологічні дослідження

Було проаналізовано тип і тривалість розгортання, а також DT, успіх, вектор атаки та формат отриманих дані були частиною розслідування. Після цього аналізуються роботи, що оцінюють ресурси обману. Вони згруповані відповідно до ресурсів обману, які вони розглядають, а також показників і характеристик, які розглядаються авторами. Зведення технологічних досліджень ресурсів обману наведено в таблиці 5. Було знайдено 14

заслуговують на увагу прикладів польових досліджень технологій обману. F.Cohen та інші [131] провели аналіз поведінки реальних зловмисників у вразливій системі. Вони познайомили експертів із безпеки з атакуваною системою, у якій вони розгорнули обманні ресурси та відстежували їхню поведінку. Результати були отримані за допомогою анкети, на яку відповіли експерти після експерименту. Fraunholz та інші провели два польових дослідження. У першому дослідженні Фраунгольц і Шоттен [51] запропонували серверні механізми обману, щоб перешкодити зловмисникам. Фальшиві банери, фальшивий Robots.txt, підроблена відповідь на помилку, адаптивна затримка та медові файли були представлені для вивчення поведінки зловмисників за цих обставин. Відстежено 1200 доступів. У другому дослідженні [132, 133] вони проаналізували поведінку зловмисників, яку спостерігали шість приманок, розгорнутих на одному споживачеві та п’яти серверах веб-хостингу, протягом 222 днів. Використані LIHP відстежували майже 12 мільйонів спроб доступу . Загальні протоколи, такі як HTTP, HTTPS, FTP, POP3, SMTP, SSH і Telnet, були запропоновані honeypots. На додаток до цього, промислові протоколи Bacnet, Modbus і S7 були емульовані, щоб отримати уявлення про ландшафт загроз для промислових застосувань. Лазаров та інші [48] навмисно злив підроблену конфіденційну інформацію в електронних таблицях Google. IP- адреси містилися в цих електронних таблицях і мали заманити зловмисників. Було відстежено 174 кліки, а також 44 відвідування за 39 унікальними IP-адресами. Лю та інші [134] дотримувалися подібного підходу до публікації очевидно конфіденційної інформації. Ключі SSH просочилися на github, спонукаючи зловмисників підключитися до HP на базі Cowrie (таблиця 2.4).

Таблиця 2.4 – Огляд заходів протидії HP

Метод		**Ціль**	**Результат (Mitigation)**	**Посилання**
Temporal behavior	Measure RTT to expose correlations between IP addresses	honeyd, virtual honeypots	Simulating timing behavior	[114, 115,118, 121, 122]
Stack finger printing	Send corrupted packets and analyze responses	Simulated communication stacks	Implementation of full TCP/IP stack	[27, 123]
Functional probing	Use provided functions and ver ify status	SMTP,DNS Implementatio	Implementation of full functionality	[113, 121]
System call behaviour	Anomalies in temporal behavior or memory locations	Linux systems Simulation	Sumulating timing behavior, KASLR	[27, 121, 122, 124]
Network traffic	Analyze RX and TX network eg number of bytes	Network based data exfiltration eg Sebek	Hinder network monitor traffic ing, VMI, Proxy	[121]
UML detection	dmesg output, network device, /proc/, memory layout	UML based host isolation	Manipulating tools to show related information	[122]
VMware de tection	Hardware eg MAC address, I/O backdoor	VMware based host isolation	Customize hardware, patch I/O backdoor	[27, 122]
Debugger detection	Use ptrace() function, IsDebug eg Cuckoo function or memory search for 0xCC	gerPresent()	-	[122, 125]
Semantic gap	Manipulate kernel data structure VMI	-	-	[116]
Customize	Search for default strings	-	Customize systems	[27, 117, 120-122]

Було відстежено близько 31000 унікальних паролів, а також поведінка користувачів після входу в систему протягом двох тижнів. O.Zohar та інші [135] створив комплексну організаційну мережу, що складається з користувачів, поштових даних, документів, профілів браузерів та інших IT - ресурсів. У цю мережу були введені пастки та приманки. Подібно до роботи Cohen та інші [131] 52 фахівці з безпеки прийняли завдання Capture the Flag (CTF) у цій мережі та спробували її скомпрометувати. Метою цього експерименту було визначення найкращих засобів для різних організаційних мереж, а також найкращих стратегій розгортання пасток і приманок у комп'ютерних мережах. Хауелл [36] використав *Sebek,* щоб отримати

уявлення про поведінку зловмисника після компрометації. Для цього експерименту використовувався набір даних Джонса [136], зібраний *HIHP*, який містить 1548 звернень 478 зловмисників. *Sobesto* [137] проаналізували реакцію зловмисника на конфігурацію системи та банери. Невикористані IP-адреси університетської мережі були використані для розгортання приманок Dionaea з низкою вразливостей і моніторингу поведінки зловмисників після вторгнення. Це відбувалося з 17 травня по 31 жовтня, під час якого було відстежено 624 сесії за допомогою інструменту honeypot Spy. Маймон та інші [138] провели два цикли своїх експериментів: один на два і один на шість місяців. За цей час було налаштовано 86 і 502 комп'ютери відповідно та доступно через Інтернет. Після підключення деякі показали попереджувальні банери, а деякі ні. Вони містили різні вразливі точки входу, створені за допомогою Sebek і OpenVZ як шлюз, які привернули 1058 і 3768 інцидентів проникнення відповідно. Метою було визначити вплив попереджень на зловмисників. Хейркхак та інші [139] проаналізували облікові дані, які використовуються під час спроб входу. Вісім HP з увімкненим підключенням SSH були представлені в шести різних мережах університетських містечок протягом семи тижнів. Було зафіксовано 98180 підключень із 1153 унікальних IP -адрес у 79 країнах. Жан та інші [140] використовували різні типи HP, а саме Dionaea, Mwcollector, Amun і Nepenthes. Вони використовувалися для статистичного аналізу шаблонів атак. Виділено п'ять періодів, протягом кожного з яких 166 зловмисників атакували вразливі сервіси: SMB, NetBIOS, HTTP, MySQL і SSH. Salles-Loustau та інші [141] проаналізували поведінку зловмисника на основі шаблонів натискання клавіш. Три приманки з різними конфігураціями зафіксували 211 сеансів атак протягом 167 днів. Бертьє та інші [142] зосереджено на поведінці та діях зловмисника після компрометації системи. Для цього було відстежено 24 різні дії як індикатори різних типів поведінки. HP були налаштовані на вісім місяців в університетських мережах, були доступні через SSH і захоплювали 20335 введених команд під час 1171 сеансу атаки. Ramsbrock та інші [143] дотримувалися подібного підходу.

Чотири Linux HP були представлені в університетській мережі, доступній через SSH з обліковими даними, які легко вгадати, протягом 24 днів. Дії зловмисників відстежувалися за допомогою syslog-ng для захоплення команд, strace для реєстрації системних викликів і Sebek для збору натискань клавіш. Зібрано 269262 спроби атаки з 229 унікальних IP. Можна побачити, що сильний акцент у застосуванні технологій обману лежить на HP. Крім того, часто використовуються реальні системи, розширені можливостями моніторингу. За визначенням, вони вважаються HIHP. Лише дві з описаних вище робіт використовують суттєво різні DT, а саме Fraunholz і Schotten [51] і Лазаров та інші [48] . Було знайдено сім вартих уваги прикладів технологічних опитувань про ДТ . Вони зведені в таблицю 5. У цих опитуваннях було визначено одинадцять ознак оцінювання, які були спільними для принаймні двох опитувань. Вони пронумеровані від F1 до F11 і визначаються таким чином: Інтерактивність (F1), масштабованість (F2), юридичні чи етичні міркування (F3), тип (F4), розгортання (F5), переваги та недоліки порівняно з іншими типами оборонні технології (F6), якість і тип даних і отримані знання (F7), тип ресурсу DT (F8), технічний спосіб розгортання та тип DT (F9), здатність до виявлення та запобігання виявленню (F10) та розширюваність (F11) (табл.2.5).

J.Smith [144] розглядає різновиди DT s. В якості загальних понять розглядаються ГП, а також ГТ і медові сітки, медові пастки та медові сітки. З точки зору інструментів, він аналізує VMWare, Chroot і Honeyd щодо їх зручності використання як інструментів DT . Навроцький та інші [60] аналізують і порівнюють 68 різних HP. T.Grudziecki та інші [145] аналізують 33 різні HP. Лише вони розглядають надійність і підтримку інструментів як функцію оцінки. Брінгер та інші [149] оглядає понад 80 статей, які представляють технології HP, 60 з яких, імовірно, мали значний вплив на сферу DT. Гірдхар і Каур [146] аналізує п’ять різних HP: ManTrap, Back officer friendly, Spectre, Honeyd і Honeynet.

Таблиця 2.5 – Огляд документів технологічного огляду

Робота	**рік**	**DT**	**FI**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**	**F11**	**інший**
Сміт [144]	2016	HP/HT	c	X	c	c	X	X	c	c	c	c	X	-
Навроцький та інші [60]	2016	HP	c	X	X	c	X	c	c	c	c	X	X	-
Grudziecki та інші [145]	2012	HP	c	c	X	c	c	c	c	X	c	X	c	Віднос. / Доп.
Гірдхар і Каур [146]	2012	HP	c	X	X	c	X	c	X	X	X	X	X	-
Горжелак та інші [147]	2011	HP	X	c	X	X	c	X	c	X	c	X	c	-
Лахані [148]	2003	HP	X	X	c	c	c	X	X	X	c	X	X	-

K.Gorzelak та інші [147] порівнюють honeypots з іншими інструментами виявлення інцидентів і реагування. У їхній роботі розглянуто 30 різних інструментів або послуг і дванадцять різних методологій. A.D.Lakhani [148] у своїй роботі порівнює чотири різні медовини: LaBrea, Spectre, Honeyd і ManTrap. Таким чином, більшість робіт розрізняють тип HP, а саме дослідження та виробництво. Двома другими найважливішими характеристиками оцінки є тип технології обману та якість і тип даних, які генерує ресурс. Тільки Grudziecki та інші [145] розглянули надійність і технічну підтримку DT.

2.4 Юридичний та етичні аспекти питання

У цьому розділі розглядаються правові та етичні аспекти. Пастка, конфіденційність і відповідальність обговорюються в більшості літератури. Тому таблиця 6 розрізняє розглянуту літературу за цими предметами. Крім того, роботи згруповані за країною, яку вони розглядають, і якщо також враховуються етичні аспекти.

Шпіцнер [42] і I.Mokube та інші [150] обговорював Четверту поправку до Конституції США, тоді як Закон про прослуховування телефонних переговорів аналізував A.Burstein [151] P.Ohm та інші [152] або [42]. Дорнсайф та інші [153] обговорювали аспекти кримінального та деліктного права. Патріотичний акт був розглянутий Шпіцнером [42] . A.Burstein [151] пов'язує Закон про захист авторських прав у цифрову епоху, Закон про комп'ютерне шахрайство та зловживання та Закон про збережені комунікації. B.Scottberg та інші [154] і Y.Jain [155] зробив висновок, що ввести будь-кого в пастку можуть тільки правоохоронці. Крім того, Y.Jain робить висновок, що «організації чи навчальні заклади не можуть бути звинувачені у пастці». Конфіденційність гарантується Європейською конвенцією з прав людини [152] і розглядається в контексті обману Y.Jain [155], I.Mokube [150], B.Schaufenbuel [156] і P.Sokol [157, 158]. D.Fraunholz та інші [159] стосується конфіденційності, авторського права, самозахисту та законодавства, що стосується конкретної сфери, наприклад, прав правоохоронних органів, дослідницьких інститутів і провайдерів телекомунікацій. Відповідальність покривається B.Schaufenbuel [156], Кампелл [160], D.Fraunholz та інші [159] і Nawrocki та інші [60]. Шауфенбюель [156] запропонував пропозиції щодо функціонування та використання DT для пом'якшення правових ризиків. 16 робіт, що висвітлюють аспекти юрисдикції США, на відміну від п'яти, що охоплюють європейське право, демонструють порівняно вищий науковий інтерес до права США. Нарешті Кемпбелл [160] порівнює США з африканською юрисдикцією, тоді як Warren [161] стосується законодавства Австралії (табл.2.6).

На відміну від 22 робіт, які висвітлюють правові аспекти, є лише шість робіт, які також стосуються етичних питань з DT. M.Dornseif та інші [153] описують проблему підвищення безпеки Інтернету шляхом впровадження вразливостей у загальнодоступний Інтернет. Питання про те, як можна захистити Інтернет, додавши слабкі місця, а також про моральну доцільність участі людей в експерименті ¬без їхнього відома та згоди задає Хольц [168]. Кемпбелл [160] ставить під сумнів моральні наслідки спонукання когось до

вчинення злочину, а також згадує проблему «підливання масла у вогонь», коли система робиться вразливою до атак. N.C.Rowe і J.Rrushi [162] приписують етичну відповідальність DT програмісту, розглядаючи при цьому проблему самозміни коду та штучного інтелекту.

Таблиця 2.6 – Огляд юридичних та етичних досліджень ДТ

	Країна			**Юридичні аспекти**			
Робота	**США**	**Європа**	**інші**	**Пастка**	**Конфіденційність**	**Відповідальність**	**Етика**
Дорнсайф	с	X	X	X	с	X	с
Шпіцнер [42]	с	X	X	с	X	X	X
Скоттберг	с	X	X	с	с	X	X
джайни [155]	с	X	X	с	с	X	X
Мокубе [150]	с	X	X	с	с	с	X
Привізник	с	X	X	X	X	X	с
Шауфенбюель	с	X	X	с	с	с	X
Сокіл [157]	X	X	X	X	с	X	X
Кемпбелл [160]	X	X	Африканці проти США	с	с	с	с
Навроцький	с	с	X	с	с	с	X
Роу [162]	с	X	X	с	X	X	с
Бурштейн	с	X	X	X	X	X	X
Редкліфф	с	X	X	X	X	X	X
Рубін [164]	с	X	X	X	с	с	X
Карида [165]	с	X	X	X	X	X	X
Беллоні [166]	с	X	X	X	X	X	X
Сокіл [158]	X	с	X	X	с	X	X
Ом [152]	X	с	X	X	с	с	X
Ненс [167]	с	X	X	X	X	X	с
Воррен [161]	X	X	Австралія	X	X	X	X
Дорнсайф	X	с	X	X	X	X	с
Fraunholz та	X	с	X	с	с	с	X

На відміну від 22 робіт, які висвітлюють правові аспекти, є лише шість робіт, які також стосуються етичних питань з DT. M.Dornseif та інші [153] описують проблему підвищення безпеки Інтернету шляхом впровадження вразливостей у загальнодоступний Інтернет. Питання про те, як можна захистити Інтернет, додавши слабкі місця, а також про моральну доцільність участі людей в експерименті ¬без їхнього відома та згоди задає Хольц [168]. Кемпбелл [160] ставить під сумнів моральні наслідки спонукання когось до

вчинення злочину, а також згадує проблему «підливання масла у вогонь», коли система робиться вразливою до атак. N.C.Rowe і J.Rrushi [162] приписують етичну відповідальність DT програмісту, розглядаючи при цьому проблему самозміни коду та штучного інтелекту.

Включення DT до рекомендацій тільки починається. Німецьке федеральне відомство з інформаційної безпеки (BSI) опублікувало рекомендації щодо базового захисту [169], в яких згадується використання HP s. Вони розглядають їх як механізми виявлення аномалій, при цьому не згадуючи DT прямо. Однак неявно вони описані, наприклад, у підробці банерів сервера.

2.5 Нетривіальні шляхи застосування техніки обману

Сучасні кібероперації еволюціонують від прямих атак і оборони до складних кібероперацій, які пов'язані з обманом. Оскільки обмани включаються в кібератаки та захист, слід визначити елементи обману для реагування на кібероперації. Якщо для виявлених елементів обману можна вжити відповідних контрзаходів, вони можуть отримати стратегічну перевагу в кіберпросторі. Пов'язані з цим дослідження кібер-обману включають розробку інструментів реагування для зловмисників з оборонної точки зору та розробку методів атаки, які використовують когнітивні вразливості людини. Інші дослідження класифікували інструменти обману відповідно до їх цілей і вивчили процедури ефективного здійснення обману. Однак існуючі дослідження не розглядають конкретні цілі обману і не класифікують обман у складних кіберопераціях. Класифікація обману в кіберопераціях вимагає поділу кіберпростору на фізичний, логічний і персонний шари, цілі кібероперацій повинні бути ідентифіковані від машин до людей, а процедури обману повинні бути визначені від TTP до цілей. У відповідь на це пропонується «модель дерева обману», яку можна класифікувати з точки зору

кібер-брехливого TTP. Модель дерева обману може відрізняти цілі від людей і машин з точки зору атаки і оборони і систематично встановлювати ефекти, тактику, прийоми і процедури обраних цілей. Було застосовано та проаналізовано три випадки для перевірки працездатності моделі дерева обману. Перший випадок - це кіберінцидент, що стався на KHNP в 2014 році, в якому була проведена брехлива атака на людей, другий випадок - використання технології Honeynet для обману зловмисника, а третій випадок - використання технології Anti-Ransomware для обману шкідливих програм [170].

2.6 Дослідження нових алгоритмів серед технології обману

Надійний алгоритм мультимодального виявлення обману [171]. Автоматичне виявлення обману з відео набуло першорядного інтересу через їх застосовність у різних реальних додатках. Записані відео містять різну інформацію, таку як тимчасові варіації обличчя, лінгвістика та акустика, які можна використовувати разом, щоб автоматично виявити обман. У цій роботі запропонували новий підхід, заснований на мультимодальній інформації, такій як аудіо, лінгвістичні (або текстові) та невербальні особливості. Запропонована мультимодальна структура виявлення обману базується на поєднанні рішення з аудіо-, текстових та невербальних ознак за допомогою голосування більшості. Запропонована мультимодальна система обману побудована на аудіосистемі на основі Cepstral Coefficients (CC) та спектрального регресійного аналізу ядра дискримінантів (SRKDA) аудіопослідовностей фіксованої довжини. Текстова система базується на функціях bag-of-n-grams та лінійному класифікаторі Support Vector Machine (SVM), тоді як невербальні ознаки класифікуються за допомогою класифікатора AdaBoost. Проводяться великі експерименти на загальнодоступному в реальному житті наборі відеоданих про обман для

оцінки ефективності запропонованої схеми. Отримані результати 25-перехресної перевірки показали точність виявлення обману, яка на 97% перевершує як найсучасніші методи, так і людські показники на всьому наборі даних.

Лабіринт обману: ігровий теоретичний захисний механізм Стекельберга для інтранет-загроз [172]. Інтрамережі в сучасних організаціях стикаються з серйозними порушеннями даних і критичними зловживаннями ресурсами. Повторно використовуючи облікові дані користувачів із скомпрометованих систем, зловмисники розширеної постійної загрози (APT) можуть переміщатися латерально у внутрішній мережі. Новий перспективний підхід, який називається технологією обману, робить мережевого адміністратора (тобто захисника) здатним розгортати манки, щоб обдурити зловмисника в інтрамережі і заманити його в HP. Тоді захисник повинен розумно виділяти манки потенційно невпевненим господарям. На жаль, існуючі моделі розподілу оборонних ресурсів, пов'язані з APT, неможливі через нехтування багатьма реалістичними факторами. У цій роботі розгортання приманки здійсненною, пропонуючи теоретико-ігрову модель під назвою APT Deception Game для опису взаємодій між захисником і нападником. Більш конкретно описано проблему розгортання манки на дві підпроблеми і робимо проблему вирішуваною. Враховуючи найкращу реакцію нападника, який знає про стратегію розгортання захисника, надано елітарний генетичний алгоритм резервування для вирішення цієї гри. Результати моделювання демонструють ефективність нашої стратегії розгортання в порівнянні з іншими евристичними стратегіями.

Моделювання впливу обсягу і термінів обману в змодельованих мережевих сценаріях [173]. Моделююче середовище в реальному часі («Гра в обман»), яке и використали для оцінки і моделювання прийняття рішень хакерами при наявності обману. В експерименті, використовуючи повторну Гру обману (N = 100 учасників), ми проаналізували вплив двох факторів на рішення учасників атакувати комп'ютерну мережу: кількість використаного

обману і терміни обману. Протягом 10-ти випробувань атаки кількість використаного обману маніпулювали на 2-х рівнях: низькому та високому. Термінами обману маніпулювали на 2-х рівнях: ранньому і пізньому. Результати показали, що використання пізнього і високого обману викликало зниження атак на звичайний веб-сервер в порівнянні з раннім і низьким обманом. Крім того, зроблено когнітивну модель прийняття рішень хакерами, використовуючи теорію інстанційного навчання (IBL), теорію рішень з досвіду. Параметри, отримані з моделі, допомогли пояснити причини експериментальних результатів.

Проектування системи захисту від нападу APT на основі динамічного обману [174]. Атака Advanced Persistent Threat (APT) має характеристики складних засобів атаки, велику тривалість і велику шкідливість. Виходячи з ідеї динамічного обману, в роботі була запропонована структура системи захисту APT, а також проаналізовано процес захисту від обману. У роботі був запропонований гібридний механізм зв'язку шифрування на основі сокету, метод генерації динамічних IP-адрес на основі SM4, метод динамічного вибору часу на основі алгоритму Viterbi і динамічний механізм розподілу політики на основі DHCPv6. Випробування показують, що система захисту може динамічно змінюватися і ефективно захищати атаки APT.

Метод, заснований на MD5 і час запобігання обману в електронній комерції [175]. Прагнучи знизити ризик перехоплення даних і запобігти обману в електронній комерції, в даній роботі пропонується метод, заснований на MD5 і часі, який може ефективно запобігти обману при передачі даних. Метод використовує традиційний алгоритм MD5 для шифрування та перевірки важливих даних, а потім перевірки дійсності та легітимності даних за допомогою часу.

3 ІМПЛЕМЕНТАЦІЯ ТЕХНОЛОГІЇ ОБМАНУ DECEPTION

3.1 Архітектура і варіанти постачання

Для розуміння моделі реалізації, зупинимось на тому, що вважають за першу інкарнацію технології Deception. Початок можна простежити з кінця вісімдесятих та початку дев'яностих років. Honeypot (HP) – мережевий об'єкт, що має єдину ціль: спокушувати зловмисника атакувати його, в мімікрії уподібнившись під автентичну жертву. Даний об'єкт не несе ніякої іншої цінності в мережі, в котрій він встановлений, з ним не ведеться ніяких легітимних мережевих взаємодій. При атаці на себе, він займається фіксацією даних атакуючого і його цифрового сліду так й самого процесу, систематизуючи його. Побічною ціллю honeypot'а вважається стримування у просування атакуючого по інфраструктурі мережі, змушуючи його витрачати час на вивчення хибної моделі, запропонованої конфігураторами захисту. Виглядати дана система захисту може як типова операційна система, що емулює робоче місце працівника, сервер, окремий сервіс тощо (рис.3.1).

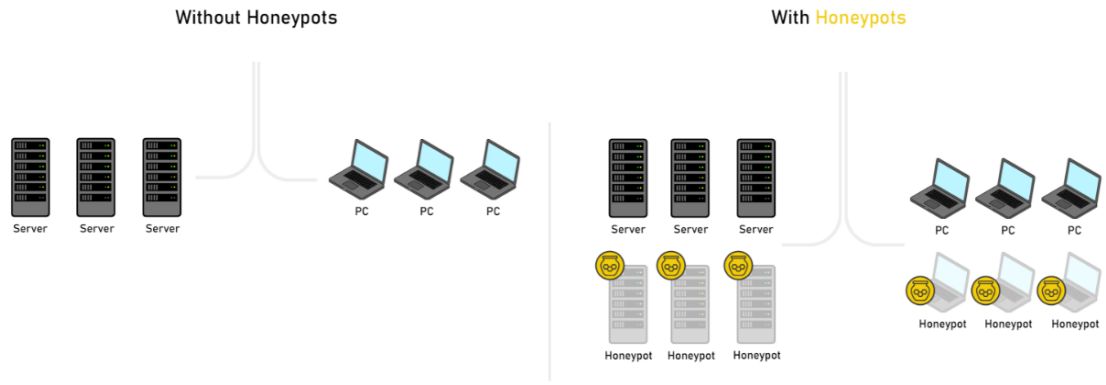

Рисунок 3.1 – Схема організації зв'язків з використанням Honeypot.

Варто вказати на «ложку дьогтю» в цьому «горщику меду», а саме:

– потреба окремої конфігурації кожного мімікруючого серверу;

– вузли симуляції не взаємодіють між собою та з елементами справжньої інфраструктури (відсутній цифровий слід, менші шанси визначення як цілі атаки);

– об’єкти приманки, як правило, не організовані в єдину, централізовану систему.

Виправляючи недоліки, на зміну даній технології і прийшла технологія DT як більш розумна ітерація активної протидії зламу.

DT відноситься до рішень класу Intrusion Detection System (IDS) – системам виявлення вторгнень. Головна ціль даної системи – виявляти спроби небажаного доступу до мережі (мережеві атаки).

Відмінність технології DT від HP простежується у формі об’єкту: на відміну від архаїчного попередника, технологія обману наступної ітерації має форму централізованої системи управління фіктивними мережевими об’єктами (пастками (decoy’s). Кожна така пастка є окремим honeypot’ом, що зв’язаний з центральним вузлом (рис.3.2).

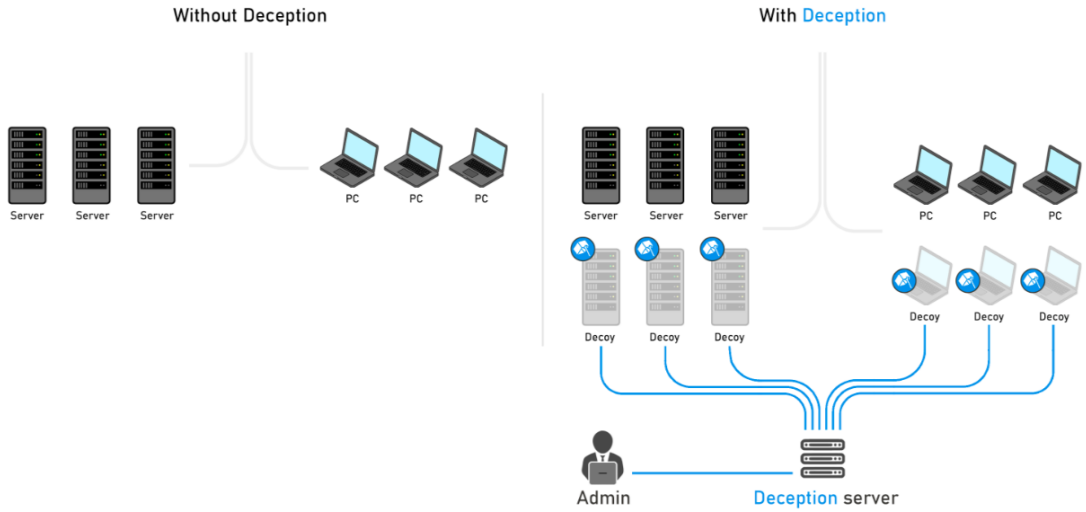

Рисунок 3.2 – Схема організації зв’язків з використанням Deception.

Варто оглянути функціональні можливості даного класу систем, а саме: автоматичне розгортання та збір даних з робочих вузлів. Потенційною проблемою DT виступає трудомісткість первинного налаштування. Рішеннями даного недоліку можуть слугувати контрміри: інтегрування сервісу DT до системи аналізу трафіку, з подальшою автоматичною генерацією пасток на базі отриманої інформації, або проведення активного сканування мереж, пасивне прослуховування трафіку (при технічній

можливості) тощо. Звісно, лишається і змога ручної конфігурації, але це вже виступає недоліком даної технології. Про це буде йти мова в кінці розділу детальніше.

До поняття DT можуть відноситись і інші модулі: деякі компоненти дають змогу спростити конфігурацію та автоматичне розгортання системи, інші ж роблять пастки більше схожі на справжні мережеві сервіси, інакші ж – слугують Пастками на хибні цілі для хакерів.

Окремі компоненти можуть вирішувати суміжні завдання – наприклад, реагувати на інциденти, збирати індикатори компрометації з робочих станцій та шукати на них вразливе ПЗ.

Архітектурні рішення DT, традиційно, діляться на кілька класів:

− on-premise рішення, що повністю розгортаються на обладнання замовника;

− переважно on-premise рішення, частину функціоналу яких реалізована у вигляди хмарних сервісів та служб;

− гібридні рішення: on-premise частково, при потребі захисту хмарних платформ – реалізовані рішення захисту в хмарі;

− хмарні рішення, реалізовані в повному обсязі в хмарі (аналог on-premise, але лише для хмар);

− змішані хмарні рішення (на приклад, модулі пасток та центри керування розгортаються в своїх хмарах, а аналітика та фіктивна база – вже розгорнуті в інших).

Якщо говорити про варіанти постачання, то вони залежать від обраної архітектури рішення:

− попередньо-конфігурований образ віртуальної машини (OVA);

− інтегровані ISO образи ОС з попередньо-встановленими пакетами та скриптами розгортання (характерно для Linux);

− дистрибутивне ПЗ, що дає змогу встановити рішення на вже наявну інфраструктуру (характерно для Windows);

– програмно-апаратний комплекс, що йде із власним, завчасно налагодженим обладнанням.

Вибір потрібного варіанту пов'язаний не лише із запропонованими до імплементації рішеннями, але й вимогами законодавства в країні, або корпоративною політикою (корпоративним стандартом організації). Варто відзначити, що в якості хмарних платформ, в основному, використовуються AWS, Azure та GCP.

3.2 Агенти

Агент – програма, яка встановлюється на реальні робочі станції користувачів чи серверні одиниці. Вона вміє спілкуватись із сервером DT та виконувати його команди, або передавати на центр керування користі дані.

Серед рішень класу DT є продукти, в склад котрих входить агент, як є й такі, функціонування яких полягає у його відсутності (рис.3.3).

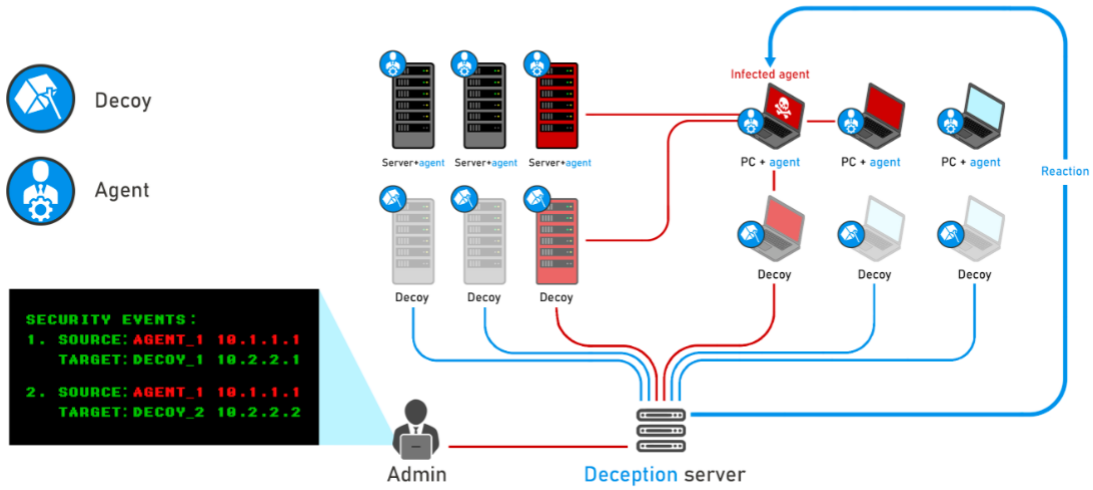

Рисунок 3.3 – Схематичне зображення агента.

Зоною відповідальності та функціональних можливостей агента є:

– збір даних про стан робочих станцій;

– розповсюдження приманок;

- емуляція активності мережі;
- реагування на інциденти (ручний чи автоматичний режими);
- збір даних для форензики;
- додаткові критерії функціонування (побажання клієнтів).

Діяльність агента має бути прихованою від людини, який працює за комп'ютером. Це покликано загрозою намірів видалити агента чи його компоненти; уникненню дискомфорту при усвідомленні роботи даного ПЗ на робочій станції; а також простежуваність: що бачить користувач те й побачить зловмисник. Саме в цих причинах агентські рішення DT варто конфігурувати таким чином, аби користувач не знав про його наявність чи, взагалі, факт існування (а також сліду виконання). Якщо ж це неможливо – варто мінімізувати наслідки імплементації агента.

Як результат, агенти, зазвичай, працюють в режимі розширених повноважень, у вигляді драйверу для Windows чи модуля ядра для Linux. Це дає змогу робити перехват системних викликів для забезпечення режиму конфіденційності, а також не дає змоги користувачу впливати на роботу агента чи видаляти його.

3.3 Прийоми мімікрії

Завдання DT як технології – переконати атакуючого в тому, що всі пастки і взаємодії між ними – реальні, цінні та використовуються (являючись інформаційними активами, втрата котрих призведе до збитків). Тобто домінуючим завданням є зробити хибні цілі привабливими для атаки. В сучасних системах є ряд компонентів, що відповідають за ці цілі.

3.3.1 Пастки

Пастки – фіктивні вузли мережі, що дають змогу виявити зловмисника і знешкодити його деструктивну діяльність на справжніх вузлах, сповільнив

горизонтальний рух по мережі. Це можуть бути симуляційні чи емульовані робочі станції, мережеве обладнання, сервери та різні сервіси.

Історично, з часів HP, пастки поділяються на 4 основні типи:

- низько-інтерактивні пастки (Low interactive);
- середньо-інтерактивні пастки (Medium interactive);
- високо-інтерактивні пастки (High interactive);
- повноцінні ОС (FullOS).

Від рівню інтерактивності залежить рівень правдоподібності пастки, але, є й пряма залежність: окремі пастки, що емулюють роботу БД, або подібні по трудомісткості сервіси, споживають стільки ж, або й більше ресурсів.

Розглянемо це на прикладі SSH пастки. У всіх випадках події з пастки відразу віддаються на консоль керування:

− на низькому рівні інтерактивності SSH пастки система буде пропонувати лише можливість аутентифікації, але увійти в систему не буде можливим; будуть збиратись необхідні клієнтські дані, ключі, логіни та паролі тощо;

− на середньому рівні інтерактивності SSH пастка може бути схожа на restricted shell; в цьому випадку, зловмисник може успішно увійти в систему, але список команд буде обмеженим;

− на високому рівні інтерактивності в пастці будуть присутні різні процеси, домашні папки користувачів, фіктивні дані; окремі пастки, при цьому, обмежують можливості встановлення подальшого SSH з’єднання, аби не стати проміжним хостом для розвитку атаки.

В залежності від рівня інтерактивності, система може бути практично ідентична справжній. Суперечки, наскільки зобов’язана бути інтерактивність пастки – суб’єктивна тема для обговорення: тут варто відштовхуватись від конкретно окремих взятих постановлених завдань. Для звичайного виявлення підійде пастка з низьким рівнем інтерактивності, для дослідження поведінкової моделі хакера та сповільнення горизонтального руху по мережі, найкращим рішенням буде високий рівень інтерактивності.

Класифікація пасток виглядає наступним чином:

− *Файлові ресурси SMB різної степені інтерактивності*: можуть бути як емуляцією низької інтерактивності, де зловмисник може побачити лише список доступних розшарених папок, без можливості зайти на них (скануючі malware, тощо); більш інтерактивні пастки вже підтримують SMB 2.0/3.0, та можуть динамічно змінювати свій вміст, інтегруючись з середовищами для перевірки контенту, що завантажується;

− *FTP/TFTP/SFTP сервера*: як і файлові пастки SMB, можуть бути різної степені інтерактивності, зазвичай вони реалізуються на базі готових FTP чи SSH рішень;

− *Бази даних (SQL, Oracle)*: емуляція баз даних найбільш цікава задача, адже достатньо індивідуальна як для клієнтів та й в баченні постачальника сервісу/продукту;

− *Пастки для Active Directory*: спеціалізовані пастки, що дають змогу визначати такі речі як LLMNR атаки, атаки на Kerberos, Pass the Hash атаки;

− *Сервіси віддаленого доступу VPN (OpenVPN, IPSec, l2tp)*: через пандемію COVID-19 і масового переходу на віддалений режим роботи, такий тип пасток став найбільш пріоритетним і наявний в більшості портфелів пропозицій вендорів;

− *Роутери та інше активне мережеве обладнання*: ці пастки діляться на дві підкатегорії: вебінтерфейс обладнання, який може детально симулювати окремі повторені компоненти інтерфейсу з можливістю конфігурації та перегляду статистики, а також SSH інтерфейс, в тій чи іншій мірі емулюючий інтерфейс реального мережевого обладнання;

− *Вебсервіси*: емуляція справжніх внутрішніх ресурсів, розпочинаючи з форми логіну в Outlook Web Access і закінчуючи можливістю клонування форми входу будь якої внутрішньої самописної системи і кількох типових сторінок;

- *RDP/VNC системи*: пастки, що емулюють різні системи віддаленого керування робочими столами; зазвичай бувають низько-інтерактивними, але ці ж сервіси працюють добре і в FullOS пастках, де забезпечується їх повноцінна реалізація;
- *Принтери та МФУ*: емуляція протоколів друку; просунуті версії йдуть в комплекті відразу з вебінтерфейсами принтера, даючи змогу створити повноцінну емуляцію;
- *Емуляція SCADA та IoT пристроїв*: складна задача з великою кількість пропрієтарних протоколів; існують окремі DT рішення, що спеціалізуються на даних системах;
- *IP-телефони*: окремий клас пасток, що емулює вузли IP-телефонії; телефонувати з використання таких пасток не вийде, а ось отримати web, SSH чи SIP інтерфейс можливо;
- *CI/CD системи*: імітація популярних систем на кшталт Jenkins чи GitLab;
- *iLO/iDRAC/Server Management*: спеціалізовані пастки для емуляції систем управління серверним обладнанням;
- *Поштові пастки*: імітація популярних протоколів SMTP, IMAP, POP3;
- *Емуляція Private Cloud*: спектр емуляції OpenShift/Docker/k8s;
- *Генерація пасток під популярні та актуальні CVE*: Зарання створюються «вразливі» сервіси, аби викликати у зловмисника якомога більший інтерес саме до них.

На цьому перелік пасток не закінчується. Постачальники DT послуг мають в своєму арсеналі спеціальні набори пасток під інші задачі (на приклад, для MacOS).

FullOS пастки використовуються для розгортання систем, емуляція яких не ефективна. Це повноцінні ОС, що перетворені в пастку для DT платформи, і нічим не відрізняється від реальних систем, окрім, наявності детального

протоколювання. Найчастіше це RDP сервери, що спеціалізуються на банківських АРМ.

Для FullOS важлива ідентичність з реальними, в організації, хостами, тому, найчастіше, вони виконуються з Golden Image. Це дає змогу вирішити питання ліцензії такої ОС. При розгортанні до її вмісту добавляються сервіси-кейлогери, монітори журналів подій, проводиться більш детальна конфігурація засобів протоколювання. Окремі постачальники DT послуг дають змогу встановлювати низькорівневий драйвер-rootkit, що максимально приховує присутність стороннього ПЗ на хості, та інтегрується із засобами віртуалізації. Це дає змогу реалізувати додатковий функціонал:

- кейлогери, що знімають всю активність зловмисника;
- автоматичне створювання snapshot'ів та dump'ів пам'яті для подальшого аналізу;
- повернення (до контрольної точки) після атаки чи відвідування зловмисниками;
- збір повноцінної форензики з хоста.

Рівень приховування обирається індивідуально.

3.3.2 Приманки

Аби атакуючий з більшою вірогідністю наткнувся на пастку, його можна до цього підштовхнути: використовувати, так звані, «хлібні крихти» (breadcrumbs).

Приманка – об'єкт, що розміщується на реальній робочій станції, приховано чи не дуже. Пастка виглядає як щось звичайне, типове та привабливе для атакуючого («випадково» забутий файл з паролем, збережена сесія, сторінка браузера, запис в реєстрі, приєднаний share вузол, тощо). Приманка містить посилання та дані для доступу на фіктивний мережевий ресурс.

Зловмисник, який ідентифікував «привабливий» елемент (посилання та авторизаційні дані), звичайно, захоче перевірити що це за сервіс. Після переходу на пастку спрацьовує сигналізація про інцидент.

Види та способи розміщення приманки залежать від її типу: вони можуть розповсюджуватись кількома способами: якщо в складі DT присутні агенти, задача котрих «розкидування» приманок, тоді цей процес можна достатньо ефективно автоматизувати (сервер керування відправляє команду на агент, і той виконує необхідні дії для встановлення пастки) (рис.3.4).

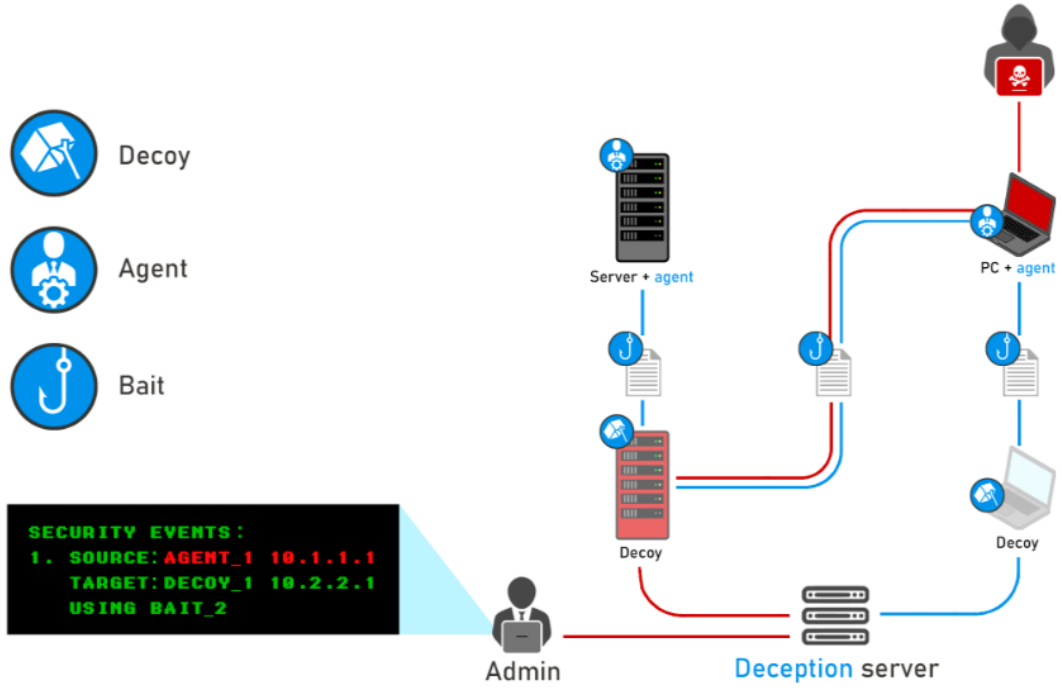

Рисунок 3.4 – Інцидент кібератаки з використанням пастки.

Якщо ж агентів немає, DT-рішення можуть пропонувати готові скрипти, які необхідно виконати на робочих станціях вручну. Даний підхід має очевидні наслідки: при переконфігурації пасток відсутня можливість автоматичного оновлення пастки на робочих станціях, тоді як агенти дають змогу це реалізувати.

Варто, на скільки це можливо, обмежити взаємодію справжніх користувачів ПК з пастками. Однак, надмірні зусилля у приховуванні

фіктивних цілей також шкідливі: якщо вони будуть неідентифіковані для атаки, їхня ефективність прямуватиме до нуля.

Звісно, приманки мають бути вдало мімікріювати під реальні об’єкти: якщо ми поставим пастку у вигляді SSH на комп’ютер бухгалтера – це може викликати підозру у атакуючого, адже це за межами типового корпоративного стандарту по ІБ.

Найчастіше, приманка містить авторизаційні дані для доступу до пастки (логін та пароль, або ключ). Окрім того, можна реалізувати, в межах DT базу фіктивних користувачів, зробивши «легенду» більш глибокою в мімікрії реального середовища роботи.

Популярні види приманок:

- облікові дані користувачів Active Directory;
- файли підключень до віддаленого столу RDP/VNC;
- записи в HOSTS файлі;
- збережені підключення до мережевих ресурсів та принтерів;
- історія команд в BASH та PowerShell;
- історія в браузерах та сховищі паролів в них;
- збережені дані в браузерах;
- мережеві підключення в SSH/FTP клієнтах;
- налаштування популярних програм (реєстр);
- менеджери паролів (на кшталт KeePass);
- сховище паролів (на кшталт MS Vault, LSASS тощо).

Варто відзначити, що приманки мають повністю відповідати інфраструктурі замовника.

3.3.3 Фіктивні користувачі

Поставлене завдання для приманки – надавати дані для авторизації, максимально схожі на справжні з врахуванням особливостей кожного фіктивного користувача (типові нюанси генерації авторизаційних даних, на приклад: перша буква імені, крапка, прізвище латиницею, потреба в поштовій

адресі, адресі домена тощо. Дані складнощі можна вирішити вводячи базу даних фіктивних користувачів мережі.

Інтеграція до DT системи аналізу трафіку дасть можливість розпізнавання в мережевому трафіку наявність даних для авторизації, знаходити в цьому трафіку спільні риси та генерувати користувачів, що схожі на справжніх, по критеріям та правилам, що додатково конфігуруються (рис3.5).

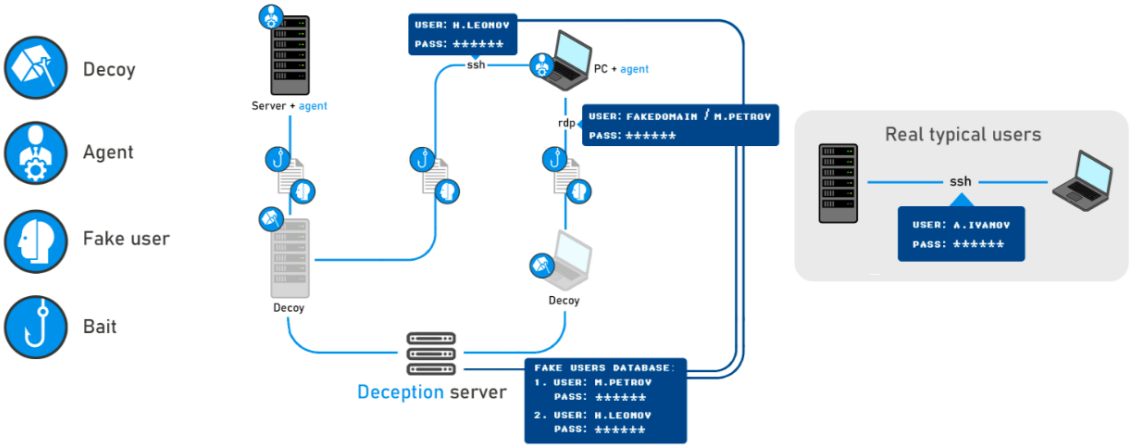

Рисунок 3.5 – Схематичне зображення фіктивної бази користувачів.

Якщо така інтеграція відсутня, хорошим рішенням буде генерація користувачів по заданим, вручну, параметрам. Серед таких правил можуть бути певні словники імен, шаблони для логінів, генерація інтуїтивних паролів (наявність спецсимволів, мінімальна довжина текстових елементів, вибір паролів зі списку широкого вжитку, тощо), вибір фіктивних адрес доменів, поштових серверів.

Даний підхід може згодитись, якщо компанія розгортає DT захищаючи філіали, розміщені в різних регіональних зонах (використання територіальних словників, тощо).

Після формування бази фіктивних користувачів, DT може використовувати її при створенні приманок. Для більшої реалістичності, варто

зробити зв’язки між агентом та фіктивним користувачем так, аби всі приманки, розміщені на цьому агенті, були від імені іншої людини.

3.3.4 Емуляція мережевої взаємодії

Якщо традиційні HP існують самі по собі, без взаємодії з іншими елементами мережі й не лишаючи мережевого сліду, то DT направлена саме на ініціацію та конфронтацію атаки через мережевий слід. Для цього, атакуючому варто «підказати», де шукати пастку, змушуючи його думати, що це реальний сервіс (на приклад, роблячи приманку вразливою до активного сніфінгу, роблячи сервіс «живим» в просторі трафіку).

В цьому можна реалізувати ще одну перевагу DT над HP – можливість активно емулювати мережеву взаємодію. Взаємодіяти можуть будь які точки в мережевій інфраструктури: пастки з приманками, агенти з пастками. Реалізація залежить від конкретного рішення і може включати в себе як взаємодію простими пакетами TCP/UDP, так й передачу даних по окремо взятому протоколу вищого рівня стеку. Конкретика залежить лише від типу пастки: для прикладу, можна «навчити» агента з деякою періодичністю «відвідувати» пастку, яка проводить емуляцію SSH, проходячи авторизацію, і, навіть, виконувати команди.

Важлива деталь: пастка сповіщає про будь які спроби підключення до неї та передає повідомлення про це на сервер, і про підключення симуляційного характеру, також, буде повідомлення. На сервері DT має бути вбудований механізм (трігер), який зможе відрізнити симуляційні та реальні події спрацювання протоколів безпеки один від одного.

Емуляція, в певній мірі, пересікається з ідеєю приманок: наявні мережеві протоколи, які мають на меті передачу авторизованих даних у відкритому вигляді (для прикладу, FTP). Емуляція підключення по даному протоколу буде працювати так само як і приманка: ми провокуємо атакуючого, який прослуховує трафік, логін та пароль для доступу до пастки (рис.3.6).

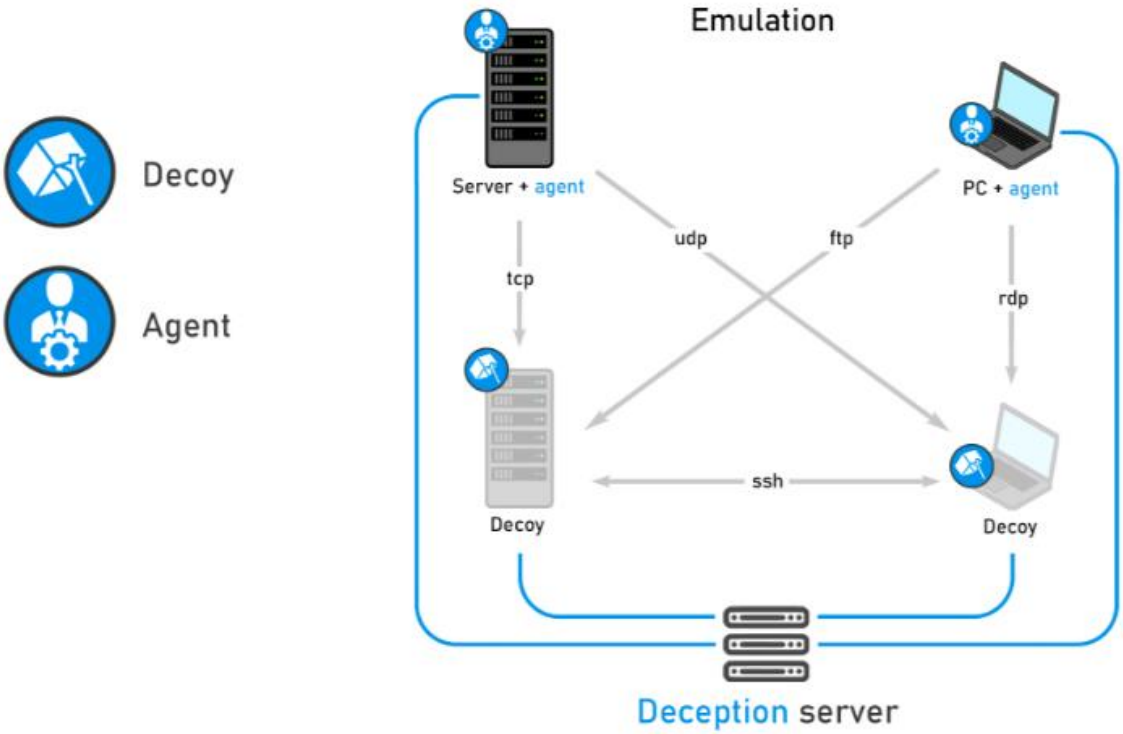

Рисунок 3.6 – Емуляція пастки для слухача трафіку.

Використовувати цю функціональність варто з обережністю, адже хакер може запідозрити провокацію аналізуючи трафік: якщо запити однакові (носять фіктивний характер, часто відбуваються, носять систематичний характер нелюдського походження, мають нетипову довжину тощо). При розробці та конфігурації систем DT необхідно враховувати ці нюанси: вводити рандомізацію, або інші методи маскування та мімікрії.

3.4 Імплементація додаткових компонентів DT

Розглянемо імплементацію двох найбільш вживаних та актуальних компонентів – автоматичного розгортання та збір даних з робочих станцій.

Автоматичне розгортання. Одна із потенційних проблем DT – це трудомісткість первинної конфігурації. Без автоматичного деплоя при встановленні DT в мережу, необхідно вручну визначати список пасток та сервісів емуляції, правильного їх конфігурувати і для кожного пастки, вручну,

створювати та розміщувати приманки. Що є досить трудомістко. Формальна слабка сторона DT. При цьому, відсутність типових рішень, які можна імплементувати «з коробки». В кожній, окремо взятій, організації є свій набір мережевих ресурсів, який є сенс розміщувати у вигляді пасток. Якщо мережа компанії достатньо компактна, то вистачить і одного фахівця для конфігурації. Якщо ж мережа компанії достатньо поскладнена функціональними можливостями та робочими станціями, підмережами – імплементація сервісу DT виглядатиме непомірно-дорогою.

Одна з можливих реалізацій рішення проблеми полягає в інтеграції із системою аналізу трафіку (аналогічно, як й з фіктивними користувачами), система отримує дані про протоколи взаємодій в кожній окремій підмережі. На підґрунті даної аналітики, DT може автоматично встановлювати необхідні пастки в потрібній кількості і, навіть, обновлювати кожен рівень мережі під час додавання нових, реальних, ресурсів. Якщо ж така інтеграція відсутня, проблему можна виправити активним скануванням мережі, отримуючі дані про відкриті порти на реальних машинах, або, пасивно прослуховуючи трафік, де це можливо. Зібрану інформацію DT буде використовувати для автоматичного встановлення пасток.

Останнім розглянутим варіантом швидкої імплементації буде спосіб вибору бажаного списку мережевих сервісів і певної кількості пасток. При цьому налаштування будуть виконані по-шаблону за критеріями конфігурації. Звісно, це спрощує імплементацію, але, з негативних наслідків буде непрозорі можливості та, можлива, невідповідність симуляції реальним процесам мережі (рис.3.7).

Збір даних з робочих станцій. Розглянемо використання сервісу DT для нетривіальних завдань, а саме – збір агентом даних про встановлені на комп’ютерах ПЗ, включаючи версію та дату встановлення. Результати можна порівнювати з базами CVE і вчасно попереджувати про вразливості, базовані на версіях ПЗ.

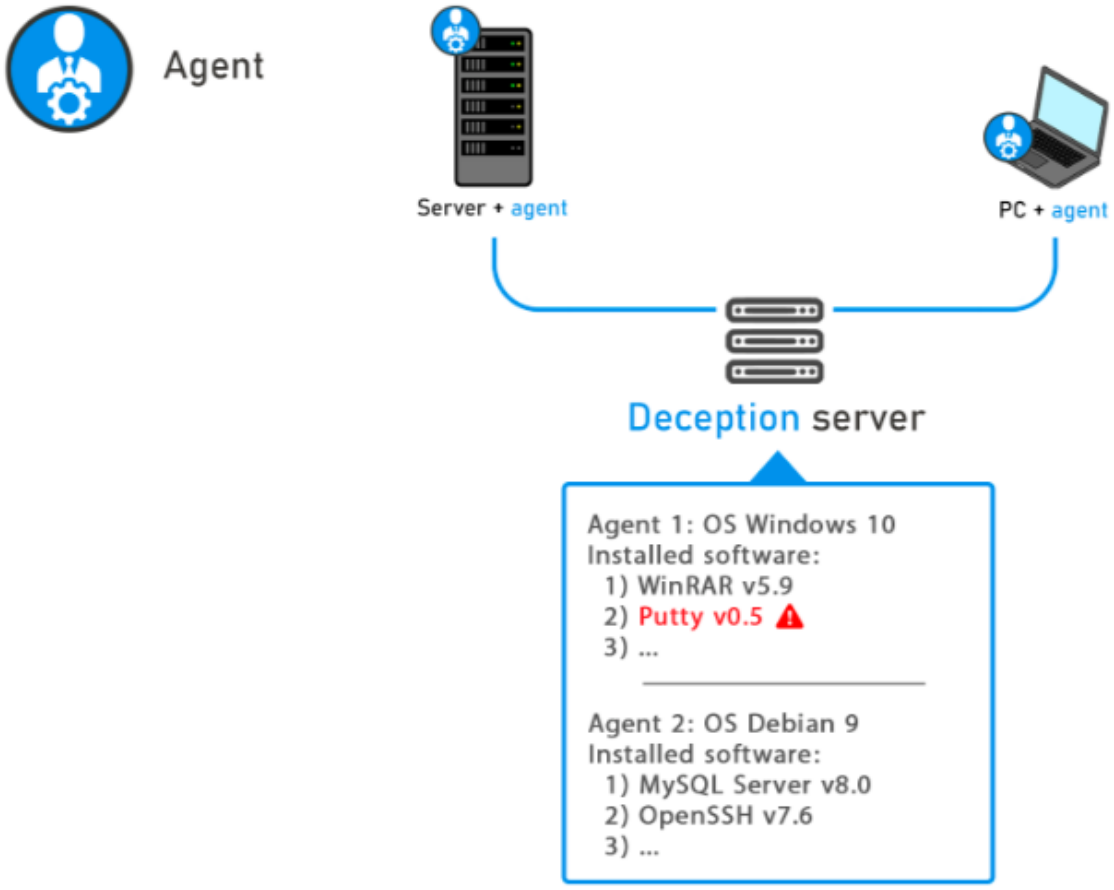

Рисунок 3.7 – Моніторинг версії ПЗ на робочих станціях.

Наступним важливим аспектом цього питання є збір даних для форензики. Під час виявлення пасткою атаки, джерело котрої є робоча станція з агентом, DT співставляє дані від пастки (час, порт джерела) з інформацією, якою володіє агент. Таким чином отримуємо корисні дані про атаку: який процес її ініціював, як він проник на комп'ютер тощо.

Також, агент може збирати різні індикатори компрометації на робочій станції працівників, що дає змогу отримувати повідомлення про активні дії атакуючих в мережі.

Хоч й DT доволі нова технологія, і рішення цього класу на ринку не так давно, але сервісів та програмних продуктів стає все більше, і дані рішення завойовують популярність заслужено. В останньому звіті Gartner Deception була названа одною з найбільш ефективних технологій в області захисту

інформації: за думкою експертів, вона знаходиться на піку і її вихід на плато очікується в горизонті 5-10 років. DT не замінює загальноприйняті стандарти, але їх доповнює, даючи змогу визначити атаки, що обійшли всі інші засоби ІБ.

Окрім того, це достатньо гнучка система, що при інтеграції з іншими засобами ІБ дає змогу реалізувати широкі можливості виявлення атак. В DT є змога надбудовувати ефективні механізми інвентаризації мережевих активів, реагувати на мережеві інциденти тощо. Ефективність системи залежить від її конфігурації, при вдалій – хакер не буде знати, що перед ним фіктивна ціль протягом усього циклу атаки.

Сучасні технології DT вже давно перейшли кордон «мережі HP» і пропонують значно більший функціонал, що постійно розвивається.

3.5 Симуляція атаки в середовищі Deception

Поставлене завдання кожного засобу та сервісу ІБ – попередити проникнення зловмисника, зробивши периметр інфраструктури неприступною фортецею.

Обман в середовищі DT має місце бути за двома сценаріями: створення фіктивних мережевих вузлів з різними сервісами, або розміщення на вузлах даних, які приведуть зловмисника на взаємодію з trap-сервером.

Зупинимось на варіанти Xello Deception, проводячи модель атаки в інфраструктурі з діючою DT в системі.

Вхідними даними є:

− домен з кількома АРМ та серверами;

− зловмисник, що має мережевий доступ до домену та АРМ, але без відомих облікових записів в інфраструктурі;

− розгорнутий Xello Deception: на кінцевих пристроях ОЗ справжні, паролі – фіктивні (з можливістю легкого підбору); обмеження лише зі

справжніми ОЗ (Dexem приманки); окрім діючих ОЗ в створюємо фіктивні (існують локально, відсутні в AD) і відключені ОЗ;

- в AD пара ОЗ/пароль – справжня, пароль стійкий до перебору;
- створені DNS-alias'и, що ведуть на сенсори Xello (trap-сервери);
- створені сервери сенсори Xello (trap-сервери).

Уявімо ситуацію, що зловмисники змогли під'єднати до внутрішньої мережі, через одноплатний комп'ютер з віддаленим керуванням (підкуп, шантаж, людський фактор тощо).

Симуляція атаки складається з послідовність кроків:

Крок 1. Збір інформації

Перший крок зловмисника – розвідка. Від пінгу до сканування. Найчастіше в ході розвідки вдається знайти облікові дані серед коду внутрішніх вебдодатків, в інструкціях, регламентах, log'ах тощо. Це все може містити корисні дані для зловмисника. Ось чому сканування мережевих директорій – пріоритетна задача на початку дій (рис.4.1).

Рисунок 4.1 – Збір інформації зловмисником.

Зловмисник не знайшов мережевих директорій в загальному доступі, підпис SMB увімкнено NTLM-relay заблоковано. Стандартний список портів Unix та Windows хостів.

Вигляд з середовища DT приведено на рисунку 4.2.

Крок 2. Отримання первинного доступу

Зловмисник в пошуках вразливих хостів: підбирає паролі, проводить LLMNR/NBTNS spoofing тощо. Оскільки саме порушення парольної політики

допомагає зловмисникам досягнути поставленої задачі в більшості випадках, то перше, на що варто від нього очікувати – PasswordSpray SMB (рис.4.2).

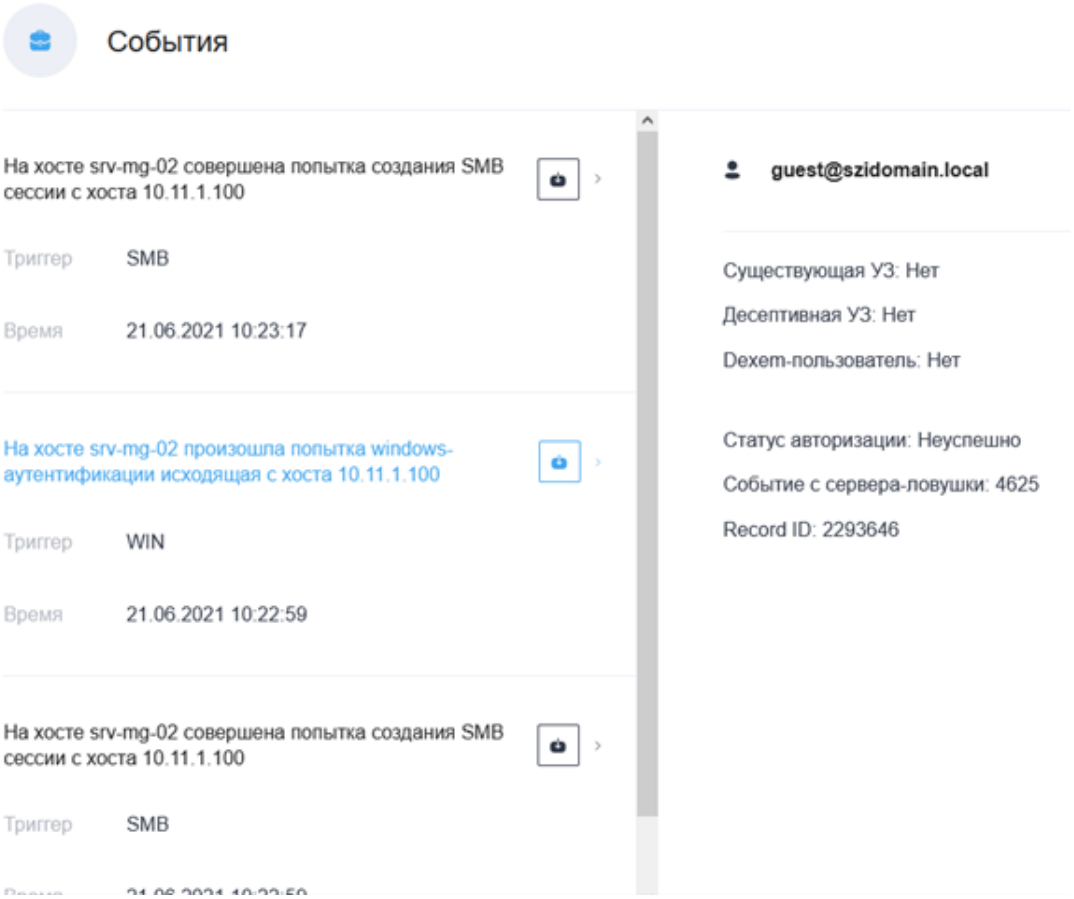

Рисунок 4.2 – Середовище DT сигналізує про спробу створення SMB сесії #1.

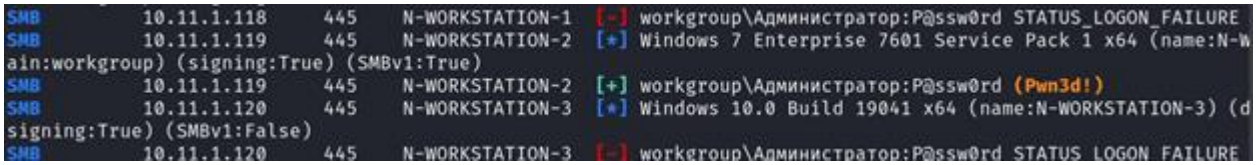

Рисунок 4.3 – Атака PasswordSpray SMB.

Після даної атаки по протоколу SMB, він виявив відсутність зміни стандартного пароля локального адміністратора. Це дало йому змогу отримати точку доторку в мережу, далі – знаходження скомпрометованого АРМ доменний ОЗ.

Вигляд з середовища DT (рис.4.4):

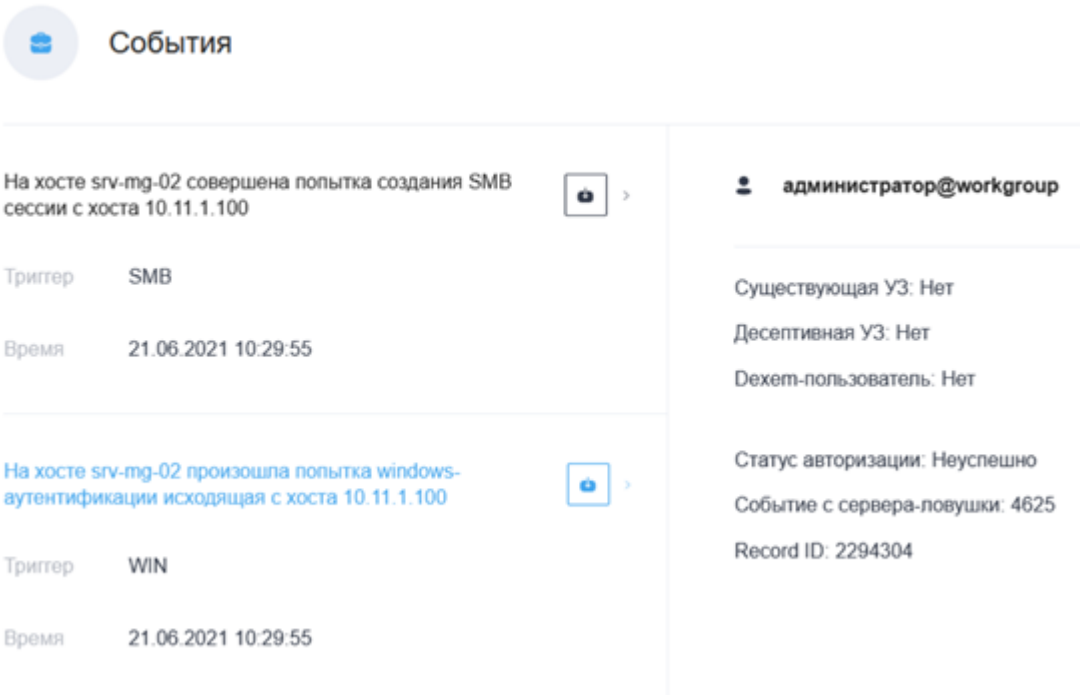

Рисунок 4.4 – Середовище DT сигналізує про спробу створення SMB сесії #2.

Крок 3. Збір додаткової інформації

Один із простих та відносно непомітних способів здобути збережені на АРМ хеши та паролі користувачів – dump процесу lsass.exe та деталізований огляд результату. Враховуючи наявність необхідних прав у зловмисника, виконуються наступні дії (рис.4.5, рис.4.6, рис.4.7, рис.4.8, рис.4.9):

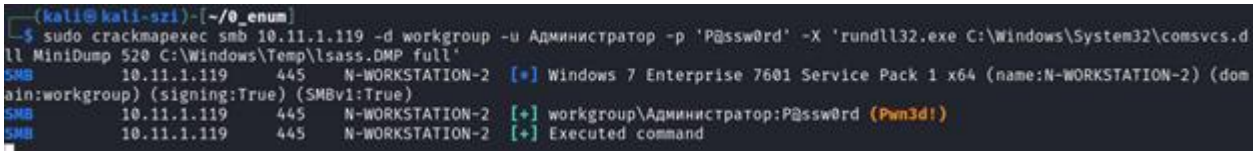

Рисунок 4.5 – Створення dump LSASS за допомогою команди crackmapexec.

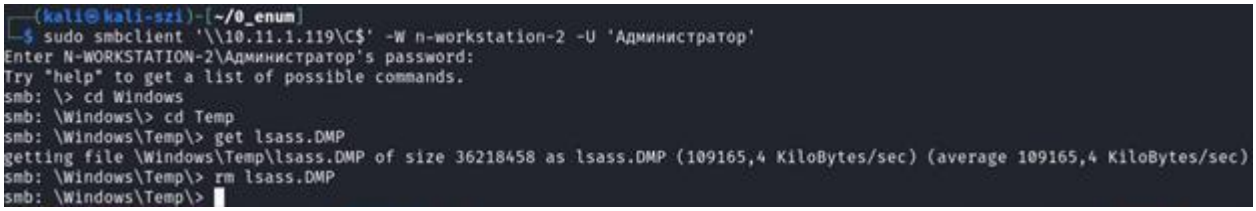

Рисунок 4.6 – Викачування dump’а та видалення його за допомогою smbclient.

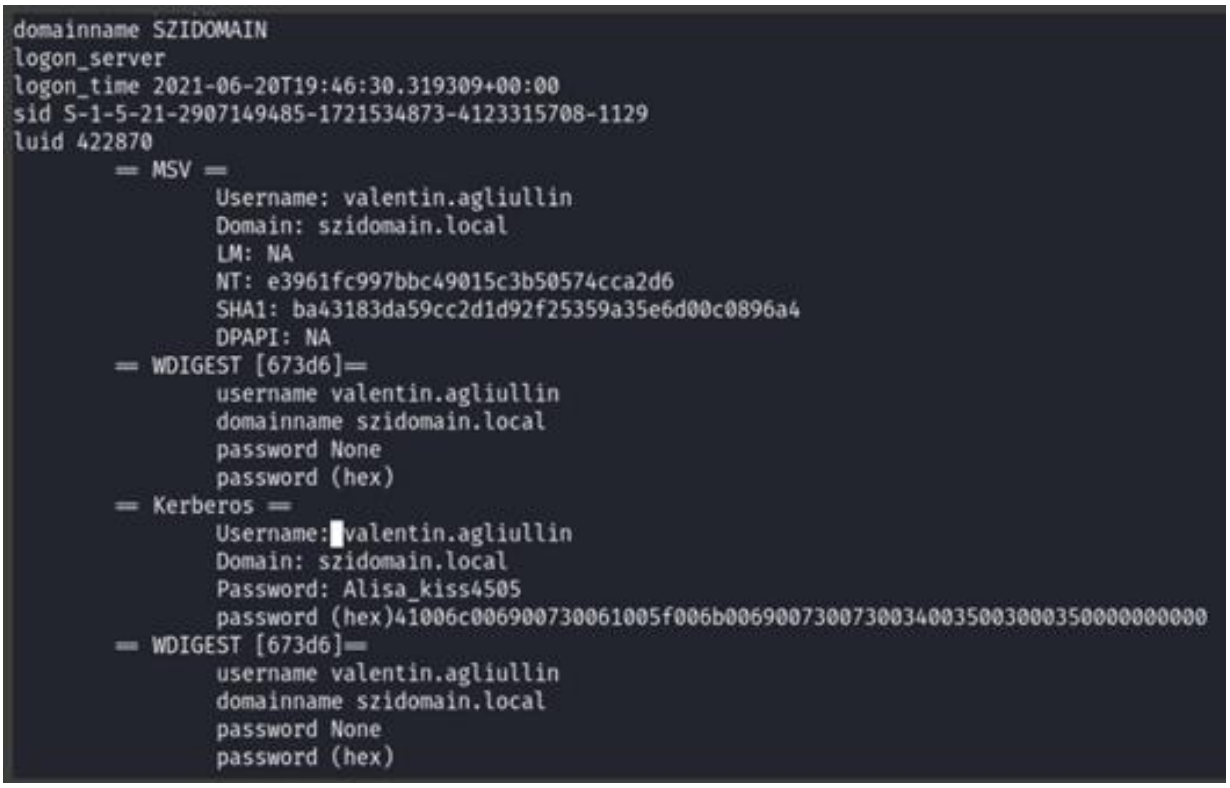

Рисунок 4.7 – Відкриті паролі доменних користувачів за допомогою pypykatz.

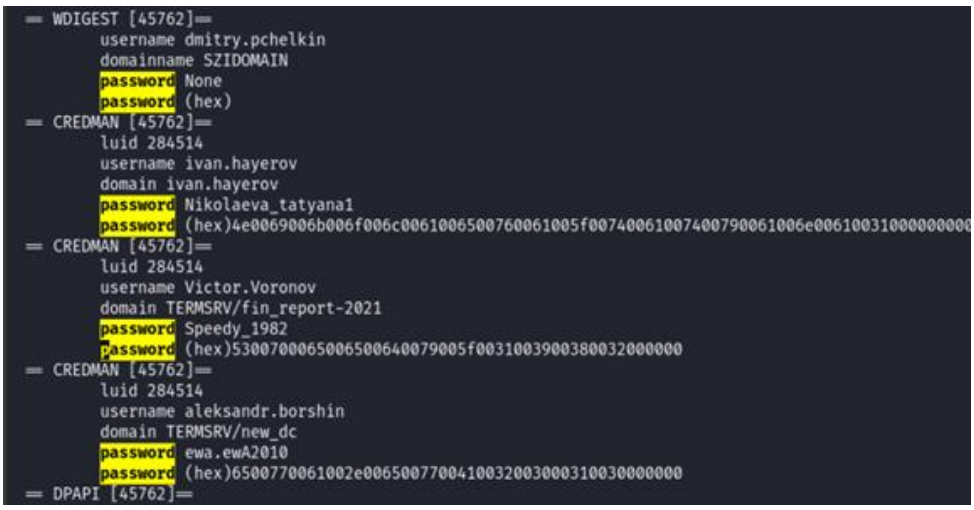

Рисунок 4.8 – Облікові записи користувачів дискредитовано

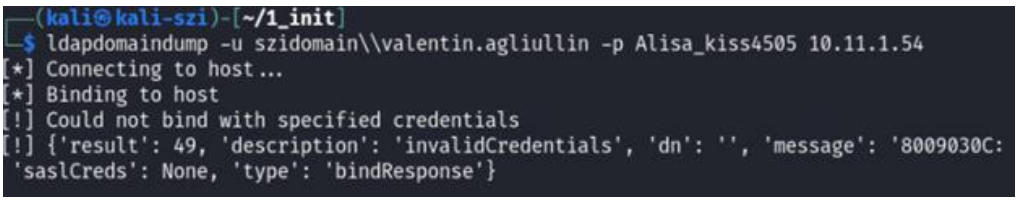

Рисунок 4.9 – Збір інформації, пошук критичних точок.

На цьому етапі у зловмисника виникатимуть колізії: дані облікові записи є приманками, що створені DT системою автоматизовано, на базі реальних даних з домену (технологія Dexem в середовищі Xello Deception), це слугує для примусу хакера витрачати власний час, сигналізуючи про його дії. ОЗ, які дискредитовано хакеру на вузлах – реальні, пароль LSASS – піддається перебору, а справжній пароль даного ОЗ не піддається перебору.

Будь яка спроба запиту за даними ОЗ викликає створення події ІБ. Зазвичай, організація може вимкнути в SIEM кореляцію на некоректну автентифікацію, але в Xell ідентифіковано дані ОЗ як фіктивні і на кожен прецедент логіну з фіктивними даними спрацьовує, також, подія ІБ.

Окрім того, дані ОЗ приманки не відрізняються від справжніх, оскільки це і є типові доменні ОЗ.В цьому і простежується основний функціонал Xello Deception: спотворення справжньої інфраструктури фіктивними даними, аби викликати тривогу кожен раз, як ці дані використовуються. При цьому, це не зменшує ефективність роботи в середовищі для звичайних користувачів.

Вигляд в середовищі DT (рис.4.10):

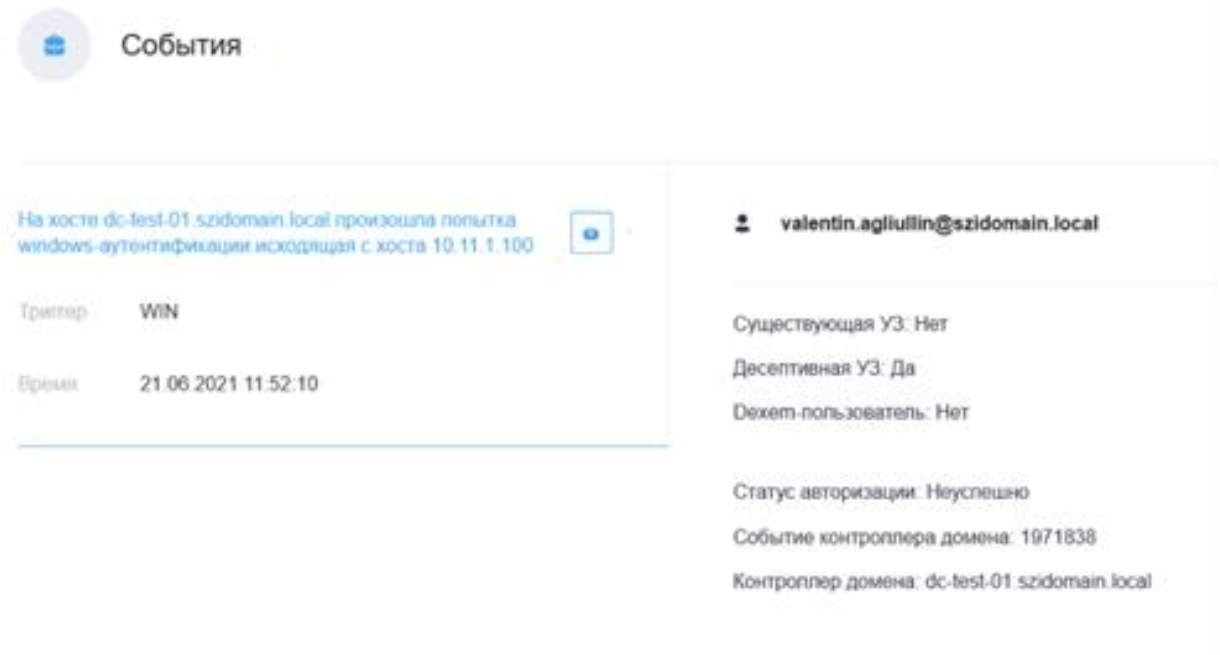

Рисунок 4.10 – Генерація тривоги в DT при використанні фіктивних даних

Зі скріншотів вище, видно, що зловмисник відновив не один ОЗ, а кілька: одна із успішних спроб має успішний наслідок (генерація тривоги, Рис. 18). Серед множини відновлених ОЗ приманок зловмисник знайшов справжню ОЗ. Після відновлення хешу знаходиться пароль Q1w2e3r4 та отримується інформація про домен (рис.4.11).

```
┌──(kali㉿kali-szi)-[~/3_domain_dump]
└─$ ldapdomaindump -u szidomain\\dmitry.pchelkin -p Q1w2e3r4 10.11.1.54
[*] Connecting to host...
[*] Binding to host
[+] Bind OK
[*] Starting domain dump
[+] Domain dump finished

┌──(kali㉿kali-szi)-[~/3_domain_dump]
└─$ cat domain_users.json | jq '.[].attributes.sAMAccountName[0]' -r
dmitry.borisov
boris.aleksandrov
ivan.hayerov
dmitry.sharapov
dmitry.pchelkin
Victor Voronin
Petr Kasatkin
Maria Kapustina
Anna Zayceva
```

Рисунок 4.11 – Отримання інформації зловмисником.

Отримав списки користувачів, зловмисник продовжує збирати більше ОЗ: виконуючи атаку PasswordSpray по протоколу Kerberos. Зловмисник збільшує інтервал запиту з ціллю уникнути блокування (рис.4.12):

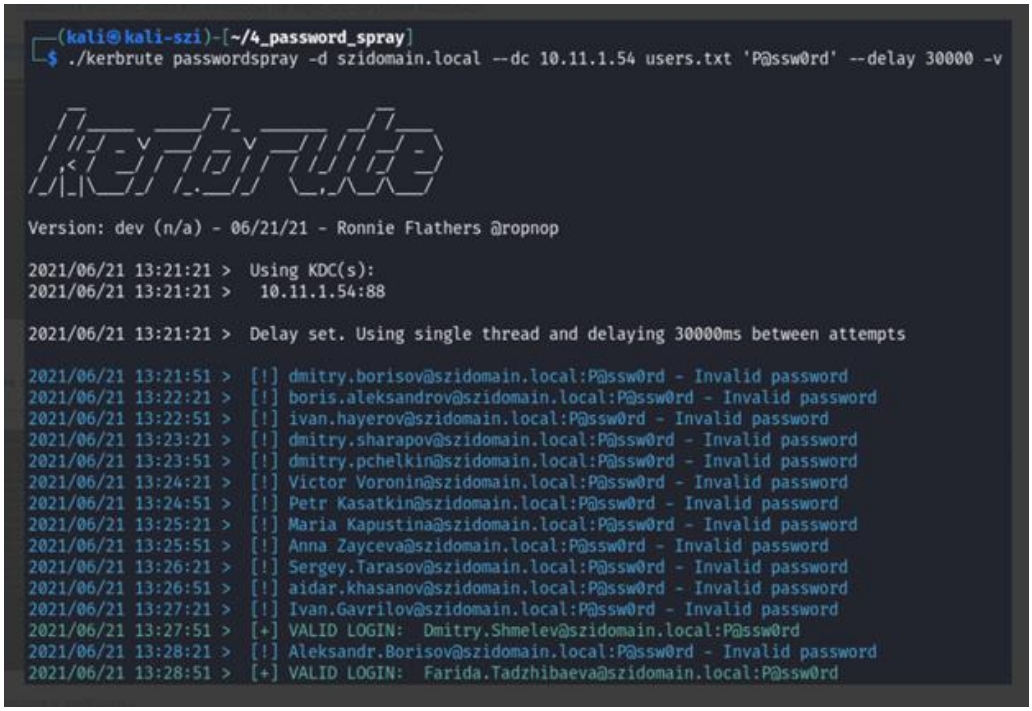

Рисунок 4.12 – Спроба уникнути блокування в системі зловмисником.

Під час даних операцій зловмисником, система DT сигналізує (рис.4.13, рис.4.14, рис.4.15):

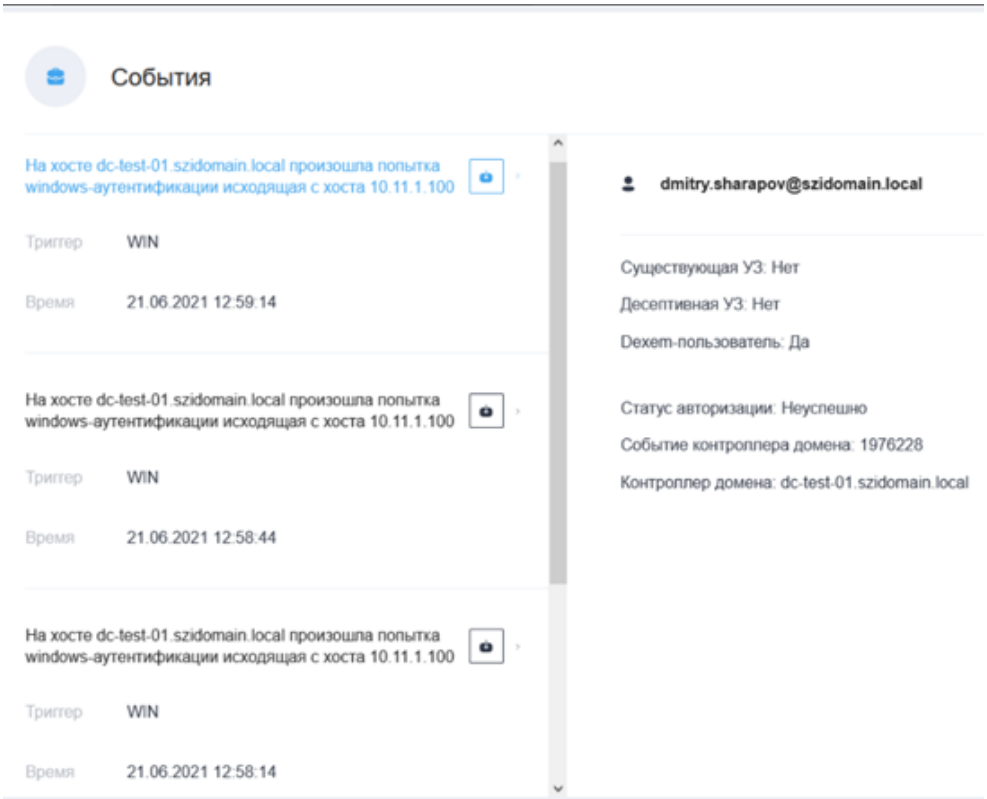

Рисунок 4.13 – Генерація інциденту ідентифікації Dexem приманки.

```
┌──(kali㉿kali-szi)-[~/5_shares]
└─$ sudo smbmap --host-file ./open_445.txt -u dmitry.pchelkin -p Q1w2e3r4 -d szdiomain.local
[!] 445 not open on ....
[!] Authentication error on 10.11.1.55
[!] Authentication error on 10.11.1.56
[!] Authentication error on 10.11.1.57
[!] Authentication error on 10.11.1.129
[!] Authentication error on 10.11.1.120
[!] Authentication error on 10.11.1.130
[!] Authentication error on 10.11.1.119
[!] Authentication error on 10.11.1.118
[!] Authentication error on 10.11.1.168
[!] Authentication error on 10.11.1.220
[!] Authentication error on 10.11.1.58
[!] Authentication error on 10.11.1.219
[!] Authentication error on 10.11.1.172
[+] IP: 10.11.1.54:445  Name: 10.11.1.54
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        SYSVOL                                                  READ ONLY       Logon server share
```

Рисунок 4.14 – Перевірка мережевих файлових сховищ зловмисником.

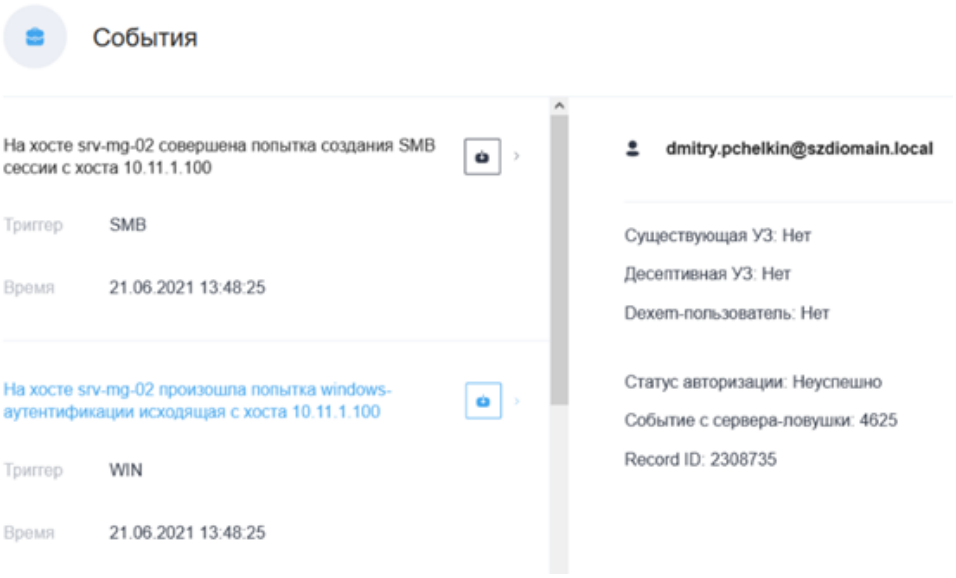

Рисунок 4.15 – Реакція Xello на активацію пастки.

Крок 4. Завершення моделювання атаки

Якщо зловмисник мотивувався атакою на мережеву інфраструктуру отримати дані, то у нього вже є доступ до мережевих файлових сховищ (з отриманим ОЗ) і це фінал (атаки та витоку інформації).

Цілі зловмисника можуть бути індивідуальними: дослідження оточення і переміщення між станціями та серверами, подальші спроби підвищення доступу в домені, шифрування даних тощо.

Результат для служби ІБ: поінформування про наявність зловмисника, про дискредитовані дані, можливість застосування санкцій щодо зловмисника.

Результатом виконання даної симуляції можна вважати виявлення деструктивної діяльності в мережі, практично, з першого пакету. Звісно, це лише спроба уявити психологічну мотивацію зловмисника, але дана симуляція вказує на можливості використання DT систем з приманками в конфігурації Xello. Попри все, частина кроків зловмисників – буде типовою (на приклад, отримання доступу доменних ОЗ).

ВИСНОВКИ

Парадигма кіберзахисту розширюється зі статично пасивного периметру захисту до концептів у вигляді активних систем контратаки з використанням обману, самонавчання яких продовжується протягом усього життєвого циклу системи. Симетрична еволюція хакерських методів та інструментів рухаються пліч - о – пліч в цьому процесі розвитку. У даній роботі було досліджено технології та алгоритми обману для боротьби з кібератаками, представлено поточний стан технології в сфері сервісів DT і суміжних областях.

Аналітичні відомості та бібліографічний аналіз доповнюється посиланнями на відповідні дослідження галузі.

Було зазначено, що DT є вигідним розширенням традиційної IT - безпеки. Акцент було зроблено на категоріях вимог, таких як психологічні, формальні, юридичні та етичні, а також на останніх тенденціях, таких як VMI та галузь безпеки промислової та критичної інфраструктури.

Структуровано етапи імплементації. Проведено симуляцію атаки в середовищі DT на підставі огляду даної системи: виявлення та реакція на хакерську атаку мають часовий показник, який значно мінімізовано при використанні DT – систем, на прикладі DT Xello.

Результат симуляціє слугує критичним аргументом, щодо ефективності імплементації даного класу продуктів. Змога проводити аналіз журналів та мережі в режимі, що наближений до realtime, буде, однозначно, додатковим фактором в користь використання системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wu Sun. The Art of War. Mens sana. Knaur, München, 2001.

2. Kevin David Mitnick and William L. Simon. The art of deception: Controlling the human element of security. Safari Books Online. Wiley, Indianapolis, Ind, 2002.

3. Clifford Stoll. The Cuckoo’s Egg: Tracking a Spy Through the Maze of Computer Espionage. 1989.

4. Bill Cheswick. An evening with berferd in which a cracker is lured, endured, and studied, 1991.

5. Fred Cohen. The deception toolkit home page and mailing list, 1995.

6. Simon Duque Anton, Daniel Fraunholz, and Hans Dieter Schotten. Angriffserken¬nung für industrielle netze innerhalb des projektes iuno. Mobilkommunikation - Technologien und Anwendungen, 2017.

7. Daniel Fraunholz, Simon Duque Anton, and Hans Dieter Schotten. Introducing gamfis: A generic attacker model for information security. International Conference on Software, Telecommunications and Computer Networks, 25, 2017.

8. Daniel Fraunholz, Daniel Krohmer, Simon Duque Anton, and Hans Dieter Schotten. Yaas - on the attribution of honeypot data. International Journal on Cyber Situational Awareness, 2(1), 2017.

9. Barton Whaley. Toward a general theory of deception. Journal of Strategic Studies, 5(1):178-192, 2008.

10. John B. Bell and Barton Whaley. Cheating and deception. Transaction Publ, New Brunswick, repr edition, 1991.

11. James F. Dunnigan and Albert A. Nofi. Victory and deceit: Deception and trickery at war. Writers Club Press, San Jose [etc.], 2nd ed. edition, op. 2001.

12. Charles A. Fowler and Robert F. Nesbit. Tactical deception in air-land warfare. Journal of Electronic Defense, 18(6):37-40, 1995.

13. Neil C. Rowe and Hy S. Rothstein. Two taxonomies of deception for attacks on information systems. Journal of Information Warfare, 3(2), 2004.

14. Charles Fillmore. The case for case. Universals in Linguistic Theory, pages 1-25, 1992.

15. Terry Copeck, Sylvain Delisle, and Stan Szpakowicz. Parsing and case analysis in tanka. Proc. of COLING-92, pages 1008-1012, 1992.

16. John L. Austin and James O. Urmson, editors. How to do things with words: [the William James lectures delivered at Harvard University in 1955]. Harvard Univ. Press, Cambridge, Mass., 2. ed., [repr.] edition, ca. 2009.

17. Frank Stech, Kristin Heckman, Phil Hilliard, and Janice Ballo. Scientometrics of deception, counter-deception, and deception detection in cyber-space. PsychNology Journal, 9(2):79-122, 2011.

18. James D. Monroe. Deception: Theory and practice.

19. U.S. Army. Information operations: Joint publication 3-13, 2014.

20. Jaeeun Shim and Ronald C. Arkin. A taxonomy of robot deception and its benefits in hri. In 2013 IEEE International Conference on Systems, Man, and Cybernetics, pages 2328-2335. IEEE, 2013.

21. Mohammed H. Almeshekah and Eugene H. Spafford. Planning and integrating deception into computer security defenses. The 2014 workshop, pages 127-138, 2014.

22. Mohammed H. Almeshekah. Using Deception to Enhance Security: A Taxonomy, Model, and Novel Uses. Ph.d. thesis, Purdue University, Department of Computer Sciences, 2015.

23. Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains.

24. Jeffrey Pawlick, Edward Colbert, and Quanyan Zhu. A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy. ACM Computing Surveys, 2017.

25. C. Seifert, I. Welch, and P. Komisarczuk. Taxonomy of honeypots: Technical report cs-tr-06/12.

26. F. Pouget, M. Dacier, and H. Debar. Honeypot, honeynet, honeytoken: Terminological issues1.

27. Josef Corey. Advanced honey pot identification and exploitation. Phrack Inc., 11, 2003.

28. Martin C. Libicki. Mesh and the net: Speculations on armed conflict in a time of free silicon. Diane Pub Co, [Place of publication not identified], 2004.

29. David B. Buller and Judee K. Burgoon. Interpersonal deception theory. Communication Theory, 6(3):203-242, 1996.

30. G. Cybenko, A. Giani, and P. Thompson. Cognitive hacking: A battle for the mind. Computer, 35(8):50-56, 2002.

31. Jinwei Cao, Janna M. Crews, Ming Lin, Judee Burgoon, and Jay F. Nunamaker. Designing agent99 trainer: A learner-centered, web-based training system for deception detection. In G. Goos, J. Hartmanis, J. van Leeuwen, Hsinchun Chen, Richard Miranda, Daniel D. Zeng, Chris Demchak, Jenny Schroeder, and Therani Madhusudan, editors, Intelligence and Security Informatics, volume 2665 of Lecture Notes in Computer Science, pages 358-365. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

32. Lorrie Cranor and Simson Garfinkel. Security and Usability: Designing Secure Systems that People Can Use. O'Reilly Media Inc, Sebastopol, 2008.

33. Jim Yuill, Dorothy Denning, and Fred Feer. Psychological vulnerabilities to deception, for use in computer security. DoD Cyber Crime Conference, 2007.

34. Ramsbrock Daniel, Berthier Robin, and Cuckier Michel. Profiling attacker behavior following ssh compromises. International Conference on Dependable Systems and Networks, 2007.

35. Bertrand Sobesto. Empirical Studies based on Empirical Studies based on Honeypots for Characterizing Attackers Behavior. PhD thesis, University of Maryland, Maryland,, 2015.

36. Christian Jordan-Michael Howell. The Restrictive Deterrent Effect of Warning Banners in a Compromised Computer System. Master thesis, University of South Florida, 2011.

37. Daniel Fraunholz, Frederic Pohl, and Hans Dieter Schotten. Towards basic design principles for high- and medium-interaction honeypots. European Conference on Cyber Warfare and Security, 16, 2017.

38. Daniel Kahneman, editor. Judgment under uncertainty: Heuristics and biases. Cambridge Univ. Press, Cambridge, 24. printing edition, 2008.

39. Kun Wang, Miao Du, Sabita Maharjan, and Yanfei Sun. Strategic honeypot game model for distributed denial of service attacks in the smart grid. IEEE Transactions on Smart Grid, page 1, 2017.

40. Daniel Fraunholz and Hans Dieter Schotten. Strategic defense and attack in deception based network security. International Conference on Information Networking, 32, 2018.

41. Augusto de Barros. Dlp and honeytokens, 2007.

42. Lance Spitzner. The honeynet project: Trapping the hackers. IEEE Security and Privacy, 2003.

43. Eugene H. Spafford. More than passive defense, 2011.

44. Neil C. Rowe, John Custy, and Duong Binh. Defending cyberspace with fake honeypots. Journal of Computers, 2(2):25-36, 2007.

45. Samuel Lauren, Sampsa Rauti, and Ville Leppanen. An interface diversified honeypot for malware analysis. In Rami Bahsoon and Rainer Weinreich, editors, Proccedings of the 10th European Conference on Software Architecture Workshops - ECSAW ’16, pages 1-6, New York, New York, USA, 2016. ACM Press.

46. Maya Bercovitch, Meir Renford, Lior Hasson, Asaf Shabtai, Lior Rokach, and Yuval Elovici. Honeygen: an automated honey tokens generator. International Conference on Intelligence and Security Informatics, 2011.

47. Ari Juels and Ronald Rivest. Honeywords: Making password-cracking detectable. Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, pages 145-160, 2013.

48. Martin Lazarov, Jeremiah Onaolapo, and Gianluca Stringhini. Honey sheets: What happens to leaked google spreadsheets? USENIX Workshop on Cyber Security Experimentation and Test, 9, 2016.

49. Frederico Araujo, Kevin W. Hamlen, Sebastian Biedermann, and Stefan Katzen- beisser. From patches to honey-patches. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, Proceedings of the 2014 ACM SIGSAC Conference on Com¬puter and Communications Security - CCS '14, pages 942-953, New York, New York, USA, 2014. ACM Press.

50. Jeffrey Avery and Eugene H. Spafford. Ghost patches: Fake patches for fake vulnerabilities. Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection, pages 399-412, 2017.

51. Daniel Fraunholz and Hans Dieter Schotten. Defending web servers with feints, distraction and obfuscation. International Conference on Computing, Networking and Communications, 2018.

52. Hamed Okhravi, Thomas Hobson, David Bigelow, and William Streilein. Finding focus in the blur of moving-target techniques. IEEE SECURITY & PRIVACY, 12(2):16—26, 2014.

53. Ehab Al-Shaer, Qi Duan, and Jafar Jafarian. Random host mutation for moving target defense. Proceedings of the International Conference on Security and Privacy in Communication Systems, pages 310—327, 2012.

54. Kyungmin Park, Samuel Woo, Daesung Moon, and Hoon Choi. Secure cyber deception architecture and decoy injection to mitigate the insider threat. Symmetry, 10(1):14, 2018.

55. Daniel Fraunholz, Daniel Krohmer, Frederic Pohl, and Hans Dieter Schotten. On the detection and handling of security incidents and perimeter breaches - a modular and flexible honeytoken based framework. Proceedings of the 9th IFIP International Conference on New Technologies, Mobility & Security, 9, 2018.

56. Michae Hoglund and Shawn Bracken. Inoculator and antibody for computer security, 2011.

57. Mark Dowd, John McDonald, and Justin Schuh. The art of software security assessment: Identifying and preventing software vulnerabilities. Addison-Wesley, Upper Saddle River, N.J., 2007.

58. Neil C. Rowe. Deception in defense of computer systems from cyber attack. In Lech Janczewski and Andrew Colarik, editors, Cyber Warfare and Cyber Terrorism, pages 97—104. IGI Global, 2007.

59. Nikos Virvillis, Bart Vanautgaerden, and Oscar Serrano. Changing the game: The art of deceiving sophisticated attackers. International Conference on Cyber Confl ict, 6, 2014.

60. Martin Nawrocki, Matthias Wahlisch, Thomas Schmidt, Christian Keil, and Jochen Schonfelder. A survey on honeypot software and data analysis. CoRR, 2016.

61. Hamid Mohammadzadeh, Roza Honarbakhsh, and Omar Zakaria. A survey on dynamic honeypots. International Journal of Information and Electronics Engineering, (Vol. 2, No. 2), 2012.

62. Wira Zanoramy Ansiry Zakaria and Laiha Mat Kiah. A Review on Artificial Intelligence Techniques for Developing Intelligent Honeypot. PhD thesis, University of Malaya, Faculty of Computer Science and Information Technology, Kuala Lumpur, 2012.

63. Wira Zanoramy Ansiry Zakaria and Laiha Mat Kiah. A review of dynamic and intelligent honeypots. ScienceAsia 39S, 2013.

64. Daniel Fraunholz, Marc Zimmermann, Alexander Hafner, and Hans Dieter Schotten. Data mining in long-term honeypot data. IEEE International Conference on Data Mining series - Workshop on Data Mining for Cyber-Security, 2017.

65. Daniel Fraunholz, Marc Zimmermann, and Hans Dieter Schotten. An adaptive honeypot configuration, deployment and maintenance strategy. International Conference on Advanced Communications Technology. International Conference on Advanced Communications Technology, 19, 2017.

66. Daniel Fraunholz, Daniel Krohmer, Simon Duque Anton, and Hans Dieter Schotten. Investigation of cyber crime conducted by abusing weak or default passwords with a medium interaction honeypot. International Conference On Cyber Security And Protection Of Digital Services, 2017.

67. Simon Duque Anton, Daniel Fraunholz, Christoph Lipps, Frederic Pohl, Marc Zimmermann, and Hans Dieter Schotten. Two decades of scada exploitation: A brief history. Workshop On The Security Of Industrial Control Systems & Of Cyber-Physical Systems, 3, 2017.

68. Venkat Pothamsetty and Matthew Franz. Scada honeynet project: Building honeypots for industrial networks, 2004.

69. N. Provos. A virtual honeypot framework. USENIX Security Symposium, 2004.

70. Digital Bond Inc. Digitalbond virtual plc honeynet, 2007.

71. Lukas Rist, Johnny Vestergaard, Daniel Haslinger, Andrea Pasquale, and John Smith. Conpot ics/scada honeypot, 2015.

72. Samuel Litchfield. HoneyPhy: A physics-aware CPS honeypot framework. PhD thesis, Georgia Tech, 2017.

73. GridPot: Symbolic Cyber-Physical Honeynet Framework. sk4ld, 2015.

74. Owen Redwood, Joshua Lawrence, and Mike Burmester. A symbolic honeynet framework for scada system threat intelligence. In Mason Rice and Sujeet Shenoi, editors, Critical Infrastructure Protection IX, volume 466 of IFIP Advances in Information and Communication Technology, pages 103-118. Springer International Publishing, Cham, 2015.

75. Stipe Kuman, Stjepan Gros, and Miljenko Mikuc. An experiment in using imunes and conpot to emulate honeypot control networks. In 2017 fOth International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pages 1262-1268. IEEE, 2017.

76. Kyle Wilhoit, Hilt, and Stephen. The gaspot experiment: Unexamined perils in using gas-tank-monitoring systems.

77. Michael Winn, Mason Rice, Stephen Dunlap, Juan Lopez, and Barry Mullins. Constructing cost-effective and targetable industrial control system honeypots for production networks. International Journal of Critical Infrastructure Protection, 10:47-58, 2015.

78. Juan David Guarnizo, Amit Tambe, Suman Sankar Bhunia, Martin Ochoa, Nils Ole Tippenhauer, Asaf Shabtai, and Yuval Elovici. Siphon. In Jianying Zhou and Ernesto Damiani, editors, Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security - CPSS '17, pages 57-68, New York, New York, USA, 2017. ACM Press.

79. J. Cao, W. Li, and B. Li. Dipot: A distributed industrial honeypot system. Smart Computing and Communication, 10699, 2018.

80. Dhavy Gantsou and Patrick Sondi. Toward a honeypot solution for proactive security in vehicular ad hoc networks. In James J. Park, Ivan Stojmenovic, Min Choi, and Fatos Xhafa, editors, Future Information Technology, volume 276 of Lecture Notes in Electrical Engineering, pages 145-150. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

81. Paulo Simoes, Jorge Proenca, Tiago Cruz, and Edmundo Monteiro. On the use of honeypots for detecting cyber attacks on industrial control networks. European Conference on Information Warfare and Security, 12, 2013.

82. T. Holczer, M. Felegyhazi, and L. Buttyan. The design and implementation of a plc honeypot for detecting cyber attacks against industrial control systems. Proceedings of International Conference on Computer Security in a Nuclear World: Expert Discussion and Exchange, 2015.

83. Michael Haney and Mauricio Papa. A framework for the design and deployment of a scada honeynet. Cyber and Information Security Research Conference, 9(1):121- 124, 2014.

84. Emmanouil Vasilomanolakis, Shreyas Srinivase, and Max Mühlhäuser. Did you really hack a nuclear power plant? an industrial control mobile honeypot. IEEE Conference on Communications and Network Security, 2015.

85. Kamil Koltys and Robert Gajewski. Shape: A honeypot for electric power sub¬station. Journal of Telecommunications and Information Technologies, 4:37-43, 2015.

86. Charlie Scott. Designing and implementing a honeypot for a scada network. SANS Institute InfoSec Readning Room, 2014.

87. Jules Disso, Kevin Jones, and Steven Bailey. A plausible soluation to scada security: Honeypot systems. International Conference on Broadband, Wireless Computing, Communication and Applications, 8:443-448, 2013.

88. Daniek Buza, Ferenc Juhasz, György Miru, Mark Felegyhazi, and Tamas Holczer. Cryplh: Protecting smart energy systems from targeted attacks with a plc honeypot. International Workshop on Smart Grid Security, pages 181-192, 2014.

89. Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, and Tsutomu Matsumoto. Iotpot: Analysing the rise of iot compromises. USENIX Workshop on Offensive Technologies, 9, 2015.

90. Daniele Antonioli, Anand Agrawal, and Nils Ole Tippenhauer. Towards high¬interaction virtual ics honeypots-in-a-box. In Edgar Weippl, Stefan Katzenbeisser, Mathias Payer, Stefan Mangard, Alvaro Cardenas, and Rakesh B. Bobba, editors, Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy - CPS-SPC ’16, pages 13-22, New York, New York, USA, 2016. ACM Press.

91. Celine Irvene, David Formby, Samuel Litchfield, and Raheem Beyah. Honeybot: A honeypot for robotic systems. Proceedings of the IEEE, 106(1):61-70, 2018.

92. Arthur Jicha, Mark Patton, and Hsinchun Chen. Scada honeypots: An in-depth analysis of conpot. In 2016 IEEE Conference on Intelligence and Security Informatics (ISI), pages 196-198. IEEE, 2016.

93. Alexandru Vlad Serbanescu, Sebastian Obermeier, and Der-Yeuan Yu. Ics threat analysis using a large-scale honeynet. Electronic Workshops in Computing, pages 20-30. BCS Learning &Development Ltd, 2015.

94. Stewart Sentanoe, Benjamin Taubmann, and Hans P. Reiser. Virtual machine introspection based ssh honeypot. In Unknown, editor, Proceedings of the fth Workshop on Security in Highly Connected IT Systems - SHCIS ’17, pages 13-18, New York, New York, USA, 2017. ACM Press.

95. Michael Vrable, Justin Ma, Jay Chen, David Moore, Erik Vandekieft, Alex C. Sno- eren, Geoffrey M. Voelker, and Stefan Savage. Scalability, fidelity, and containment in the potemkin virtual honeyfarm. In Andrew Herbert and Ken Birman, editors, Proceedings of the twentieth ACM symposium on Operating systems principles - SOSP ’05, page 148, New York, New York, USA, 2005. ACM Press.

96. Georgios Portokalidis, Asia Slowinska, and Herbert Bos. Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation. European Conference on Computer Systems, pages 15-27, 2006.

97. The Honeynet Project. Know your tools: Qebek - conceal the monitoring, 2010.

98. Xuxian Jiang, Xinyuan Wang, and Dongyan Xu. Stealthy malware detection through vmm-based "out-of-the-box" semantic view reconstruction. In Peng Ning, Sabrina de Di Capitani Vimercati, and Paul Syverson, editors, Proceedings of the 14th ACM conference on Computer and communications security - CCS ’07, page 128, New York, New York, USA, 2007. ACM Press.

99. Xuxian Jiang and Xinyuan Wang. Out-of-the-box" monitoring of vm-based highinteraction honeypots. Proceedings of the 10th International Conference on Recent Advaneces in Intrusion Detection, 10(1):198-218, 2007.

100. Brian Hay and Kara Nance. Forensics examination of volatile system data using virtual introspection. ACM SIGOPS Operating Systems Review, 42(3):74, 2008.

101. Brendan Dolan-Gavitt, Tim Leek, Michael Zhivich, Jonathon Giffin, and Wenke Lee. Virtuoso: Narrowing the semantic gap in virtual machine introspection. In 2011 IEEE Symposium on Security and Privacy, pages 297-312. IEEE, 2011.

102. Jonas Pfoh, Christian Schneider, and Claudia Eckert. Nitro: Hardware-based system call tracing for virtual machines. Advances in Information and Computer Security, (7038):96-112, 2011.

103. Deepa Srinivasan and Xuxian Jiang. Time-traveling forensic analysis of vm-based high-interaction honeypots. International Conference on Security and Privacy in Communication Systems, 7:209-226, 2011.

104. Sebastian Biedermann, Martin Mink, and Stefan Katzenbeisser. Fast dynamic extracted honeypots in cloud computing. In Ting Yu, Srdjan Capkun, and Seny Kamara, editors, Proceedings of the 2012 ACM Workshop on Cloud computing security workshop - CCSW '12, page 13, New York, New York, USA, 2012. ACM Press.

105. Tamas Lengyel, Justin Neumann, Steve Maresca, Bryan Payne, and Aggelos Ki- ayias. Virtual machine introspection in a hybrid honeypot architecture. Presented as part of the 5th Workshop on Cyber Security Experimentation and Test, 2012.

106. Tamas K. Lengyel, Steve Maresca, Bryan D. Payne, George D. Webster, Sebastian Vogl, and Aggelos Kiayias. Scalability, fidelity and stealth in the drakvuf dynamic malware analysis system. In Charles N. Payne, Kevin Butler, Micah Sherr, and Adam Hahn, editors, Proceedings of the 30th Annual Computer Security Applications Conference on - ACSAC '14, pages 386-395, New York, New York, USA, 2014. ACM Press.

107. Vincent E. Urias, William M.S. Stout, and Caleb Loverro. Computer network deception as a moving target defense. In 2015 International Carnahan Conference on Security Technology (ICCST), pages 1-6. IEEE, 2015.

108. Jiangyong Shi, Yuexiang Yang, Chengye Li, and Xiaolei Wang. Spems: A stealthy and practical execution monitoring system based on vmi. In Zhiqiu Huang, Xingming Sun, Junzhou Luo, and Jian Wang, editors, Cloud Computing and Security, volume 9483 of Lecture Notes in Computer Science, pages 380-389. Springer International Publishing, Cham, 2015.

109. Nazar Tymoshyk, Roman Tymoshyk, Andrian Piskozub, Pavlo Khromchak, Victor Pyvovarov, and Andrij Novak. Monitoring of malefactor's activity in virtualized honeypots on the base of semantic transformation in qemu hypervisor. In 2009 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pages 370-374. IEEE, 2009.

110. Tamas K. Lengyel, Justin Neumann, Steve Maresca, and Aggelos Kiayias. Towards hybrid honeynets via virtual machine introspection and cloning. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Ger¬hard Weikum, Javier Lopez, Xinyi Huang, and Ravi Sandhu, editors, Network and System Security, volume 7873 of Lecture Notes in Computer Science, pages 164-177. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

111. Michael Beham, Marius Vlad, and Hans P. Reiser. Intrusion detection and honeypots in nested virtualization environments. In 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pages 1-6. IEEE, 2013.

112. Alexander Kedrowitsch, Danfeng Yao, Gang Wang, and Kirk Cameron. A first look. In Nicholas J. Multari, Anoop Singhal, and Erin Miller, editors, Proceedings of the 2017 Workshop on Automated Decision Making for Active Cyber Defense - SafeConfig '17, pages 15-22, New York, New York, USA, 2017. ACM Press.

113. N. Krawetz. Anti-honeypot technology. IEEE Security and Privacy, pages 76-79, 2004.

114. X. Fu, W. Yu, D. Cheng, X. Tan, and S. Graham. On recognizing virtual honeypots and countermeasures. International Symposium on Dependable, Autonomic and Secure Computing, 2, 2006.

115. S. Mukkamala, K. Yendrapalli, R. Basnet, M. Shankarapani, and H. Sung. Detec¬tion of virtual environments and low interaction honeypots. Information Assurance and Security Workshop, 2007.

116. Sina Bahram, Xuxian Jiang, Zhi Wang, Junghwan Rhee, and Dongyan Xu. Dksm: Subverting virtual machine introspection for fun and profit. International Symposium on Reliable Distributed Systems, 9:82-91, 2010.

117. Dean Sysman, Itamar Sher, and Gadi Evron. Breaking honeypots for fun and profit. Black hat USA, 2015.

118. Xu Chen, Jon Anderson, Morley Mao, Michael Bailey, and Jose Nazario. Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware. International Conference on Dependable Systems &Networks, pages 177-186, 2008.

119. Joni Uitto, Sampsa Rauti, Samuel Lauren, and Ville Leppânen. A survey on anti-honeypot and anti-introspection methods. Recent Advances in Information Systems and Technologies, pages 125-134, 2017.

120. R. Dahbul, C. Lim, and J. Purnama. Enhancing honeypot deception capability through network service fingerprinting. International Conference on Computing and Applied Informatics, 2016.

121. M. Dornseif, T. Holz, and Christian Klein. Nosebreak - attacking honeynets. Workshop on Information Assurance and Security, pages 123-129, 2004.

122. T. Holz and F. Raynal. Detecting honeypots and other suspicious environments. Workshop on Information Assurance and Security, pages 1-8, 2005.

123. Andrew Morris. Kippo ssh honeypot detector, 2014.

124. J. Corey. Local honeypot identification. Phrack Inc., 11(62), 2003.

125. Olivier Ferrand. How to detect the cuckoo sandbox and hardening it ? EICAR. Annual Conference, 22:131-148, 2013.

126. Osama Hayatle, Amr Youssef, and Hadi Otrok. Dempster-shafer evidence combining for (anti)-honeypot technologies. Information Security Journal, 21(6):306, 2012.

127. Osama Hayatle, Hadi Otrok, and Amr Youssef. A markov decision process model for high interaction honeypots. Information Security Journal, 22(4):159—170, 2013.

128. Ping Wang, Lei Wu, Ryan Cunningham, and Cliff C. Zou. Honeypot detection in advanced botnet attacks. International Journal of Information and Computer Security, 4(1):30, 2010.

129. Charles Costarella, Sam Chung, Barbara Endicoot-Popovsky, and David Dittrich. Hardening honeynets against honeypot-aware botnet attacks. International Conference on Cloud Security and Management, 3:135-150, 2015.

130.M. Husak and M. Vizvary. Reflected attacks abusing honeypots. Communications security, pages 1449-1452, 2013.

131. Fred Cohen, Irwin Marin, Jeanne Sappington, Corbin Stewart, and Eric Thomas. Red teaming experiments with deception technologies, 2001.

132. Daniel Fraunholz, Marc Zimmermann, Simon Duque Anton, Jorg Schneider, and Hans Dieter Schotten. Distributed and highly-scalable wan network attack sensing and sophisticated analysing framework based on honeypot technology. International Conference on Cloud Computing, Data Science & Engineering, 7, 2017.

133. Daniel Fraunholz, Marc Zimmermann, and Hans Dieter Schotten. Towards deployment strategies for deception systems. Advances in Science, Technology and Engineering Systems Journal, Special Issue on Recent Advances in Engineering Systems, 2017.

134. Linda Liu, Kaitlin Mahar, Cimran Virdi, and Helen Zhou. Hack like no one is watching: Using a honeypot to spy on attackers. MIT Computer and Network Security Term Projects, 2016.

135. Omer Zohar, Alex Barbalat, and Raz Elhara. Applying deception mechanisms for detecting for sophisticated cyber attacks: A research paper by topspin security.

136. Harriet Mary Jones. THE RESTRICTIVE DETERRENT EFFECT OF WARNING MESSAGES ON THE BEHAVIOR OF COMPUTER SYSTEM TRESPASSERS. PhD thesis, University of Maryland, 2014.

137. Bertrand Sobesto. Empirical Studies based on Honeypots for Characterizing Attackers Behaviour. Dissertation, University of Maryland, 2015.

138. David Maimon, Alper Mariel, Bertrand Sobesto, and Cuckier Michael. Restrictive deterrent effects of a warning banner in an attacked computer system. 2014.

139. Esmaeil Kheirkhah, Sayyed Mehdi Poustchi Amin, Hediyeh Amir Jahanshahi Sistani, and Haridas Acharya. An experimental study of ssh attacks by using honeypot decoys. Indian Journal of Science and Technology, 6, 2013.

140. Zhenxin Zhan, Maochao Xu, and Shouhuai Xu. Characterizing honeypot-captured cyber attacks: Statistical framework and case study. IEEE Transactions on Information Forensics and Security, 8(11):1775-1789, 2013.

141. Gabriel Salles-Loustau, Robin Berthier, Etienne Collange, Bertrand Sobesto, and Michel Cukier. Characterizing attackers and attacks: An empirical study. Proceedings of the IEEE 17th Pacific Rim International Symposium on Dependable Computing, pages 174-183, 2011.

142. Robin Berthier, Jorge Arjona, and Michel Cukier. Analyzing the process of installing rogue software. International Conference on Dependable Systems and Networks, 39, 2009.

143. Daniel Ramsbrock, Robin Berthier, and Michel Cukier. Profiling attacker behavior following ssh compromises. International Conference on Dependable Systems and Networks, 37, 2007.

144. John Smith. Catching flies: A guide to the various flavors of honeypots. SANS Institute GIAC (GCIA) Gold Certification, 2016.

145. Tomasz Grudziecki, Pawel Jacewicz, Lukasz Juszczyk, Kijewski, Piotr, and Pawel Pawlinki. Proactive Detection of Security Incidents: Honeypots. 2012.

146. Ashish Girdhar and Sanmeet Kaur. Comparative study of different honeypots system. International Journal of Engineering Research and Development, 2(10):23- 27, 2012.

147. Katarzyna Gorzelak, Tomasz Grudziecki, Pawel Jacewicz, Premyslaw Jaroszewski, Lukasz Juszczyk, and Piotr Kijewski. Proactive Detection of Network Security Incidents. 2011.

148. Amit D. Lakhani. Deception Techniques Using Honeypots. PhD thesis, University of London, 2003.

149. Matthew L. Bringer, Christopher A. Chelmecki, and Hiroshi Fujinoki. A survey: Recent advances and future trends in honeypot research. International Journal of Computer Network and Information Security, 4(10):63-75, 2012.

150. Iyatiti Mokube and Michele Adams. Honeypots. In David John and Sandria Kerr, editors, Proceedings of the 45th annual southeast regional conference on - ACM-SE 45, pages 321-326, New York, New York, USA, 2007. ACM Press.

151. Aaron Burstein. Conducting cybersecurity research legally and ethically. Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, 2008.

152. Paul Ohm, Douglas Sicker, and Dirk Grunwald. Legal issues surrounding monitoring during network research. Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pages 141-148, 2007.

153. M. Dornseif, F. C. Gärtner, and T. Holz. Vulnerability assessment using honeypots. Praxis der Informationsverarbeitung und Kommunikation, 2004.

154. Brian Scottberg, Willian Yurcik, and David Doss. Internet honeypots: Protection or entrapment? International Symposium on Technology and Society, 2002.

155. Yogendra Jain and Surabhi Singh. Honeypot based secure network system. International Journal on Computer Science and Engineering, 3(2), 2011.

156. Bradley Schaufenbuel. The legality of honeypots. The Global Voice of Information Security, pages 16-20, 2008.

157. Pavol Sokol. Legal issues of honeynet's generations. International Conference on Electronics, Computers and Artificial Intelligence, 2014.

158. Pavol Sokol, Jakub Misek, and Martin Husak. Honeypots and honeynets: issues of privacy. EURASIP Journal on Information Security, 2017.

159. Daniel Fraunholz, Christoph Lipps, Marc Zimmermann, Simon Duque Anton, and Hans Dieter Schotten. Deception in information security: Legal considerations in the context of german and european law. International Symposium on Foundations & Practice of Security, 10, 2017.

160. Roland Campbell. The Legal and Ethical Issues of Deploying Honeypots. Bachelor thesis, University of South Africa, 2014.

161. M. J. Warren and W. Hutchinson. Australian hackers and ethics. Australasian Journal of Information Systems, 10(2), 2003.

162. Neil C. Rowe and Julian Rrushi. Introduction to Cyberdeception. Springer International Publishing, Cham, 2016.

163. Jerome Radcliffe. Cyberlaw 101: A primer on us laws related to honeypot deployments. Information Security Reading Room, 2007.

164. Bradley Rubin and Donald Cheung. Computer security education and research: Handle with care. IEEE SECURITY & PRIVACY, 4(6):56-59, 2006.

165. Maria Karyda and Lilian Mitrou. Internet forensics: Legal and technical issues. Second International Workshop on Digital Forensics and Incident Analysis, 2007.

166. Aline Belloni, Alain Berger, Oliver Boissier, Gregory Bonnet, Gauvain Bourgne, Pierre-Antoine Chardel, Jean-Pierre Cotton, Nicolas Evreux, Jean-Gabriel Ganascia, Philippe Jaillon, Bruno Mermet, Gauthir Picard, Bernard Rever, Simon, Gaele, de Swartem Thibault, Catherine Tessier, Francois Vexler, Robert Voyer, and Antoine Zimmermann. Dealing with ethical conflicts in autonomous agents and multi-agent systems. Proceedings of the 2015 AAAI workshop on artificial intelligence and ethics, pages 21-27, 2015.

167. Kara L. Nance and Daniel Ryan. Legal aspects of digital forensics: A research agenda. Proceedings of the 44th Hawaii International Conference on System Sciences, 2011.

168. T. Holz, F. Freiling, and M. Dornseif. Ermittlung von Verwundbarkeiten mit elektronischen ködern. Detection of intrusions and malware & vulnerability assessment, 2004.

169. German Federal Office of Information Security. It base line protection.

170. Changwook-Park and Y. -g. Kim, "Deception Tree Model for Cyber Operation," 2019 International Conference on Platform Technology and Service (PlatCon), 2019, pp. 1-4, doi: 10.1109/PlatCon.2019.8669410.

171. S. Venkatesh, R. Ramachandra and P. Bours, "Robust Algorithm for Multimodal Deception Detection," 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 2019, pp. 534-537, doi: 10.1109/MIPR.2019.00108.

172. J. Liu et al., "Deception Maze: A Stackelberg Game-Theoretic Defense Mechanism for Intranet Threats," ICC 2021 - IEEE International Conference on Communications, 2021, pp. 1-6, doi: 10.1109/ICC42927.2021.9500765.

173. P. Aggarwal, C. Gonzalez and V. Dutt, "Modeling the effects of amount and timing of deception in simulated network scenarios," 2017 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA), 2017, pp. 1-7, doi: 10.1109/CyberSA.2017.8073405.

174. X. Liu, L. Li, Z. Ma, X. Lin and J. Cao, "Design of APT Attack Defense System Based on Dynamic Deception," 2019 IEEE 5th International Conference on Computer and Communications (ICCC), 2019, pp. 1655-1659, doi: 10.1109/ICCC47050.2019.9064206.

175. Zhenggang Hu and Yueming Lu, "A method based on MD5 and time for preventing deception in electronic commerce," International Conference on Cyberspace Technology (CCT 2014), 2014, pp. 1-3, doi: 10.1049/cp.2014.1289.