УДК 004.056

В. А. Гарнага
Д. О. Немировська

ПОРІВНЯННЯ HONEYPOT-СИСТЕМ ДЛЯ ДОСЛІДЖЕННЯ МЕТОДІВ КОМПРОМЕТАЦІЇ IoT ПРИСТРОЇВ

Вінницький національний технічний університет

Анотація

У роботі проведено порівняльний аналіз популярних honeypot-систем, що застосовуються для дослідження методів компрометації пристроїв Інтернету речей. Розглянуто архітектурні особливості, переваги та недоліки Cowrie, IoTPOT, HoneyThing, Dionaea, T-Pot та ThingPot. На підставі аналізу виявлено спільні обмеження існуючих рішень та визначено напрямки їх подолання.

Ключові слова: інтернет речей, honeypot, кібербезпека, методи компрометації, IoT-загрози, порівняльний аналіз, Cowrie, IoTPOT, HoneyThing, Dionaea, T-Pot, ThingPot.

Abstract

The paper presents a comparative analysis of popular honeypot systems used to study methods of compromising Internet of Things devices. Architectural features, advantages and disadvantages of Cowrie, IoTPOT, HoneyThing, Dionaea, T-Pot and ThingPot are examined. Common limitations of existing solutions are identified and directions for their improvement are proposed.

Keywords: Internet of Things, honeypot, cybersecurity, compromise methods, IoT threats, comparative analysis, Cowrie, IoTPOT, HoneyThing, Dionaea, T-Pot, ThingPot.

Вступ

Стрімке зростання кількості підключених IoT-пристроїв супроводжується пропорційним розширенням поверхні кібератак. За прогнозами аналітиків, до 2030 року їх кількість у світі перевищить 30 мільярдів [1]. Показовим прикладом масштабу пов'язаних загроз є ботнет Mirai, який компрометував велику кількість IoT-пристроїв через незахищений Telnet і спричинив глобальні DDoS-атаки безпрецедентного масштабу [2].

Honeypot – програмна система, що імітує поведінку реального пристрою або сервісу з метою приваблення та документування дій зловмисника – є одним із визнаних інструментів дослідження тактик і технік атак. За рівнем взаємодії з атакуючим такі системи прийнято поділяти на три категорії:

- низького рівня – імітують лише мережеві сервіси без надання оболонки;
- середнього – емулюють поведінку пристрою та базові команди ОС;
- високого – надають повноцінне реальне або емульоване середовище.

Вибір конкретного інструменту суттєво впливає на повноту та репрезентативність зібраних даних. Метою цієї роботи є систематичний аналіз наявних рішень та виявлення їх обмежень у контексті дослідження IoT-загроз.

Основна частина

Існує значна кількість honeypot-інструментів різного ступеня спеціалізації – від універсальних мережевих пасток до вузькоорієнтованих систем для конкретних протоколів чи класів пристроїв. Для порівняльного аналізу відібрано шість систем, що різняться за архітектурним підходом, охопленням протоколів та рівнем взаємодії

Cowrie – honeypot середнього та високого рівня взаємодії для SSH і Telnet, розроблений як наступник проєкту Kippo [3]. Система підтримує два режими роботи: режим емуляції оболонки, де реалізовано емульовану файлову структуру, що відтворює вигляд Debian-інсталяції та режим проксі, що перенаправляє трафік на реальний бекенд-хост для автентичнішої взаємодії. Серед технічних особливостей – детальне журналювання у форматі JSON, захоплення файлів, завантажених атакуючим через wget, curl, SFTP та scp, а також підтримка LLM-режиму для генерування динамічних відповідей. Розгортання здійснюється через Docker, git clone або pip. Практичні результати добре задокументовані в науковій літературі: 30-денне розгортання Cowrie дозволило виявити переважання SSH-атак над Telnet, географічну концентрацію джерел та характерні патерни перебору паролів [4]. Разом із тим

система має задокументовані недоліки. Зокрема, Cowrie схильний до fingerprinting через особливості використовуваного SSH-стеку та типових конфігурацій за замовчуванням, що дозволяє досвідченим зловмисникам ідентифікувати honeypot [5]. Дослідження також показали, що переважна більшість розгортань здійснюється саме з налаштуваннями за замовчуванням, відомими кіберзлочинцям [6]. Cowrie орієнтований виключно на SSH та Telnet і не охоплює прикладних протоколів, характерних для IoT-середовища.

IoTPOT розроблено командою дослідників Йокогамського національного університету спільно з Національним інститутом інформаційних та комунікаційних технологій Японії (NICT) [7]. Система реалізує двоступеневу архітектуру: Frontend Responder взаємодіє з бекенд-пісочницею IoTBOX, що запускає емульовані середовища для CPU-архітектур MIPS, MIPSEL, ARM, PPC, SPARC, MIPS64, sh4, x86 на базі QEMU та OpenWRT. Компонент Profiler автоматично збирає банери реальних вразливих IoT-пристроїв, динамічно навчаючи систему їх імітувати без ручного налаштування – саме це відрізняє IoTPOT від решти розглянутих рішень. За 81 день роботи зафіксовано 481 521 спробу завантаження malware-бінарників з 79 935 унікальних IP-адрес та ідентифіковано щонайменше 5 сімей шкідливого програмного забезпечення [8]. Попри концептуальну значущість, IoTPOT орієнтований виключно на Telnet і не демонструє активної підтримки. Розгортання вимагає конфігурування QEMU та OpenWRT, що суттєво підвищує поріг входу. Підтримки прикладних IoT-протоколів не передбачено.

HoneyThing розроблено у рамках Google Summer of Code проєкту Honeynet та результати опубліковані у KSII Transactions on Internet and Information Systems [9]. Це єдиний серед розглянутих систем honeypot, орієнтований на протокол TR-069 (CWMP) – стандарт дистанційного управління пристроями клієнтського приміщення (CPE), що використовується інтернет-провайдерами для конфігурування роутерів та модемів. Система складається з двох компонентів: реалізації клієнтської частини TR-069 та вебінтерфейсу управління модемом. HoneyThing емулює CVE-вразливості RomPager (Misfortune Cookie, Rom-0), підтримує основні TR-069 CPE-команди та веде роздільне журналювання HTTP- і CWMP-взаємодій. Додатково розроблено скрипт для Bro IDS. Практичне розгортання підтвердило ефективність у виявленні реальних атак на CPE-пристрої [9]. Обмеженням системи є вузька спеціалізація: підтримується виключно TR-069, публічний проєкт давно не оновлюється, а кодова база орієнтована на Python 2.7-era стек. Docker-образ відсутній.

Dionaea – низького рівня взаємодії honeypot, розроблений командою DinoTools як наступник проєкту Nepenthes [10]. Реалізований на Python із застосуванням libemu для виявлення shellcode, підтримує IPv6 та TLS. Офіційна документація містить модулі для широкого спектру протоколів: SMB, FTP, HTTP, TFTP, MSSQL, MySQL, SIP, UPnP, MQTT та ін. Система орієнтована передусім на захоплення копій шкідливих файлів і підтримує інтеграцію з VirusTotal. Практична цінність підтверджена: SMB-стек Dionaea виявився ефективним під час глобальної епідемії WannaCry у 2017 році [11]. Разом із тим щодо MQTT-реалізації існують спеціалізовані сканери, що ідентифікують Dionaea за нестандартною поведінкою пакетів [12]. Dionaea не емулює конкретних IoT-пристроїв, обмежуючись абстрактними мережевими сервісами.

T-Pot – платформа-агрегатор від Deutsche Telekom Security, що об'єднує понад 20 honeypot-систем у єдиному централізованому середовищі [13]. Платформа використовує Docker Compose для одночасного запуску компонентів (Cowrie, Dionaea, Conpot, Heralding, Medpot та ін.) та надає повний Elastic Stack із готовими Kibana-дашбордами й анімованою картою атак у реальному часі. Актуальна версія 24.04.1 підтримується, зокрема, на Alma Linux, Debian, Fedora, openSUSE, Rocky Linux, Ubuntu та Raspberry Pi [14]. Серед найновіших компонентів – LLM-based honeypot Beelzebub (SSH) та Galah (HTTP), що генерують динамічні відповіді на базі великих мовних моделей. T-Pot є найповнішим готовим рішенням серед розглянутих, однак потребує щонайменше 8-16 ГБ оперативної пам'яті та 128 ГБ дискового простору [13], що суттєво обмежує варіанти розгортання. Усі IoT-специфічні можливості успадковуються від вбудованих компонентів зі всіма притаманними їм обмеженнями.

ThingPot розроблено у Делфтському технічному університеті [15]. Система є першим honeypot, що орієнтований не лише на окремі мережеві протоколи, а на емуляцію повноцінної IoT-платформи з усіма її компонентами: бекенд-API, клієнтські застосунки та самі пристрої. ThingPot класифікується як honeypot середнього або гібридного рівня взаємодії: модулі XMPP та MQTT реалізовані як компоненти високої взаємодії, тоді як емуляція пристроїв здійснюється через REST API на рівні низької взаємодії. Proof-of-Concept реалізовано як імітацію системи розумного освітлення Philips Hue з використанням XMPP та RESTful API на базі Django. За 1,5 місяці розгортання виявлено п'ять типів атак проти смарт-пристроїв, зокрема спроби отримання контролю над пристроями та використання мережі TOR для

маскування джерела [15]. Концептуальна новизна ThingPot полягає у фокусі на IoT-платформі як цілісній системі, а не на окремому протоколі чи сервісі. Водночас Proof-of-Concept прив'язаний до єдиного типу пристрою Philips Hue, що суттєво обмежує охоплення різних класів атакуючих. Активної сучасної підтримки проєкту не простежується, Docker-образ відсутній.

Для більш наочного порівняння honeypot-систем представлено таблицю 1, в якій підсумовуються головні аспекти, особливості та характеристики досліджуваних засобів.

Таблиця 1 – Зведена порівняльна таблиця honeypot-систем

Критерій	Cowrie	IoTPOT	HoneyThing	Dionaea	T-Pot	ThingPot
Рівень взаємодії	Medium / High	Low + High (IoTBOX)	Low	Low	Mixed (20+ honeypots і допоміжних інструментів)	Medium / Hybrid
Підтримувані протоколи	SSH, Telnet	Telnet	TR-069, HTTP	SMB, FTP, HTTP, MQTT, SIP, TFTP, UPnP та ін.	Залежить від компонентів	REST API, XMPP, MQTT
Захоплення шкідливого ПЗ	Так	Так	Відсутнє	Так	Так	Відсутнє
Веб-інтерфейс аналізу	Відсутній	Відсутній	Обмежений (імітація веб-інтерфейсу модема)	Відсутній	Повноцінний (Kibana + карта атак)	Відсутній
Емуляція IoT-пристрою	Абстрактна Unix-оболонка	Динамічні профілі реальних пристроїв	Профіль модему/роутера	Абстрактні мережеві сервіси	Абстрактні сервіси різних класів	Пристрій Philips Hue
Docker-підтримка	Так	Відсутня	Відсутня	Так	Так	Відсутня
Активна підтримка	Так	Публічно не простежується	Публічно давно не оновлюється	Так	Так	Активної сучасної підтримки не простежується
Системні вимоги	Низькі	Середні (QEMU + OpenWRT)	Низькі	Середні	Високі (8–16 ГБ RAM, 128 ГБ disk)	Середні

Наведена таблиця ілюструє, що жодна з розглянутих систем не забезпечує комплексного покриття за всіма критеріями одночасно: системи з широким охопленням протоколів поступаються у специфічності емуляції IoT-пристроїв, тоді як вузькоспеціалізовані рішення, попри свою точність у певному векторі, позбавлені універсальності та здебільшого не мають активної підтримки. Платформа T-Pot є найближчою до комплексного рішення, однак орієнтована радше на розгортання у корпоративному середовищі, ніж на гнучке дослідницьке використання.

Висновок

Проведений аналіз засвідчує, що кожна з розглянутих систем вирішує лише вузький клас задач і має суттєві обмеження: Cowrie та IoTPOT орієнтовані на SSH/Telnet і не охоплюють прикладних IoT-протоколів; HoneyThing вузько спеціалізована на TR-069 і публічно не оновлюється; Dionaea, попри широкий набір протоколів, не емулює реальних IoT-пристроїв, а її MQTT-реалізація є детектованою спеціалізованими сканерами [12]; T-Pot є найповнішим рішенням, але непридатним для легкого розгортання через значні апаратні вимоги; ThingPot, попри концептуальну новизну у підході до емуляції IoT-платформи, обмежений одним типом пристрою та не має активної сучасної підтримки.

На рівні спільних недоліків можна виділити три ключові проблеми. По-перше, жодна з розглянутих систем не підтримує емуляцію профілів різних класів IoT-пристроїв – камер, термостатів, промислових шлюзів тощо, що обмежує реалістичність приваблення цільового трафіку атакуючих. По-друге, більшість систем позбавлена повноцінних вбудованих інструментів візуалізації та аналізу зібраних даних, покладаючись або на зовнішні рішення , або не надаючи таких засобів взагалі. По-третє, охоплення сучасних прикладних протоколів IoT-середовища залишається фрагментарним: підтримка MQTT наявна лише у Dionaea та ThingPot, проте в обох випадках є або детектованою, або реалізованою в рамках проєктів без активної підтримки, а CoAP відсутній у жодному з широко використовуваних рішень. Сукупність зазначених обмежень підтверджує актуальність подальших досліджень у напрямку розроблення honeypot-рішень, орієнтованих на сучасне IoT-середовище.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Немировська Д. О., Гарнага В. А. Безпека пристроїв Інтернету речей // Матеріали наукової конференції ВНТУ. – Вінниця, 2024.
2. Antonakakis M. et al. Understanding the Mirai Botnet // Proceedings of the 26th USENIX Security Symposium. – 2017. – С. 1093–1110.
3. Cowrie SSH/Telnet Honeypot. GitHub. URL: https://github.com/cowrie/cowrie (дата звернення: 10.03.2025).
4. The Application of Cowrie Honeypot to Analyze Attacks on SSH and Telnet Protocols // IEEE 2nd International Conference on Electrical Engineering, Computer and Information Technology (ICEECIT). – 2024.
5. Buczak A. et al. Improvement of Cowrie Honeypot Interaction and Deception Capabilities // IEEE Xplore. – 2025.
6. Cabral W. Z. et al. Advanced Cowrie Configuration to Increase Honeypot Deceptiveness // 36th IFIP International Conference on ICT Systems Security and Privacy Protection. – Oslo, 2021.
7. Pa Y. M. P. et al. IoTPOT: Analysing the Rise of IoT Compromises // 9th USENIX Workshop on Offensive Technologies (WOOT 15). – Washington D.C., 2015.
8. Pa Y. M. P. et al. IoTPOT: A Novel Honeypot for Revealing Current IoT Threats // Journal of Information Processing. – 2016. – Т. 24, № 3.
9. Erdem Ö., Pektaş A., Kara M. HoneyThing: A New Honeypot Design for CPE Devices // KSII Transactions on Internet and Information Systems. – 2018. – Т. 12, № 9.
10. DinoTools. Dionaea Honeypot Documentation. URL: https://dionaea.readthedocs.io/ (дата звернення: 10.03.2025).
11. Dionaea Honeypot: From Conficker to WannaCry / SambaCry // The Honeynet Project. – 2017. URL: https://www.honeynet.org/2017/05/30/dionaea-honeypot-from-conficker-to-wannacry-sambacry-cve-2017-7494/ (дата звернення: 10.03.2025).
12. Dionaea MQTT Honeypot Detection Scanner // S4E Security. URL: https://s4e.io/tools/dionaea-mqtt-honeypot-detection-scanner (дата звернення: 10.03.2025).
13. Telekom Security. T-Pot: The All In One Multi Honeypot Platform. URL: https://github.com/telekom-security/tpotce (дата звернення: 10.03.2025).
14. Telekom Security. T-Pot Version 24.04.1 Released. URL: https://github.security.telekom.com/2024/04/honeypot-tpot-24.04-released.html (дата звернення: 10.03.2025).
15. Wang M., Santillan J., Kuipers F. ThingPot: an Interactive Internet-of-Things Honeypot // arXiv:1807.04114. – 2018.

Гарнага Володимир Анатолійович – к.т.н., доцент кафедри захисту інформації, факультету інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: garnaga.volodymyr@vntu.edu.ua

Harnaha Volodymyr - Ph.D., Associate Professor of the Department of Information Protection, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia. e-mail: garnaga.volodymyr@vntu.edu.ua

Немировська Дар'я Олександрівна – студентка групи 1БКС-22б, Факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, Україна, e-mail: nemyrovskadaria@gmail.com

Nemyrovska Daria Oleksandrivna - student of group 1BKS-22b, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, Ukraine, e-mail: nemyrovskadaria@gmail.com