**الجامـــعة الافتراضيّة السوريّة**

**ماجستير تقانات الشبكات**


# استخدام مصيدة مخترقين متنقلة ضمن شبكة

**رسالة أعدت لنيل درجة الماجستير في تقانات الشبكات**

**إعداد الطالب غيث علي شقرة**

بإشراف الدكتور محمد الجنيدي

**العام الدراسي (2025)**

Syrian Virtual University
Master's in network technologies

# Using a mobile honeypot within a network

Thesis Submitted in the Requirements FOR MASTER DEGREE IN NETWORK TECHNOLOGIES

Prepared by

Ghaith Ali Shaqra

Supervisor

Dr. Mohammad Aljuneidi

Year/2025

## جدول المحتويات:

## شكر وتقدير:

أتقدم بخالص الشكر وعظيم الامتنان إلى جميع القائمين على الجامعة الافتراضية السورية، لما بذلوه من جهودٍ مخلصة في دعم العملية التعليمية وتوفير بيئة أكاديمية محفزة. كما أخص بالشكر السادة أعضاء الهيئة التدريسية في برنامج ماجستير تقانات الشبكات، على ما قدموه من علمٍ وعطاءٍ طيلة مدة البرنامج، وعلى حرصهم الدائم على إثراء المعرفة وتوجيه الطلبة.

ولا يفوتني أن أتوجه بجزيل الشكر والعرفان للدكتور محمد الجنيدي، المشرف على هذا المشروع، لما قدمه من إرشاداتٍ سديدة، ونصائحٍ قيّمة، وإجاباتٍ وافية على مختلف التساؤلات والاستفسارات خلال فترة التحضير.

دمتم لنا ذخراً وسنداً، أساتذتي الأفاضل، وبارك الله في جهودكم.

النشر ضمن مخدم IEEE Tech RXIV للمقالات والدراسات العلمية

TechRxiv

EDIT PUBLISH

DOWNLOAD PDF 170 51

# Dynamic Honeypot Deployment for Enhanced Cybersecurity: Practical Implementation and Evaluation with Adaptive Address Switching

COMPUTING AND PROCESSING

ADAPTIVE DECEPTION CYBERSECURITY DYNAMIC HONEYPOTS ELASTIC STACK INTRUSION DETECTION NETWORK MONITORING SNORT THREAT INTELLIGENCE

Ghaith Shaqra, Mohammad Aljuneidi

**Abstract**

This paper presents a dynamic honeypot framework designed to overcome the limitations of static honeypots by automatically altering the honeypot's network identity, including MAC address, IP address, hostname, and cryptographic keys-to evade attacker detection. The proposed system integrates open-source tools such as Snort for intrusion detection, OpenCanary for honeypot emulation, Vector for log forwarding, and the Elastic Stack for real-time analysis and visualization. Operating as a closed-loop system, the framework adapts based on threat intelligence inputs. Evaluation results demonstrate a 239% increase in mean time-to-detection and a 50% reduction in false positives compared to static honeypots. This research offers a scalable, cost-effective, and open-source solution for proactive network defense, advancing the state of adaptive deception techniques.

**Preprint timeline**

05 Sep 2025 Submitted to *TechRxiv*

11 Sep 2025 Published in *TechRxiv*

Cite as: Ghaith Shaqra, Mohammad Aljuneidi. Dynamic Honeypot Deployment for Enhanced Cybersecurity: Practical Implementation and Evaluation with Adaptive Address Switching. *TechRxiv*. September 11, 2025.

DOI: 10.36227/techrxiv.175756416.67230506/v1

## الملخص:

تواجه المصائد الرقمية الثابتة تحديًا جوهريًا في بيئات الأمن السيبراني، يتمثل في قابليتها العالية للكشف من قبل المهاجمين خلال فترة زمنية قصيرة، مما يؤدي إلى تحييدها وفقدان فعاليتها في الرصد والتحليل. تسعى هذه الدراسة إلى معالجة هذه الإشكالية عبر تطوير استراتيجية ديناميكية لتبديل عنوان المصيدة الشبكي بشكل دوري أو حسب الحاجة، دون التأثير على بنيتها الداخلية أو تعريض الأنظمة الفعلية للخطر.

يعتمد المشروع على إعادة توجيه الاتصالات إلى المصيدة باستخدام آلية ذكية تُعيد تشكيل موقعها الشبكي، مما يُقلل من احتمالية التعرف عليها ويُطيل من عمرها التشغيلي. كما يتم تحليل التنبيهات الناتجة باستخدام نظام كشف التدخل SNORT وتحويلها عبر Vector إلى صيغة قابلة للمعالجة، ثم ربطها بمنصة Elasticsearch لعرضها وتحليلها ضمن سياق زمني وسلوكي متكامل.

تم تنفيذ العمل التطبيقي ضمن بيئة شبه إنتاجية، مع اختبار كل مرحلة باستخدام وحدات مستقلة لضمان صحة الأداء وقابلية التوسع. وتُظهر النتائج أن تبديل عنوان المصيدة يُسهم بشكل فعّال في تحسين قدرتها على التخفي، وزيادة دقة التنبيهات، وتعزيز الرؤية الأمنية الشاملة.

تُقدم هذه الدراسة نموذجًا عمليًا قابلًا للتطبيق في المؤسسات، يُعيد تفعيل المصائد الثابتة بآلية مرنة، ويُثري المحتوى البحثي في مجال استراتيجيات الرصد والتحكم، ويفتح آفاقًا جديدة لتطوير حلول أمنية أكثر تكيفًا مع التهديدات السيبرانية المتطورة.

# ABSTRACT

Static-address honeypots face a recurring challenge in cybersecurity environments: they are often detected by attackers within a short time frame, rendering them ineffective for sustained monitoring and threat analysis. This study addresses that limitation by proposing a dynamic strategy to periodically or conditionally change the honeypot's network address without altering its internal structure or compromising system integrity.

The project implements a smart redirection mechanism that reshapes the honeypot's perceived location, reducing its detectability and extending its operational lifespan. Alerts generated by the honeypot are analyzed using Snort, transformed via Vector into a structured format, and integrated with Elasticsearch for contextual and behavioral visualization.

The implementation was carried out in a semi-production environment, with each stage validated through independent testing units to ensure reliability and scalability. Results demonstrate that dynamic address switching significantly enhances the honeypot's stealth capabilities, improves alert accuracy, and strengthens overall situational awareness.

This study presents a practical model applicable in institutional settings, offering a flexible reactivation mechanism for static honeypots. It contributes to the research landscape by advancing strategic monitoring and control methodologies and opens new avenues for developing adaptive security solutions in response to evolving cyber threats.

## الكلمات المفتاحية:

الأمن السيبراني، المصائد الرقمية، تبديل العنوان الشبكي، Snort، Vector، Elasticsearch، تحليل التنبيهات، التوجيه الديناميكي، بيئة شبه إنتاجية، التمويه السيبراني، الرصد الاستباقي.

## مشكلة البحث:

تعتمد العديد من المؤسسات على المصائد الرقمية (Honeypots) كأداة فعالة لرصد الأنشطة الخبيثة وتحليل سلوك المهاجمين. إلا أن المصائد ذات العنوان الثابت تواجه تحديًا جوهريًا يتمثل في قابليتها العالية للكشف من قبل المهاجمين خلال فترة زمنية قصيرة، غالبًا تتراوح بين 48 إلى 72 ساعة، كما تشير الملاحظات الميدانية وتقارير الرصد الأمني (Zhang et al., ؛ Spitzner, 2003 2014).

هذا الكشف المبكر يؤدي إلى تحييد المصيدة، وفقدانها لقيمتها الاستخباراتية، ويُضعف من قدرتها على جمع بيانات دقيقة حول الهجمات المتقدمة.

المشكلة لا تقتصر على قابلية الكشف فقط، بل تمتد إلى غياب آلية فعالة لإعادة تفعيل المصيدة أو تحريكها دون التأثير على بنيتها أو عنوانها الشبكي. إذ أن معظم الحلول المقترحة في الأدبيات تعتمد على بناء مصائد جديدة أو تغيير العنوان يدويًا، وهي حلول غير عملية في البيئات الإنتاجية، وتفتقر إلى المرونة والاتساق.

من هنا، تنشأ الحاجة إلى استراتيجية ديناميكية تُعيد توظيف المصيدة الثابتة بآلية ذكية، تُبدل عنوانها الشبكي حسب الحاجة، وتُحافظ على قدرتها في الرصد والتحليل دون أن تُكتشف بسهولة. هذه الفجوة بين الاستخدام التقليدي للمصائد وبين الحاجة إلى ديناميكية التوجيه هي ما تسعى هذه الدراسة إلى معالجته، عبر نموذج تطبيقي قابل للتنفيذ في بيئة شبه إنتاجية، ويعتمد على أدوات مفتوحة المصدر مثل Snort و Vector و Elasticsearch.

## هدف البحث:

يهدف هذا البحث إلى تطوير استراتيجية ديناميكية لإعادة توظيف المصيدة الرقمية ذات العنوان الثابت، من خلال آلية ذكية لتبديل عنوانها الشبكي بشكل دوري أو حسب الحاجة، دون التأثير على بنيتها الداخلية أو تعريض الأنظمة الفعلية للخطر. تسعى هذه الاستراتيجية إلى تقليل قابلية كشف المصيدة من قبل المهاجمين، وإطالة عمرها التشغيلي، بما يُعزز من قدرتها على جمع بيانات أمنية دقيقة وتحليل سلوكيات الهجوم في بيئة شبه إنتاجية.

كما يهدف البحث إلى:

- اختبار فعالية آلية التوجيه الديناميكي في تحسين التمويه السيبراني.
- تحليل التنبيهات الناتجة عن المصيدة باستخدام أدوات مفتوحة المصدر مثل Snort وVector.
- ربط التنبيهات بمنصة Elasticsearch لعرضها وتحليلها ضمن سياق زمني وسلوكي متكامل.
- تقديم نموذج تطبيقي قابل للتنفيذ في المؤسسات، يُسهم في تحسين الرؤية الأمنية واتخاذ القرار.

## أسئلة البحث:

انطلاقًا من الإشكالية المتعلقة بقابلية كشف المصائد الرقمية ذات العنوان الثابت، وما يترتب على ذلك من فقدان فعاليتها في الرصد والتحليل، سعى هذا المشروع للإجابة على مجموعة من الأسئلة البحثية التي تُحدد جدوى الاستراتيجية المقترحة، وتُقيّم أثرها في بيئة شبه إنتاجية. وتتمثل الأسئلة الرئيسة فيما يلي:

1. **هل يمكن إعادة توظيف المصيدة الرقمية الثابتة عبر تبديل عنوانها الشبكي دون التأثير على بنيتها الداخلية؟**

يختبر المشروع إمكانية تنفيذ آلية ديناميكية للتوجيه تُعيد تشكيل موقع المصيدة دون الحاجة إلى تغيير بنيتها أو إعادة نشرها.

2. **ما مدى فعالية استراتيجية تبديل العنوان في تقليل قابلية كشف المصيدة من قبل المهاجمين؟**

تم تحليل سلوك المهاجمين بعد تطبيق التبديل، ومراقبة مدة بقاء المصيدة دون اكتشاف مقارنة بالوضع الثابت.

3. **هل يُسهم ربط المصيدة بمنظومة تحليل خارجية في تحسين دقة التنبيهات؟**

تم تقييم جودة التنبيهات الناتجة، ومدى ارتباطها بالسياق الزمني والسلوكي للهجوم، بعد تحويلها وتحليلها عبر أدوات متعددة.

4. **ما مدى قابلية تطبيق النموذج المقترح في بيئات إنتاجية حقيقية؟**

تم اختبار النموذج في بيئة شبه إنتاجية، وتحليل مدى توافقه مع متطلبات الأداء، التوسع، والتكامل مع الأنظمة الأمنية القائمة.

## أدوات البحث:

**1. الأدوات المستخدمة في البحث:**

- مصائد المخترقين Honeypot: حيث تم اختيار المصيدة Opencanary كونها مصيدة متعددة البروتوكولات (مثل HTTP، FTP، SSH) لاكتشاف الهجمات.
- نظام كشف ومنع التسلل (IDS/IPS) Snort: وهو أداة مفتوحة المصدر لتحليل حركة المرور الشبكية والكشف عن الهجمات.

**2. تحليل البيانات والمراقبة:**

Elastic Stack (ELK) **ويتضمن:**

- Elasticsearch: لتخزين وفهرسة بيانات السجلات.
- Vector: لجمع ومعالجة البيانات من مصادر متعددة.
- Kibana: لعرض البيانات وتحليلها عبر لوحات تحكم تفاعلية.

**3. الجدار الناري وإدارة الشبكة:**

- PfSense: جدار ناري مفتوح المصدر لإدارة حركة المرور وتطبيق قواعد الأمان بين الشبكات (LAN، DMZ، WAN).

**4. أدوات مساعدة:**

- Nmap: لاختبار اختراق الشبكة ومحاكاة الهجمات.
- VMWare Workstation: لإنشاء البيئة الافتراضية لتجربة النظام.
- برنامج إدارة مخصص (MobileTrapMonitor): مصمم بلغة C# لمراقبة المصائد وجمع التنبيهات.

**5. أدوات الاتصال والتوثيق:**

- SSH: للاتصال الآمن بالمصيدة والخوادم.

- Telegram API: لإرسال تنبيهات فورية عبر الرسائل.

**الخلاصة:**

تم استخدام مجموعة متكاملة من الأدوات مثل Opencanary لاكتشاف الهجمات، إلى جانب Snort للكشف عن التسلل، و Elastic Stack لتحليل البيانات. كما تم توظيف PfSense كجدار ناري، مع أدوات داعمة مثل Nmap للاختبار و VMWare لإنشاء البيئة الافتراضية. كل هذه الأدوات ساهمت في بناء نظام إنذار أمني متكامل للشبكات المحلية.

# جدول المصطلحات:

| الاختصار | المقابل بالعربية | المقابل بالإنجليزية | التعريف/الوظيفة |
|---|---|---|---|
| Honeypot | مصيدة المخترقين | Honeypot | نظام وهمي يحاكي الأنظمة الحقيقية لجذب المهاجمين وتسجيل أساليبهم |
| IDS | نظام كشف التسلل | Intrusion Detection System | يرصد الأنشطة المشبوهة ويولد إنذارات دون اتخاذ إجراءات وقائية |
| IPS | نظام منع التسلل | Intrusion Prevention System | يرصد الهجمات ويتخذ إجراءات (مثل حظر عناوين IP) تلقائية لوقفها |
| SIEM | نظام إدارة الأحداث الأمنية | Security Information and Event Management | يجمع بيانات الأمان من مصادر متعددة ويحللها لاكتشاف التهديدات |
| DMZ | المنطقة المنزوعة السلاح | Demilitarized Zone | منطقة عازلة بين الشبكة الداخلية والإنترنت تحتوي على الخدمات المعرضة للعموم (مثل خوادم الويب) |
| LAN | الشبكة المحلية | Local Area Network | شبكة مغلقة تغطي مساحة محدودة (مبنى أو شركة) |
| WAN | الشبكة الواسعة | Wide Area Network | شبكة تمتد عبر مواقع جغرافية متباعدة (مثل اتصال الفروع ببعضها) |
| SSH | بروتوكول النفاذ الآمن | Secure Shell | بروتوكول اتصال مشفر لإدارة الأنظمة عن بُعد |
| FTP | بروتوكول نقل الملفات | File Transfer Protocol | ينقل الملفات بين الأنظمة (غير آمن بدون تشفير إضافي) |
| HTTP/HTTPS | بروتوكول نقل النص التشعبي (آمن) | Hypertext Transfer Protocol (Secure) | HTTP مع طبقة تشفير SSL/TLS لحماية بيانات المستخدمين |
| DNS | نظام أسماء النطاقات | Domain Name System | يحول أسماء المواقع www.example.com إلى عناوين IP |
| API | واجهة برمجة التطبيقات | Application Programming Interface | تسمح للتطبيقات بالتواصل وتبادل البيانات بطرق محددة |

# الفصل الأول: المقدمة النظرية

## 1 مقدمة:

في ظل التزايد المستمر للهجمات السيبرانية وتطور أدوات وأساليب المهاجمين، أصبحت المصائد الرقمية (Honeypots) أحد أهم الوسائل التي تعتمدها فرق الدفاع السيبراني لرصد وتحليل الأنشطة الخبيثة داخل الشبكات. وقد أثبتت العديد من الدراسات، مثل دراسة Spitzner (2003) حول المصائد منخفضة التفاعل، أن هذه الأنظمة توفر رؤى قيمة حول سلوكيات المهاجمين، لكنها تعاني من قابلية الكشف السريع، خصوصًا إذا كانت تعتمد على عنوان ثابت أو نمط متكرر في التهيئة.

تشير الملاحظات الميدانية، المدعومة بتجارب منشورة مثل تلك التي وردت في تقرير مشروع "Honey Mesh" (Alshammari et al., 2021)، إلى أن المصائد الثابتة تُكتشف غالبًا خلال فترة تتراوح بين 48 إلى 72 ساعة من قبل نفس المهاجم، مما يؤدي إلى تحييدها وفقدان قيمتها الاستخباراتية. هذا التحدي يُضعف من قدرة فرق الأمن السيبراني على جمع معلومات دقيقة وطويلة الأمد حول أنماط الهجوم.

ينطلق هذا المشروع من هذه الإشكالية الجوهرية، ساعيًا إلى تطوير مصيدة رقمية واقعية يصعب التنبؤ بها، ويصمد أمام محاولات الكشف المبكر. لا يقتصر الحل المقترح على تغيير العنوان أو الشكل، بل يتضمن إعادة تصميم شاملة لمنظومة التنبيهات، بحيث يتم تنويع مصادرها وربطها بأنظمة تحليل خارجية مثل Elasticsearch وVector. وقد أثبتت هذه الأدوات فعاليتها في تحليل السجلات الأمنية على نطاق واسع، كما ورد في دراسة Gormley et al (2015) حول Elasticsearch، وفي وثائق Vector الرسمية التي توضح قدرته على معالجة تدفقات البيانات في الوقت الحقيقي.

يعتمد المشروع على منهجية هندسية دقيقة، تبدأ بتحليل سلوكيات المهاجمين، مرورًا بتصميم بيئة اختبار واقعية، وانتهاءً بربط النتائج بمنظومة تحليل قابلة للتطوير والتدقيق. كما يولي اهتمامًا خاصًا بتقليل الضوضاء الناتجة عن التنبيهات الزائفة، وتحسين دقة التصنيف، وهي مشكلة تناولتها دراسة Axelsson (2000) التي ناقشت أثر التنبيهات الكاذبة على كفاءة أنظمة الكشف.

يتناول هذا المشروع مراحل تطوير النظام، التحديات التقنية التي واجهت التنفيذ، الحلول المقترحة، وتحليل النتائج، مع تقديم رؤية مستقبلية لتوسيع النظام وتطبيقه في سيناريوهات أكثر تعقيدًا، بما يضمن فعالية المصيدة في بيئة إنتاجية حقيقية.

## 1-1 تنويع مصادر التنبيهات لرؤية أمنية أشمل:

في بيئات الشبكات الحديثة، لم تعد التنبيهات الأمنية المستخرجة من مصدر واحد كافية لتكوين صورة دقيقة عن الحالة الأمنية. الاعتماد على أداة واحدة أو نوع واحد من التحليل غالبًا ما يؤدي إلى نتائج مجتزأة أو مشوشة، خصوصًا في ظل تعقيد الهجمات وتعدد طبقاتها. لذلك، يسعى هذا المشروع إلى تنويع مصادر التنبيهات، بحيث يتم جمع وتحليل البيانات من أكثر من نقطة مراقبة، باستخدام أدوات متعددة مثل Snort، وVector، وربطها بمنصات تحليل خارجية مثل Elasticsearch.

وقد أثبتت دراسات متعددة، مثل دراسة Sommer & Paxson (2010) ، أن الدمج بين مصادر متعددة للتنبيهات يساهم في تحسين دقة الكشف وتقليل معدل التنبيهات الكاذبة. كما أن الربط بين أدوات التحليل في الزمن الحقيقي، كما هو الحال في Vector ، يتيح تتبع السياق الكامل للهجوم، وليس فقط الحدث المنعزل، مما يعزز من قدرة النظام على اتخاذ قرارات أمنية أكثر ذكاءً.

هذا التنوع في المصادر لا يهدف فقط إلى تحسين الكشف، بل إلى بناء **رؤية أمنية شاملة** (Security Situational Awareness)، وهي مفهوم ناقشته دراسة Endsley (1995) في سياق الأنظمة المعقدة، ويُقصد به قدرة النظام على فهم الحالة الراهنة، توقع التطورات، واتخاذ قرارات استباقية.

في هذا المشروع، يتم تطبيق هذا المفهوم عمليًا من خلال تصميم pipeline ديناميكي يجمع التنبيهات من مصادر متعددة، يطبعها، يصنفها، ويربطها بسياقها الزمني والمكاني، مما يتيح للباحث أو المسؤول الأمني رؤية متكاملة للحالة الأمنية، وليس مجرد إشارات متفرقة.

## 1-2 تحسين الكشف عبر التحليل الدقيق وربط النتائج بأنظمة خارجية:

في أنظمة الدفاع السيبراني، لا يكفي مجرد التقاط التنبيهات، بل يجب تحليلها بدقة وربطها بسياقها الكامل لفهم طبيعة التهديد واتخاذ القرار المناسب. تعتمد العديد من المنصات التقليدية على قواعد ثابتة أو نماذج تصنيف أولية، مما يؤدي إلى ارتفاع معدل التنبيهات الكاذبة (False Positives) أو فقدان التنبيهات الدقيقة (False Negatives) ، وهي مشكلة تناولتها دراسة Axelsson (2000) التي ناقشت حدود أنظمة الكشف التقليدية وأثر الضوضاء على كفاءة التحليل.

يسعى هذا المشروع إلى تحسين جودة الكشف من خلال تطبيق منهجيات تحليل دقيقة تعتمد على استخراج الحقول التقنية من التنبيهات وربطها بمنصات خارجية مثل Elasticsearch، التي تتيح البحث السياقي وتحليل الأنماط، وVector، التي تُمكّن من معالجة التدفقات وتحويل البيانات في الزمن الحقيقي. وقد أثبتت دراسة Gormley et al. (2015) أن استخدام Elasticsearch في تحليل

السجلات الأمنية يساهم في تحسين سرعة ودقة الاستجابة، بينما توضح وثائق Vector الرسمية كيف يمكن تحويل البيانات الخام إلى تنبيهات قابلة للتصنيف والتحليل.

كما يتم في هذا المشروع بناء وحدات اختبار صغيرة لكل مرحلة في pipeline التحليل، لضمان صحة كل خطوة قبل التشغيل الكامل، وهي منهجية هندسية تُستخدم في بيئات الإنتاج عالية الحساسية، كما ورد في دراسة Chen et al. (2018) حول تصميم أنظمة كشف قابلة للتوسع والتدقيق.

الهدف من هذا التحسين ليس فقط تقليل الضوضاء، بل بناء نظام قادر على التكيف مع أنماط الهجوم المتغيرة، وتحقيق توازن بين الحساسية والدقة، بما يضمن فعالية المصيدة في بيئة واقعية دون الحاجة إلى تدخل بشري مستمر.

## 1-3 بناء بيئة اختبار واقعية لدراسة سلوكيات الهجوم:

من التحديات الأساسية في تطوير أنظمة أمنية فعالة هو غياب بيئة اختبار واقعية تُحاكي ظروف الإنتاج الفعلية دون تعريض الأنظمة الحقيقية للخطر. تعتمد العديد من الدراسات على محاكاة نظرية أو بيئات معزولة، مما يُضعف من دقة النتائج ويحدّ من قابلية تعميمها. وقد ناقشت دراسة Mirkovic & Reiher (2004) هذه الإشكالية، مشيرة إلى أن فعالية أنظمة الدفاع ترتبط ارتباطًا مباشرًا بمدى واقعية بيئة الاختبار المستخدمة.

في هذا المشروع، تم تصميم بيئة اختبار شبه إنتاجية، تحتوي على خدمات حقيقية، سجلات فعلية، وتفاعل مباشر مع شبكة خارجية، مع تطبيق ضوابط صارمة للعزل والتحكم. هذا التصميم يتيح للمصيدة الرقمية أن تتفاعل مع المهاجمين كما لو كانت جزءًا من شبكة حقيقية، مما يسمح برصد سلوكياتهم بدقة، وتحليل أنماط الهجوم في سياقها الكامل.

وقد أثبتت تجارب سابقة، مثل مشروع "HoneyNet" (Spitzner, 2003)، أن المصائد الواقعية تُنتج بيانات أكثر ثراءً من المصائد النظرية، وتُظهر سلوكيات معقدة لا يمكن ملاحظتها في بيئات معزولة. كما أن دمج هذه البيئة مع أدوات تحليل مثل Vector و Elasticsearch يتيح تتبع الهجوم من لحظة الدخول وحتى مرحلة التفاعل، مما يوفر سلسلة زمنية دقيقة يمكن تحليلها لاحقًا.

هذا النهج لا يهدف فقط إلى جمع بيانات أكثر واقعية، بل إلى اختبار فعالية النظام نفسه تحت ضغط حقيقي، مما يُمكّن من تحسينه وتعديله قبل نشره في بيئة إنتاجية فعلية.

## 1-4 بناء نظام قابل للتطوير والتدقيق:

في بيئات الأمن السيبراني الإنتاجية، لا يكفي أن يكون النظام فعالًا في لحظة معينة، بل يجب أن يكون قابلًا للتطوير، قابلًا للتدقيق، وقادرًا على التكيف مع تغيرات البنية التحتية والهجمات. يعتمد

هذا المشروع على فلسفة هندسية واضحة ترفض التعقيد غير المبرر، وتُصر على أن تكون كل خطوة في النظام قابلة للقياس، الفحص، والتطوير المستقبلي دون الحاجة إلى إعادة بناء شاملة.

وقد ناقشت دراسة Kim & Lee (2019) أهمية الوضوح البنيوي في تصميم أنظمة الأمن، مشيرة إلى أن الأنظمة التي تفصل بوضوح بين مهام الاتصال، التحليل، والعرض تكون أكثر قابلية للصيانة والتوسع. في هذا السياق، تم تصميم pipeline المشروع بحيث يفصل بين مراحل جمع البيانات، تحليلها، تحويلها، وعرضها، مع إمكانية اختبار كل مرحلة بشكل مستقل عبر وحدات اختبار صغيرة(Unit Testing) ، وهي منهجية أثبتت فعاليتها في تقليل الأخطاء وتحسين جودة النشر، كما ورد في دراسة Fowler (2004) حول التصميم القائم على الاختبار.

كما أن رفض التعقيد غير المدعوم بالبيانات يُعد من المبادئ الأساسية في هذا المشروع، حيث يتم تجنب أي افتراضات غير مثبتة، ويُعتمد فقط على التحليل الفعلي للبيانات القادمة من المصيدة. هذا النهج يُمكّن من بناء نظام واقعي، قابل للتدقيق من قبل فرق أخرى، ويمكن توسيعه مستقبلًا بإضافة مصادر جديدة أو أدوات تحليل دون الحاجة إلى تعديل جذري في البنية.

النتيجة هي نظام هندسي متماسك، يربط بين النظرية والتطبيق، ويُقدّم نموذجًا عمليًا لتصميم أنظمة أمنية قابلة للتكيف مع الواقع، دون التضحية بالوضوح أو القابلية للتطوير.

## 1-5 تحويل النموذج التجريبي إلى نظام إنتاجي قابل للنشر:

غالبًا ما تبقى مشاريع المصائد الرقمية في إطار النموذج التجريبي (Prototype) ، دون أن تصل إلى مرحلة النشر الفعلي في بيئات إنتاجية، بسبب تعقيدات التكامل، ضعف التوثيق، أو محدودية التوسع. في هذا المشروع، تم تجاوز هذه العقبة من خلال تصميم نظام هندسي متكامل، قابل للنشر في بيئات حقيقية، مع مراعاة متطلبات الأداء، الأمان، والتوافق مع أنظمة التحليل الخارجية.

وقد ناقشت دراسة Shiravi et al. (2012) أهمية الانتقال من النماذج التجريبية إلى أنظمة قابلة للنشر، مشيرة إلى أن القيمة الحقيقية للمصيدة لا تتحقق إلا عندما تُدمج ضمن منظومة أمنية متكاملة، وتُستخدم فعليًا في مراقبة الشبكة. في هذا السياق، تم بناء النظام بحيث يدعم التهيئة الديناميكية، الربط عبر واجهات API ، وتوليد التنبيهات بصيغة قياسية مثل NDJSON ، مما يُسهّل إرسالها إلى أدوات مثل Vector و Elasticsearch دون الحاجة إلى تحويلات إضافية.

كما تم اختبار كل مرحلة من مراحل النظام باستخدام وحدات اختبار مستقلة، لضمان صحة الأداء قبل النشر، وهي منهجية موصى بها كما ورد في دراسة Williams & Wiggins (2020) ، التي تناولت أهمية التحقق المستمر في أنظمة الأمن السيبراني.

هذا التحول من النموذج التجريبي إلى النظام الإنتاجي لا يُعد مجرد خطوة تقنية، بل هو انتقال فلسفي نحو بناء أدوات أمنية واقعية، قابلة للتشغيل والتوسع، وتُعالج التهديدات الفعلية بدلًا من الاكتفاء بمحاكاتها.

## 1-6 أهمية الأمن السيبراني والتحديات المعاصرة:

في ظل التحول الرقمي المتسارع الذي يشهده العالم، أصبح الأمن السيبراني ضرورة استراتيجية لا غنى عنها لحماية الأصول الرقمية، وضمان استمرارية الأعمال، والحفاظ على خصوصية الأفراد والمؤسسات. فمع تزايد الاعتماد على الأنظمة الذكية، والحوسبة السحابية، وإنترنت الأشياء، باتت البيانات الحساسة عرضة لهجمات إلكترونية متطورة تستهدف البنية التحتية الحيوية، وتستغل الثغرات الأمنية غير المكتشفة.

يُعد الأمن السيبراني اليوم أكثر من مجرد تقنية دفاعية؛ إنه منظومة متكاملة تشمل السياسات، والتقنيات، والوعي البشري، تهدف إلى تحقيق ثلاثية الحماية: السرية، والسلامة، والتوافر. وقد أظهرت تقارير دولية أن الهجمات السيبرانية مثل هجمات الفدية (Ransomware) ، والتصيد الاحتيالي (Phishing)، والبرمجيات الخبيثة (Malware) ، تتزايد بوتيرة غير مسبوقة، مما يفرض على المؤسسات تبني استراتيجيات دفاعية متقدمة.

## 1-7 أبرز التحديات التي تواجه الأمن السيبراني:

- التطور المستمر للتهديدات: تتغير أساليب الهجوم بسرعة، مما يصعّب على الأنظمة التقليدية مواكبتها.
- نقص الوعي الأمني: لا يزال العديد من الموظفين يشكلون نقطة ضعف بسبب ضعف التدريب أو الجهل بالمخاطر.
- هجمات سلسلة التوريد: تستهدف الجهات المزودة بالخدمات، مما يهدد شبكات متعددة في آن واحد.
- هجمات (Zero-Day): تستغل ثغرات لم تُكتشف بعد، مما يجعلها من أخطر أنواع الهجمات.
- نقص الموارد البشرية والمالية: تعاني كثير من المؤسسات من ضعف في الكفاءات أو التمويل اللازم لتطبيق حلول فعالة.

## 1-8 التحديات التي تواجه الشبكات والمؤسسات في مواجهة الهجمات السيبرانية:

تواجه الشبكات والمؤسسات اليوم بيئة رقمية شديدة التعقيد، تتسم بتوسع سطح الهجوم، وتنوع مصادر التهديد، وتسارع وتيرة الابتكار في أدوات الاختراق. هذا الواقع يفرض تحديات متعددة على المستويين التقني والإداري، ويستلزم استجابة أمنية مرنة ومبنية على تحليل دقيق للبيانات، ومنها:

- **تعدد نقاط الدخول** :مع انتشار الأجهزة الطرفية، والخدمات السحابية، وتكامل أنظمة الطرف الثالث، أصبحت الشبكات عرضة لهجمات من عدة اتجاهات يصعب مراقبتها جميعًا بشكل فعال.
- **الهجمات المتقدمة المستمرة (APT):** تعتمد هذه الهجمات على التسلل التدريجي والبقاء داخل الشبكة لفترات طويلة دون اكتشاف، مما يتطلب أدوات رصد وتحليل سلوكي متقدمة.
- **ضعف التنسيق بين الفرق الفنية والإدارية:** يؤدي غياب التكامل بين فرق الشبكات، والأمن، والإدارة إلى تأخر الاستجابة للحوادث، أو اتخاذ قرارات غير مدعومة بالبيانات.
- **نقص الموارد البشرية المؤهلة:** تعاني المؤسسات من نقص في الكفاءات القادرة على تحليل التنبيهات، وتطوير قواعد كشف دقيقة، وربطها بأنظمة خارجية مثل SIEM أو Elasticsearch.
- **الاعتماد الزائد على الحلول الجاهزة:** كثير من المؤسسات تعتمد على أدوات كشف جاهزة دون تخصيص أو اختبار فعلي، مما يؤدي إلى نتائج غير دقيقة أو تنبيهات زائفة.
- **ضعف إدارة التنبيهات:** تتراكم التنبيهات دون تنظيم أو تصنيف، مما يصعب على الفرق الأمنية التمييز بين الحوادث الحقيقية والزائفة، ويؤدي إلى إرهاق تنبيهي (Alert Fatigue).

**وقد تجلت انعكاسات هذه التحديات بما يلي:**

- ارتفاع زمن الاستجابة للحوادث.
- زيادة احتمالية اختراق الشبكة دون اكتشاف.
- تراجع الثقة في النظام الأمني الداخلي.
- صعوبة إثبات الامتثال للمعايير التنظيمية مثل ISO 27001 أوNIST .

### 1-9 دور تقنيات Honeypots و Elastic في تحسين الأمن السيبراني:

برزت تقنيات مثل Honeypots و Elastic Stack كأدوات فعالة لتحسين الرصد والتحليل والاستجابة للحوادث الأمنية في مواجهة التهديدات السيبرانية المتزايدة. هذه التقنيات لا تقتصر على الكشف، بل تتيح أيضًا فهمًا عميقًا لسلوك المهاجمين، وتوفير بيانات دقيقة لدعم اتخاذ القرار الأمني.

#### 1-9-1 مصائد المحترقين (Honeypots) للرصد الاستباقي وفهم سلوك المهاجم:

تُعد الـ Honeypots أنظمة افتراضية أو حقيقية تُصمم لاستدراج المهاجمين، بحيث تُظهر نفسها كأهداف سهلة، بينما تقوم بتسجيل وتحليل كل نشاط يتم داخلها. ومن أبرز فوائدها:

- كشف الهجمات غير المعروفة: لأنها لا تحتوي على حركة شرعية، فإن أي نشاط يُعتبر مشبوهًا.
- تحليل أساليب الهجوم: توفر بيانات واقعية عن أدوات وتقنيات المهاجمين.
- تقليل الضوضاء الأمنية: عبر فصل التنبيهات الحقيقية عن الزائفة.
- دعم تطوير قواعد كشف دقيقة: من خلال دراسة الأنماط الفعلية للهجمات.

#### 1-9-2 تحليل البيانات وتكامل التنبيهات من خلال (Elastic Stack):

يُستخدم Elastic Stack (Elasticsearch, Vector, Kibana) كمنصة قوية لتحليل البيانات الأمنية، ويُعد حجر الأساس في بناء أنظمة SIEM مرنة. وتشمل أبرز أدواره مايلي:

- تخزين واسترجاع السجلات بكفاءة عالية مما يتيح تحليلًا زمنيًا دقيقًا للحوادث.
- تصور البيانات عبر Kibana لتحديد الأنماط والاتجاهات الأمنية.
- دمج التنبيهات من مصادر متعددة مثل Snort وHoneypots، مما يعزز الرؤية الشاملة.
- دعم التحقيقات الجنائية الرقمية :عبر تتبع الأحداث وربطها زمنياً وسلوكياً.

#### 1-9-3 التكامل بين Honeypots و Elastic في بيئة إنتاجية:

عند دمج Honeypots مع Elastic ، يتم إنشاء نظام رصد وتحليل متكامل، حيث تُرسل التنبيهات الناتجة عن الأنشطة المشبوهة إلى Elastic لتحليلها وتصنيفها. هذا التكامل يتيح:

- بناء نموذج سلوكي واقعي للهجمات.
- تحسين دقة التنبيهات وتقليل الإنذارات الكاذبة.
- دعم الاستجابة التلقائية عبر ربط النتائج بأنظمة خارجية.

## 1-10 مواضع الجذب (مصائد المخترقين) Honeypots:

كفكرة عامة تعمل مواضع الجذب honeypots على جذب المهاجمين وإعطائهم التصور والاعتقاد بأنهم قد نجحوا بالدخول إلى الشبكة الحقيقية، وذلك من خلال إنشاء نظام شبيه بها ولكنه غير حقيقي حيث يتم استخدامه كفخ للمهاجمين، ومن خلال مراقبة هذه المواضع الوهمية، يمكن للمؤسسات التحقق من أساليب الهجمات المختلفة التي تستخدمها الجهات الخبيثة، وبالتالي تطوير حلول مضادة لهذه الهجمات.

وتعتبر مواضع الجذب honeypots من التقنيات الفعالة في مجال الأمن السيبراني، حيث تتيح للقائمين على الأمن الكشف المبكر عن الهجمات، والتحكم فيها، ومنع التعرض للأجهزة والبيانات المخزنة في الشبكة.

إن تقنيات مواضع الجذب honeypots تتطور باستمرار لتصبح أكثر فعالية وصعوبة في كشفها من قبل المهاجمين، مما يساعد في تحسين مستوى الأمان السيبراني للمؤسسات.

يمكن أن تتلخص فكرة مواضع الجذب أو مصائد المخترقين بعبارة بسيطة كالتالي:

**" دعوة المتسللين للداخل بدلاً من ابقائهم في الخارج وهذا يعني السماح للقائمين على الأمن بالهجوم بدلاً من الدفاع فقط".**

تعرف مواضع الجذب honeypots بأنها أجهزة أو أنظمة وهمية يتم إنشاؤها لتحاكي الأنظمة والخدمات الحقيقية في شبكات المؤسسات بهدف جذب المهاجمين واختبار قدراتهم ومعرفة أساليبهم وأدواتهم في الهجوم على الشبكات.

وتعود فكرة استخدام مواضع الجذب honeypots إلى العام 1986، حيث قام باحثان في مجال الأمن السيبراني يدعيان "كليفود" و"بروفيت" باستخدام أنظمة معدلة لجذب المهاجمين وتتبع أنشطتهم. وقد اتخذت هذه المواضع شكلاً بسيطاً واستخدمت بشكل أساسي لأغراض البحث والتجربة.

ومنذ ذلك الحين، تطورت فكرة مواضع الجذب honeypots وأصبحت تستخدم بشكل واسع في الأنظمة الأمنية للمؤسسات والشركات والحكومات لمراقبة وتتبع نشاط المهاجمين والهجمات السيبرانية المنفذة، ويمكن تلخيص ضرورات استخدام مواضع الجذب Honeypot بما يلي:

- ✓ تحديد وتقييم مستوى التهديد: حيث يمكن لمواضع الجذب honeypots تسجيل كل الأنشطة المرتبطة بالمهاجمين، مما يتيح للمؤسسة تقييم مدى تعرضها للهجمات ومعرفة أنواع الهجمات التي يتعرضون لها.

- ✓ تطوير وتحسين أنظمة الحماية والأمان: حيث يمكن للمؤسسات والشركات الاستفادة من المعلومات المتعلقة بنوع الهجمات المنفذة ضدهم والأنظمة التي يستخدمها المهاجمون لتحديث وتحسين أنظمة الحماية والأمان لديهم.
- ✓ التدريب والتعليم: حيث يمكن استخدام مواضع الجذب honeypots لتدريب الفرق الأمنية الخاصة بالمؤسسة على كيفية التعامل مع الهجمات السيبرانية والتعرف على أساليب الهجوم المستخدمة من قبل المهاجمين.
- ✓ تحديد محاولات الاختراق: حيث يتم استخدام مواضع الجذب للكشف عن محاولات الاختراق وتحديد أساليب الهجوم وأدواته.
- ✓ تجميع المعلومات الهامة: حيث يمكن لمواضع الجذب honeypots جمع معلومات هامة عن المهاجمين وأدواتهم ونشاطاتهم، ويمكن استخدام هذه المعلومات لتحسين أمن الشبكات وتطوير إجراءات الحماية.
- ✓ منع الهجمات الحقيقية: حيث يمكن استخدام مواضع الجذب honeypots لجذب المهاجمين وتشغيلهم في بيئة آمنة ومنعهم من الوصول إلى الأنظمة والبيانات الحقيقية في الشبكة.
- ✓ تقليل الضرر الناجم عن الهجمات: حيث يمكن استخدام مواضع الجذب honeypots لتوجيه المهاجمين إلى بيئة افتراضية منفصلة عن الأنظمة الحقيقية في الشبكة، وبذلك يتم تقليل الضرر الناجم عن الهجمات السيبرانية.
- ✓ تحسين التحليل الأمني: حيث يمكن استخدام مواضع الجذب honeypots لتحليل الهجمات السيبرانية وتحديد أساليب الهجوم ونقاط الضعف في الأنظمة والخدمات والبناء عليها مستقبلاً.
- ✓ التأكد من فعالية إجراءات الحماية: حيث يمكن استخدام مواضع الجذب honeypots للتأكد من فعالية إجراءات الحماية المطبقة والمتبعة في المؤسسات وتحسينها في حال كانت غير كافية.

بشكل عام، يعتبر استخدام مواضع الجذب honeypots أداة فعالة لمكافحة الهجمات السيبرانية وتقييم واختبار أمان الشبكات وتحديد أساليب الهجوم، كما تساعد على جمع المعلومات الهامة عن المهاجمين وتحسين أنظمة الحماية والأمان الخاصة بالمؤسسات والشركات والحكومات.

وعلى الرغم من فوائد ومميزات استخدام مواضع الجذب honeypots، إلا أنها تتطلب عناية وحرص في تطبيقها وتشغيلها، ويجب تطبيق إجراءات الأمان المناسبة لحمايتها من الاختراق، ويجب أن يتم تحديد أهداف استخدام مواضع الجذب honeypots وتحديد الأنظمة والخدمات التي تستخدم كمواضع جذب، ويجب تحديثها وإدارتها بشكل مستمر لتحقيق أقصى قدر من الفائدة.

## 1-11 الأخطار الأمنية التي تتعرض لها البيئات الرقمية:

مع التحول الرقمي المتسارع، أصبحت البيئات الرقمية عرضة لمجموعة متزايدة من الأخطار الأمنية التي تهدد سرية البيانات، وسلامة الأنظمة، واستمرارية الخدمات. هذه الأخطار لا تقتصر على الهجمات التقنية المباشرة، بل تشمل أيضًا التهديدات الاجتماعية والتنظيمية التي تستغل نقاط الضعف البشرية والتقنية على حد سواء.

### 1-11-1 أبرز أنواع الأخطار الأمنية:

- البرمجيات الخبيثة (Malware): تشمل الفيروسات، وأحصنة طروادة، وبرامج الفدية، وتُستخدم لاختراق الأنظمة وسرقة البيانات أو تعطيل الخدمات. تُعد من أكثر التهديدات شيوعًا، وغالبًا ما تنتقل عبر الروابط الضارة أو البريد الإلكتروني التصيدي.
- هجمات الفدية (Ransomware): تعتمد على تشفير البيانات وطلب فدية لفك التشفير، وقد تسببت في خسائر مالية ضخمة للمؤسسات حول العالم.
- التصيد الاحتيالي (Phishing): يستهدف المستخدمين برسائل مزيفة تهدف إلى سرقة بيانات الدخول أو المعلومات المالية، ويُعد من أكثر الهجمات نجاحًا بسبب ضعف الوعي الأمني.
- هجمات الحرمان من الخدمة (DDoS): تهدف إلى تعطيل الأنظمة أو المواقع عبر إغراقها بطلبات وهمية، مما يؤدي إلى توقف الخدمات الحيوية.
- الثغرات الأمنية غير المكتشفة (Zero-Day): تُستغل قبل أن يتم التعرف عليها أو إصدار تحديثات لها، مما يجعلها من أخطر أنواع الهجمات.
- الهجمات الداخلية: قد تصدر من موظفين أو شركاء لديهم صلاحيات وصول، ويستغلونها لأغراض خبيثة أو نتيجة لإهمال غير مقصود.
- الهندسة الاجتماعية (Social Engineering): تعتمد على التلاعب النفسي لخداع الأفراد ودفعهم للكشف عن معلومات حساسة أو تنفيذ إجراءات ضارة.

**ومن التأثيرات المحتملة لهذه الأخطار:**

- خسائر مالية مباشرة نتيجة توقف الأنظمة أو دفع الفدية.
- انتهاك الخصوصية وتسريب معلومات حساسة.
- فقدان الثقة المؤسسية من قبل العملاء والشركاء.
- التعرض للمساءلة القانونية بسبب خرق الامتثال التنظيمي.
- تعطيل العمليات التشغيلية مما يؤثر على الإنتاجية والاستقرار.

وقد أظهرت تقارير حديثة أن عدد الجرائم الإلكترونية تضاعف ثلاث مرات خلال السنة الأولى من جائحة كوفيد-19، نتيجة الاعتماد المتزايد على العمل عن بُعد والخدمات السحابية، مما يعكس الحاجة الملحة إلى أنظمة أمنية مرنة وقابلة للتكيف مع هذا الواقع المتغير.

### 1-11-2 الحاجة إلى حلول عملية في مواجهة الأخطار الأمنية:

في ظل هذا المشهد المعقد من التهديدات الأمنية المتنوعة، لم تعد الحلول التقليدية كافية لرصد الهجمات أو فهم سلوكيات المهاجمين بشكل دقيق. إذ تتطلب البيئات الرقمية الحديثة أدوات تحليلية متقدمة قادرة على محاكاة الواقع، واستخلاص الأنماط، وربط التنبيهات بمصادرها الفعلية. ومن هنا تنبع الحاجة إلى أنظمة رصد ديناميكية تعتمد على بيانات حقيقية، وتتيح للمختصين اختبار الفرضيات الأمنية في بيئة شبه إنتاجية دون تعريض الأنظمة الفعلية للخطر.

انطلاقًا من هذا التحدي، يأتي هذا المشروع كمحاولة لتقديم نموذج عملي متكامل يجمع بين النظرية والتطبيق، عبر بناء مصيدة رقمية (Honeypot) واقعية، وتحليل التنبيهات الناتجة عنها باستخدام أدوات متقدمة مثل Snort وVector، وربطها بمنظومات خارجية لتحسين الرؤية الأمنية واتخاذ القرار.

## 1-12 الفجوة العلمية التي تسعى الدراسة إلى سدّها:

رغم الانتشار الواسع لاستخدام المصائد الرقمية (Honeypots) في بيئات الأمن السيبراني، إلا أن معظم الدراسات والتطبيقات العملية تعتمد على مصائد ذات عنوان شبكي ثابت، مما يجعلها عرضة للكشف السريع من قبل المهاجمين، وبالتالي فقدانها لقيمتها الاستخباراتية بعد فترة قصيرة من النشر. هذه الإشكالية تُضعف من فعالية المصيدة في جمع بيانات واقعية وتحليل سلوكيات الهجوم، وتُقلل من قدرتها على دعم القرارات الأمنية المبنية على السياق.

الفجوة العلمية هنا لا تكمن في غياب المصائد الرقمية، بل في غياب آلية ديناميكية قابلة للتطبيق لإعادة توجيه المصيدة أو تبديل عنوانها الشبكي دون التأثير على بنيتها الداخلية أو تعريض الأنظمة الفعلية للخطر. فالأدبيات الحالية تفتقر إلى نماذج تطبيقية واقعية تُعالج هذه الإشكالية ضمن بيئة شبه إنتاجية، وتربط بين المصيدة الثابتة وأدوات تحليل متقدمة مثل Snort وVector و Elasticsearchبطريقة متماسكة وقابلة للتوسع.

كما أن معظم الحلول المطروحة في الدراسات السابقة إما نظرية أو تعتمد على إعادة نشر المصيدة بالكامل، وهو ما لا يتناسب مع متطلبات البيئات الإنتاجية التي تحتاج إلى استمرارية في الرصد ومرونة في التوجيه دون تعطيل أو إعادة تهيئة.

من هنا، يسعى هذا المشروع إلى سدّ هذه الفجوة من خلال:

- تقديم استراتيجية عملية لتبديل عنوان المصيدة وهويتها بشكل ديناميكي.
- اختبار هذه الاستراتيجية ضمن بيئة شبه إنتاجية باستخدام أدوات مفتوحة المصدر.
- تحليل أثر التبديل على قابلية الكشف، وجودة التنبيهات، واستمرارية الرصد.
- ربط النتائج بمنظومة تحليل خارجية تُمكّن من عرض السياق الكامل للهجوم.

ومما سبق نجد أن مشكلة الأمن السيبراني تمثل تحديًا كبيرًا يتطلب استراتيجيات مبتكرة وأدوات فعالة. من خلال التركيز على تطوير نظام إنذار أمني يعتمد على تقنية Honeypot قابلة للتنقل، يسعى هذا المشروع إلى معالجة الفجوات الحالية في الأبحاث وتقديم حلول عملية يمكن أن تعزز من مستوى الأمان في الشبكات المحلية. إن فهم هذه المشكلة وأهميتها سيساهم في تعزيز الأمن السيبراني وحماية المعلومات الحساسة في المؤسسات.

## 1-13 المخرجات الرئيسية للبحث:

يهدف هذا البحث إلى انتاج نظام أمن سيبراني للشبكات المحلية من خلال إنشاء نظام إنذار أمني شامل يعتمد على تقنية مصائد المخترقين قابلة للتنقل (Mobile Honeypots)، حيث تم السعي إلى تقديم حلول فعالة لرصد محاولات الاختراق وتحليل سلوك المهاجمين، مما يساهم في تحسين استراتيجيات الدفاع السيبراني. من خلال هذا النظام، سيتمكن الباحثون والممارسون من فهم التهديدات بشكل أعمق وتطبيق تدابير وقائية فعالة لحماية المعلومات الحساسة من خلال انتاج:

- نموذج عمل كامل: مصيدة متنقلة قابلة للتنقل باستخدام أدوات مفتوحة المصدر.
- حزم برمجية جاهزة.

- سكريبتات إدارة التنقل (C#, Bash).
- تكوينات Opencanary قابلة للتخصيص.
- دليل تشغيلي: خطوات نشر الحل في شبكات حقيقية.

حيث تعتبر هذه المبادرة خطوة مهمة نحو تعزيز قدرة المؤسسات على مواجهة التهديدات السيبرانية المتزايدة، مما يسهم في خلق بيئة أكثر أمانًا للبيانات والمعلومات بالاعتماد على برمجيات مجانية ومفتوحة المصدر مناسبة جداً للتطبيق العملي ضمن بيئة عمل المؤسسات.

## 1-14 أهمية البحث:

### 1-14-1 أهمية البحث من الناحية العلمية:

أولاً: توسيع المعرفة في مجال الأمن السيبراني:

- يسهم هذا البحث في إثراء الأدبيات العلمية المتعلقة بالأمن السيبراني من خلال تقديم نموذج متكامل يعتمد على تقنيات مصائد المخترقين قابلة للتنقل.
- يوفر البحث رؤى جديدة حول كيفية استخدام هذه التقنيات بشكل فعال لرصد الهجمات وتحليل سلوك المهاجمين، مما يعزز من فهم التهديدات السيبرانية.

ثانياً: تحديد الفجوات البحثية:

- يسلط البحث الضوء على الفجوات الحالية في الأبحاث المتعلقة بتقنيات Honeypots، مما يشجع الباحثين على استكشاف المزيد من التطبيقات والتقنيات الجديدة في هذا المجال.
- من خلال تقديم توصيات حول كيفية تحسين استراتيجيات الكشف عن التهديدات، يفتح البحث آفاقًا جديدة لدراسات مستقبلية.

ثالثاً: تطوير منهجيات جديدة:

- يقدم البحث منهجيات جديدة لتحليل البيانات المستخرجة من مصائد المخترقين المتنقلة، مما يمكن أن يسهم في تطوير أدوات وتقنيات جديدة في مجال الأمن السيبراني.
- يمكن أن تكون هذه المنهجيات أساسًا لتطوير نماذج تعلم آلي جديدة لتحسين الكشف عن التهديدات.

## 1-14-2 أهمية البحث من الناحية العملية:

أولاً: تحسين مستوى الأمان في المؤسسات:

- يوفر البحث نموذجًا عمليًا يمكن تطبيقه في المؤسسات المختلفة، مما يسهم في تحسين مستوى الأمان السيبراني وحماية البيانات الحساسة.
- من خلال استخدام تقنيات Honeypots المتنقلة، يمكن للمؤسسات جذب المهاجمين وتحليل سلوكهم، مما يساعد في تطوير استراتيجيات دفاعية أكثر فعالية.

ثانياً: توفير أدوات فعالة لرصد التهديدات:

- يساهم البحث في تطوير أدوات فعالة لرصد التهديدات في الوقت الفعلي، مما يمكن المؤسسات من اتخاذ إجراءات سريعة للتصدي للهجمات.
- من خلال دمج أدوات تحليل البيانات مثل Elastic Stack، يمكن تحسين قدرة المؤسسات على تحليل البيانات واستخراج المعلومات القيمة.

ثالثاً: تعزيز الوعي الأمني:

- يسهم البحث في زيادة الوعي الأمني بين الموظفين من خلال تقديم معلومات حول أساليب الهجوم وسلوك المهاجمين.
- يمكن أن يؤدي هذا الوعي إلى تحسين ممارسات الأمان داخل المؤسسات وتقليل الأخطاء البشرية التي قد تؤدي إلى اختراقات.

رابعاً: تقديم استراتيجيات استجابة فعالة:

- يوفر البحث توصيات حول كيفية تحسين استراتيجيات الاستجابة للهجمات، مما يساعد المؤسسات على تقليل الأضرار الناتجة عن الهجمات السيبرانية.
- من خلال تطوير إجراءات استباقية، يمكن للمؤسسات حماية نفسها بشكل أفضل من التهديدات المستقبلية.

## 1-14-3 كيف يمكن أن يساهم البحث في تحسين الأمن السيبراني؟

### 1-14-3-1 تقديم نموذج متكامل:

- من خلال تطوير نظام إنذار أمني يعتمد على مصائد المخترقين المتنقلة، يمكن للمؤسسات تحسين قدرتها على الكشف عن الهجمات وتحليلها.
- هذا النموذج يمكن أن يكون مرجعًا للمؤسسات التي تسعى لتعزيز أمانها السيبراني.

### 1-14-4 تحليل سلوك المهاجمين:

- من خلال جمع البيانات من مصائد المخترقين، يمكن للمؤسسات فهم سلوك المهاجمين بشكل أفضل، مما يساعد في تطوير استراتيجيات دفاعية أكثر فعالية.
- هذا التحليل يمكن أن يوفر رؤى حول الأنماط الشائعة للهجمات، مما يسهل على المؤسسات اتخاذ تدابير وقائية.

### 1-14-5 تطوير استراتيجيات دفاعية:

- يساهم البحث في تطوير استراتيجيات دفاعية تعتمد على البيانات المستخرجة من مصائد المخترقين، مما يمكن المؤسسات من تحسين مستوى الأمان بشكل مستمر.
- من خلال دمج تقنيات تعلم الآلة، يمكن تحسين الكشف عن التهديدات وتقديم تنبيهات فورية للمسؤولين عن الأمن.

### 1-14-6 تعزيز التعاون بين المؤسسات:

- يمكن أن يسهم البحث في تعزيز التعاون بين المؤسسات من خلال تبادل المعلومات حول التهديدات وأساليب الهجوم، مما يعزز من مستوى الأمان السيبراني بشكل عام.
- من خلال بناء شبكة من التعاون، يمكن للمؤسسات تحسين قدرتها على التصدي للهجمات السيبرانية.

### 1-14-7 الخلاصة:

إن أهمية هذا البحث تتجلى في قدرته على تحسين الأمن السيبراني من خلال تقديم نموذج متكامل يعتمد على تقنيات مصائد المخترقين المتنقلة. من خلال تعزيز المعرفة العلمية وتقديم حلول عملية، يسعى البحث إلى تعزيز مستوى الأمان في المؤسسات وحماية البيانات الحساسة من التهديدات المتزايدة. إن النتائج والتوصيات التي يقدمها البحث يمكن أن تكون أساسًا لتطوير استراتيجيات أمان أكثر فعالية في المستقبل.

## 1-15 منهجية البحث:

### 1-15-1مقدمة:

تعتبر منهجية البحث جزءًا أساسيًا من أي دراسة أكاديمية، حيث تحدد الطريقة التي سيتم بها جمع البيانات وتحليلها للوصول إلى النتائج. في هذا البحث، تم اعتماد منهجية شاملة تجمع بين التصميم العملي، الاختبارات، والتحليل، بهدف تطوير نظام إنذار أمني يعتمد على تقنية مصائد المخترقين المتنقلة (Mobile Honeypots) لتحسين أمان الشبكات المحلية.

### 1-15-2التصميم العملي:

تم تصميم النظام بشكل متكامل يتضمن عدة مكونات رئيسية، وهي:

الشبكة المعتمدة:

تم إعداد بيئة عمل افتراضية تتكون من ثلاث مناطق رئيسية:

- الشبكة الداخلية (LAN): تحتوي على المخدمات الرئيسية وأجهزة العمل.

- الشبكة المعزولة (DMZ): تضم المخدمات المعرضة للوصول من الإنترنت.
- الشبكة الخارجية (WAN): تمثل شبكة الإنترنت.
- الجدار الناري (Firewall): تم استخدام جدار ناري من نوع PfSense لربط الشبكات الثلاث وتطبيق القواعد الأساسية لحماية الشبكة الداخلية.
- المصيدة (Honeypot): تم استخدام المصيدة Opencanary لدعمها بروتوكولات متعددة مثل SSH, FTP, HTTP. HTTPS وغيرها.
- نظام كشف ومنع التسلل Snort .

### 1-15-3 الاختبارات:

تم إجراء مجموعة من الاختبارات العملية لتقييم فعالية النظام في رصد محاولات الاختراق. تضمنت هذه الاختبارات:

- محاكاة الهجمات: تم تنفيذ هجمات متنوعة مثل هجمات القوة الغاشمة والتصيد على المصائد، مما سمح بتسجيل الأنشطة وتحليل سلوك المهاجمين.
- جمع البيانات: تم جمع البيانات من المصائد المختلفة، بما في ذلك تفاصيل الهجمات، عناوين IP، البروتوكولات المستخدمة، والأوامر المنفذة.
- تحليل البيانات: تم استخدام أدوات مثل Elastic Stack لتحليل البيانات المستخرجة من المصائد. تم إعداد استعلامات لتحليل الأنشطة غير الطبيعية وتحديد الأنماط الشائعة في الهجمات.

### 1-15-4 التحليل:

بعد جمع البيانات، تم تحليل النتائج باستخدام منهجيات متعددة:

- تحليل سلوك المهاجمين: تم دراسة الأنماط السلوكية للمهاجمين من خلال البيانات المستخرجة، مما ساعد في فهم استراتيجيات الهجوم وتطوير استراتيجيات دفاعية فعالة.

- تقديم التوصيات: بناءً على نتائج التحليل، تم تقديم توصيات لتحسين استراتيجيات الأمان في المؤسسات، بما في ذلك كيفية استخدام البيانات المستخرجة من المصائد لتطوير استراتيجيات استجابة فعالة.

## 1-15-5 الخلاصة:

تجمع منهجية البحث بين التصميم العملي، الاختبارات، والتحليل، مما يوفر إطارًا شاملاً لتطوير نظام إنذار أمني يعتمد على مصائد المخترقين. من خلال هذه المنهجية، تم تحقيق أهداف البحث وتقديم حلول عملية لتحسين الأمن السيبراني في الشبكات المحلية. إن النتائج المستخلصة من هذه الدراسة تسهم في تعزيز الفهم العام للتهديدات السيبرانية وتطوير استراتيجيات فعالة لمواجهتها.

# الفصل الثاني: الإطار النظري والدراسات السابقة

## 2-1 النظام الأمني المقترح:

### 2-1-1 مقدمة:

النظام الأمني المقترح هو نظام إنذار أمني متكامل يعتمد على تقنية مصائد المخترقين المتنقلة (Mobile Honeypots) لتحسين أمان الشبكات المحلية. يهدف هذا النظام إلى الكشف عن محاولات الاختراق في الوقت الفعلي من خلال إنشاء بيئات خادعة متنقلة تجذب المهاجمين وتقوم بتسجيل أنشطتهم. حيث يعتمد النظام على مجموعة من الأدوات والتقنيات المتقدمة، بما في ذلك:

- **مصائد المخترقين:** مثل Opencanary وSnort، والتي تعمل على جذب المهاجمين وتسجيل سلوكهم.
- **Elastic Stack:** الذي يتضمن Elasticsearch و Vector وKibana، ويستخدم لتحليل البيانات المجمعة من المصائد وعرضها بشكل مرئي.
- **جدار ناري (PfSense):** الذي يعمل على حماية الشبكة الداخلية من التهديدات الخارجية ويقوم بتطبيق قواعد أمان محددة.

يعمل النظام على جمع البيانات من المصيدة من أماكن مختلفة من الشبكة وتحليلها باستخدام Elastic Stack، مما يوفر رؤى قيمة حول الأنشطة الضارة. من خلال هذا النظام، يمكن للمؤسسات تحسين استراتيجيات الدفاع الخاصة بها، وتقليل معدل الإنذارات الخاطئة، وزيادة فعالية استجابة الحوادث.

### 2-1-2 بنية الشبكة:

تتكون بنية الشبكة المستخدمة في النظام الأمني المقترح من ثلاث مناطق رئيسية، وهي:

#### 2-1-2-1 الشبكة الداخلية (LAN):

- الوصف: تمثل الشبكة الداخلية (Local Area Network) الجزء الأساسي من البنية التحتية لتكنولوجيا المعلومات في المؤسسة. تحتوي هذه الشبكة على المخدمات الرئيسية وأجهزة العمل، بما في ذلك أجهزة الحواسب والأنظمة التي تستخدمها الفرق المختلفة.

- الدور:
  - ✓ توفر الشبكة الداخلية بيئة آمنة لتبادل المعلومات والبيانات بين الموظفين.
  - ✓ تتيح الوصول إلى الموارد الداخلية مثل قواعد البيانات والخدمات المختلفة.
  - ✓ يتم تطبيق قواعد أمان صارمة على هذه الشبكة، حيث يمكنها الوصول إلى جميع الشبكات الأخرى، ولكن فقط جهاز الإدارة داخل هذه الشبكة يمكنه الوصول إلى إعدادات الجدار الناري.

### 2-1-2-2 الشبكة المعزولة (DMZ):

- الوصف: تمثل الشبكة المعزولة (Demilitarized Zone) منطقة وسيطة بين الشبكة الداخلية وشبكة الإنترنت، حيث تحتوي هذه الشبكة على المخدمات المعرضة للوصول من الإنترنت، مثل خوادم الويب وخوادم البريد الإلكتروني.

- الدور:
  - ✓تعمل الشبكة المعزولة على توفير مستوى إضافي من الأمان، حيث يمكن للمستخدمين الخارجيين الوصول إلى الخدمات العامة دون الوصول المباشر إلى الشبكة الداخلية.
  - ✓يتم تطبيق قواعد أمان محددة على هذه الشبكة، حيث يمكنها الوصول إلى الشبكة الخارجية (الإنترنت) ولكن لا يمكنها الوصول إلى الشبكة الداخلية.

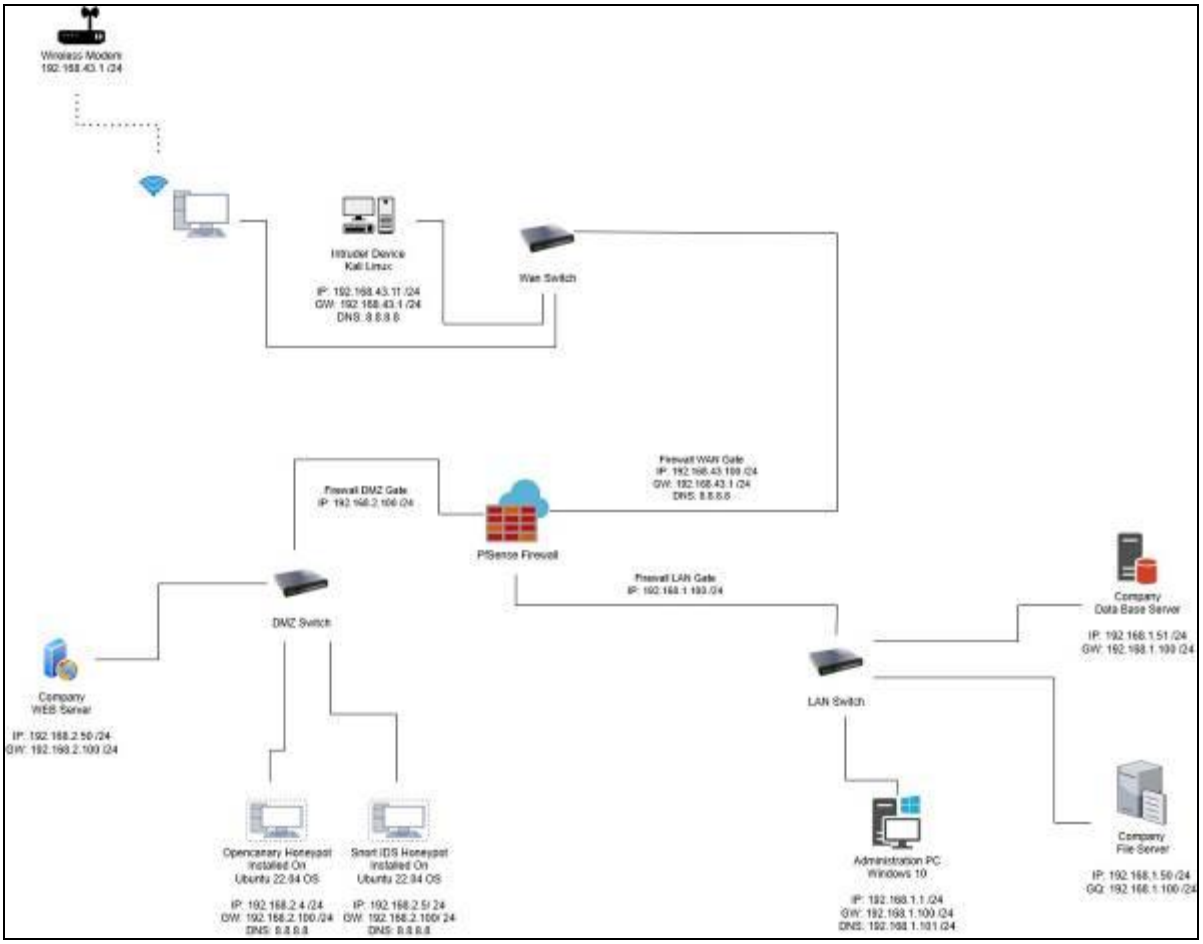
### 2-1-2-3 الشبكة الخارجية (WAN):

- الوصف: تمثل الشبكة الخارجية (Wide Area Network) شبكة الإنترنت. تعتبر هذه الشبكة هي البوابة التي تربط المؤسسة بالعالم الخارجي.

- الدور:
  - ✓ تتيح الشبكة الخارجية الوصول إلى المعلومات والخدمات عبر الإنترنت.
  - ✓ يمكن أن تكون مصدرًا للتهديدات، حيث يمكن للمهاجمين محاولة الوصول إلى الشبكة الداخلية من خلال هذه الشبكة.

✓ يتم تطبيق قواعد أمان صارمة على هذه الشبكة، حيث يمكنها الوصول إلى الشبكة المعزولة فقط، ولكن لا يمكنها الوصول إلى الشبكة الداخلية.

مما سبق يمكن توصيف البنية العامة للشبكة المقترحة كما هو موضح في الشكل التالي:




الشكل (1): البنية العامة للشبكة المقترحة

تتكون بنية الشبكة المستخدمة في النظام الأمني المقترح من ثلاث مناطق رئيسية: الشبكة الداخلية (LAN)، الشبكة المعزولة (DMZ)، والشبكة الخارجية (WAN). كل منطقة تلعب دورًا حيويًا في تعزيز الأمان السيبراني، حيث توفر هيكلًا متكاملاً لحماية البيانات والمعلومات الحساسة. من خلال تصميم هذه البنية، يمكن للمؤسسات تحسين استراتيجيات الدفاع الخاصة بها وتقليل المخاطر المرتبطة بالتهديدات السيبرانية.

## 2-2 الدراسات المرجعية:

### 2-2-1 الدراسة الأولى:

- عنوان الدراسة: SMASH: SDN-MTD Automated System with Honeypot Integration
- الباحثون: Nicola d'Ambrosioa, Claudio Listaa, Gaetano Perronea, Simon Pietro Romanoa
- سنة النشر: 2025.
- المؤسسة العلمية: University of Naples Federico II, Department of Electrical Engineering and Information Technology, Via Claudio 21, Naples, 80125, Naples, Italy
- الفكرة العامة من الدراسة:

تقديم إطار عمل يجمع بين الشبكات المعرفة برمجياً (SDN) وتقنيات الدفاع المتحرك (MTD) مع دمج مصائد المخترقين لتحسين الأمن السيبراني، حيث يهدف النظام إلى مواجهة التحديات المرتبطة بنشر المصائد التقليدية، مثل صعوبة التكوين اليدوي والتكاليف العالية، ويعتمد على تقنيات ديناميكية لتغيير مواقع المصائد وتكويناتها بشكل مستمر، مما يجعل من الصعب على المهاجمين اكتشافها أو استخدامها كنقطة انطلاق للهجمات. يتم توجيه المهاجمين إلى شبكات معزولة لتحليل سلوكهم، مما يساهم في جمع معلومات قيمة عن أساليب الهجوم. كما يوفر النظام آلية إدارة مرنة وقابلة للتوسع، مما يسمح بالنشر التلقائي والمراقبة في الوقت الحقيقي.

- النقاط الرئيسية:

✓ استخدام SDN لتسهيل إدارة المصائد.

✓ تقنيات MTD لتغيير المواقع والتكوينات بشكل ديناميكي.

✓ تحسين جمع بيانات الهجمات وتحليلها.

✓ تعزيز الأمن في الشبكات الكبيرة والمعقدة.

## 2-2-2 الدراسة الثانية:

- عنوان الدراسة: Dynamic Honeypot Conversion for Enhanced IoT Security
- الباحثون: Daniel Commey, Matilda Nkoom, Sena Hounsinou, Garth Crosby.
- سنة النشر: 2025.
- المؤسسة العلمية: Metropolitan State University.
- الفكرة العامة من الدراسة:

تناقش هذه الدراسة تحديات أمان إنترنت الأشياء (IoT) وتقدم نظامًا جديدًا يسمى BHICS (Blockchain-enabled Honeypot IoT Conversion System). يهدف النظام إلى تحويل العقد العادية في شبكات IoT إلى مصائد ديناميكية بناءً على مستوى التهديد المكتشف. يعتمد النظام على تقنيات تعلم الآلة الخفيفة لتحديد التهديدات وتغيير إعدادات المصائد بشكل تلقائي. كما يستخدم تقنية البلوكشين لتسجيل الأحداث الأمنية بشكل موثوق. أظهرت التجارب أن النظام يقلل من معدلات اختراق العقد بنسبة كبيرة، مع الحفاظ على أداء مستقر في الشبكات الكبيرة. يوفر النظام حلاً فعالًا وقابلًا للتوسع لتحسين أمان شبكات IoT.

- النقاط الرئيسية:
  - ✓ تحويل العقد العادية إلى مصائد ديناميكية.
  - ✓ استخدام تعلم الآلة لتحليل التهديدات.
  - ✓ تسجيل الأحداث الأمنية باستخدام البلوكشين.
  - ✓ تحسين أمان الشبكات الكبيرة مع تقليل التكاليف.

## 2-2-3 الدراسة الثالثة:

- عنوان الدراسة: Multi-Domain Moving Target Defense for Resilient Security in Power Cyber- Physical Systems: A Review
- الباحثون: He Wu
- سنة النشر: 2025.
- المؤسسة العلمية: School of Electric Power, Shenyang Institute of Engineering, Shenyang, China

- الفكرة العامة من الدراسة:

تستعرض هذه الدراسة كيفية تحسين أداء مصائد المخترقين في بيئات إنترنت الأشياء باستخدام تقنيات تعلم الآلة. يتم التركيز على المصائد الديناميكية التي تتكيف مع التهديدات المتغيرة والمصائد التكيفية التي تستجيب للهجمات في الوقت الحقيقي. أظهرت النتائج أن تقنيات تعلم الآلة، مثل النماذج الإشرافية (Supervised Learning)، تحسن دقة الكشف بنسبة تصل إلى 96%. كما توفر المصائد التكيفية إدارة أفضل للموارد وتقليل الإنذارات الكاذبة. ومع ذلك، تواجه هذه الأنظمة تحديات مثل الطلب العالي على الموارد وصعوبة الاختبار في البيئات الواقعية.

- النقاط الرئيسية:

✓ تحسين الكشف عن التهديدات باستخدام تعلم الآلة.

✓ تقليل الإنذارات الكاذبة وتحسين إدارة الموارد.

✓ التركيز على المصائد الديناميكية والتكيفية.

✓ تحديد الفجوات البحثية وتقديم توصيات لتحسين الأنظمة المستقبلية.

# الفصل الثالث: تصميم النظام الأمني

## 3-1 اعداد بيئة العمل:

### 3-1-1 الجدار الناري PfSense:

pfSense هو جدار ناري مجاني ومفتوح المصدر يستند إلى نظام التشغيل FreeBSD. وهو أحد أكثر أنظمة جدار الحماية شيوعًا في العالم، وهو مستخدم من قبل المستخدمين العاديين والشركات الصغيرة والمتوسطة والكبيرة.

ويمكن استخدام pfSense للحماية من التهديدات الأمنية المختلفة، بما في ذلك:

الهجمات الاستغلالية، البرمجيات الضارة، برامج التجسس، هجمات التصيد الاحتيالي، والهجمات DDoS

حيث يوفر pfSense العديد من الميزات التي تساعد على حماية الشبكة، بما في ذلك:

تصفية حزم IP، تصفية المحتوى، تحليل السلوك، ، VPN، إدارة حركة المرور كما يتميز pfSense بعدة مزايا عن غيره من الجدران النارية، بما في ذلك:

- مجاني ومفتوح المصدر.
- قابل للتخصيص بسهولة.
- يدعم الأجهزة المادية والافتراضية.
- لديه مجتمع دعم كبير.

ويعد pfSense خيارًا رائعًا لأي شخص يبحث عن جدار ناري قوي وقابل للتخصيص. فهو خيار جيد بشكل خاص للمستخدمين المنزليين والشركات الصغيرة، حيث يوفر ميزات وأداءً متفوقًا على الجدران النارية التجارية، وبشكل عام، يعد pfSense أداة قوية يمكن استخدامها لحماية الشبكة من مجموعة متنوعة من التهديدات.

#### 3-1-1-1 المتطلبات الأساسية لـ pfSense:

- معالج 64 بت.
- ذاكرة وصول عشوائي (RAM) بسعة 1 جيجابايت على الأقل.
- مساحة تخزين 8 جيجابايت على الأقل.
- بطاقة شبكة واحدة على الأقل.

### 3-1-2 خطوات تثبيت pfSense:

**أولاً:** تحميل ملف ISO الخاص بـ pfSense من الموقع الرسمي:

https://www.pfsense.org/download/

حيث يتم اختيارالنسخة المناسبة للجهاز الذي سوف نحمل عليه الجدار الناري، وهنا قمنا باختيار النسخة pfSense-CE-2.7.2-RELEASE-amd64

**ثانياً:** انشاء وسيط تثبيت قابل للتشغيل باستخدام USB Pen Drive. يمكن إنشاء وسائط تثبيت قابلة للتشغيل لـ pfSense باستخدام برنامج مثل Etcher أو Rufus.

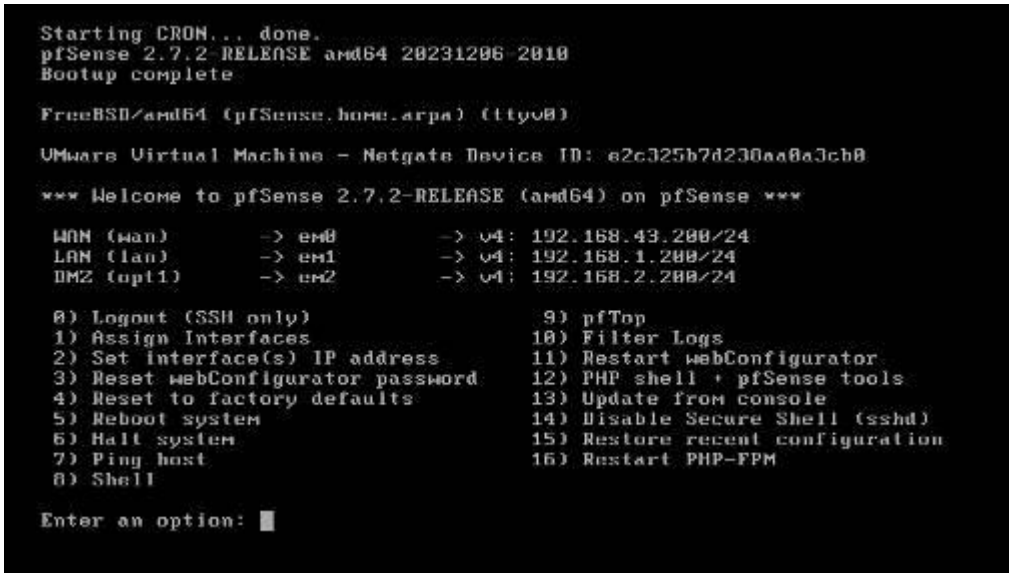
**ثالثاً:** اعادة تشغيل الجهاز واختيار الاقلاع من واسطة التخزين التي تم اعدادها في الخطوة السابقة.

**رابعاً:** نقوم باتباع التعليمات التي تظهر على الشاشة لتثبيت pfSense لحين اكتمال التثبيت.

### 3-1-3 مراحل تنصيب PfSense:

- جمع متطلبات الأجهزة.
- التأكد من أن الجهاز المادي يلبي متطلبات الأجهزة الأساسية لـ pfSense. حيث يمكن العثور على متطلبات الأجهزة المحددة على موقع ويب الرسمي لـ pfSense.
- اختيار لغة التثبيت.
- تعيين اسم المستخدم وكلمة المرور لحساب المسؤول.
- تعيين إعدادات الشبكة.
- بعد إكمال التثبيت، سيتم إعادة تشغيل الجهاز المادي مرة أخرى. بعد إعادة التشغيل، سيتم بدء تشغيل pfSense.

وبعد الاقلاع تظهر واجهة الجدار الناري على الجهاز الفيزيائي المحمل عليه كما في الشكل التالي:

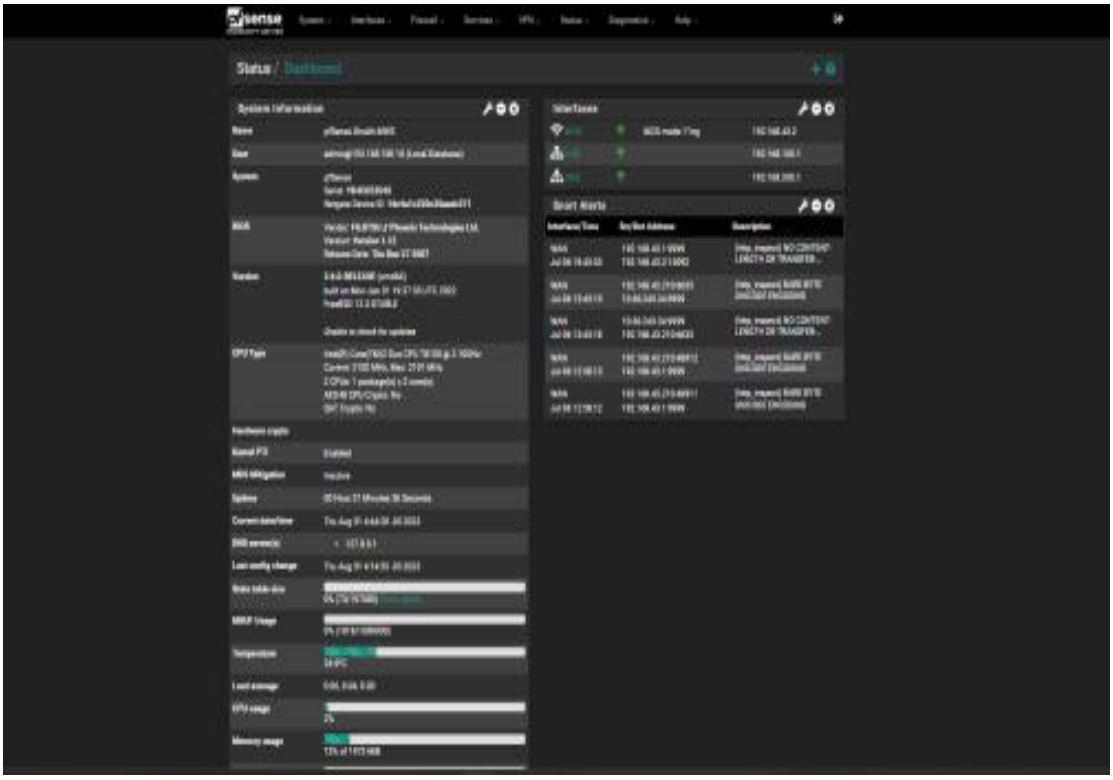



الشكل (2) الواجهة الرئيسية للجدار الناري PfSense

يظهر هنا امكانية الاعداد بالاعتماد على واجهة البرنامج والقيام بادخال الاختيارات بشكل متسلسل ولكن القيام بالضبط من خلال واجهة الويب هو أبسط ويوفر امكانية أكبر للادارة والصيانة ومراقبة الأخطاء، حيث يمكن الآن تسجيل الدخول إلى واجهة مستخدم pfSense عبر الويب باستخدام اسم المستخدم وكلمة المرور اللذين تم تحديدهما أثناء التثبيت.


الشكل (3) واجهة تسجيل الدخول للجدار الناري PfSense

وبعد تسجيل الدخول سيتم الانتقال إلى الواجهة الرئيسية لـ PfSense كما هو موضح في الشكل التالي:



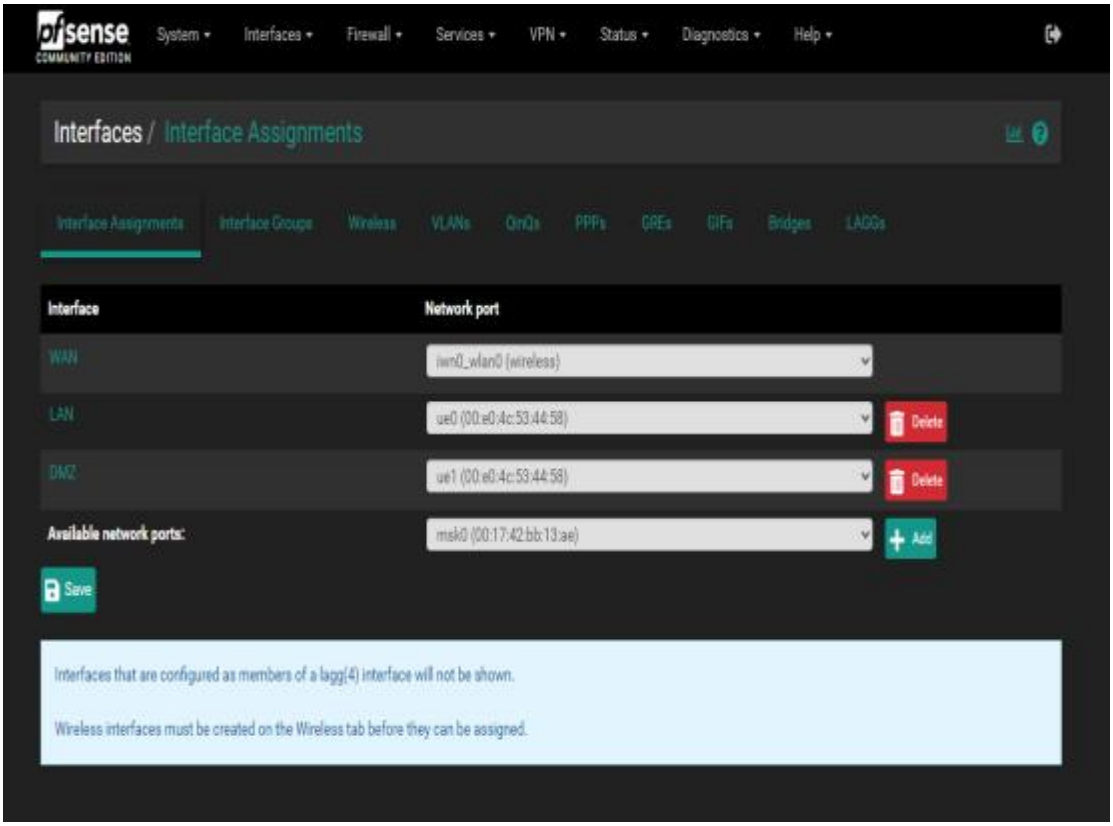
الشكل (4) واجهة ويب الرئيسية للجدار الناري PfSense

حيث تظهر هذه الواجهة العديد من المعلومات عن حالة الجدار الناري مثل اسم الجدار الناري، نسخة نظام تشغيل الجدار الناري، نسخة BIOS للجهاز المحمل بـ PfSense، نوع المعالج، حجم الذاكرة الكلي/ المستخدم، درجة حرارة المعالج، معدل التحميل للمعالج ودرجة استخدامه، حجم ملف الصفحات المستخدم، بطاقات الشبكة الموصولة بالجهاز، عناوين ip للشبكات التي تم تنصيبها على الجدار الناري، وغير ذلك الكثير.

حيث يمكن من الواجهة Interfaces اختيار Interface Assignments ومنها نستطيع أن نحدد عدد اشبكات التي سيقوم الجدار الناري بالربط بينها وهنا نحتاج ثلاث شبكات هي:

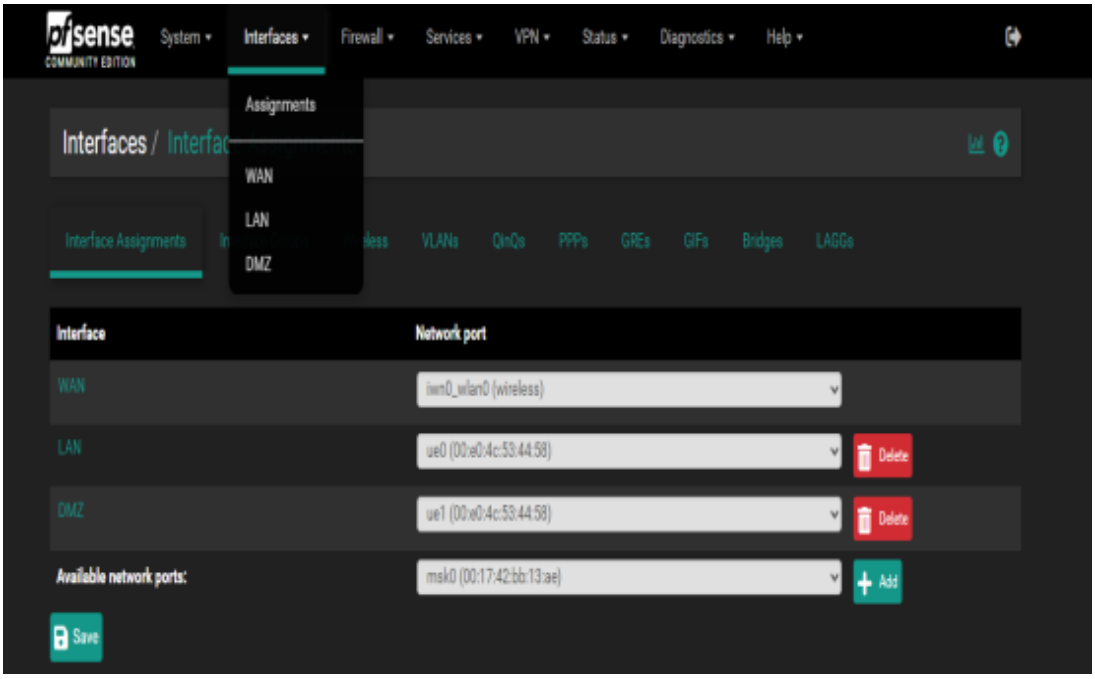
- ✓ الشبكة الداخلية LAN
- ✓ الشبكة Demilitarized Zone DMZ
- ✓ الشبكة الخارجية وهي شبكة الانترنت WAN

ومن خلال النقر على ADD واختيار اسم الشبكة وبطاقة الشبكة المرتبطة معها كما هو موضح في الشكل التالي:

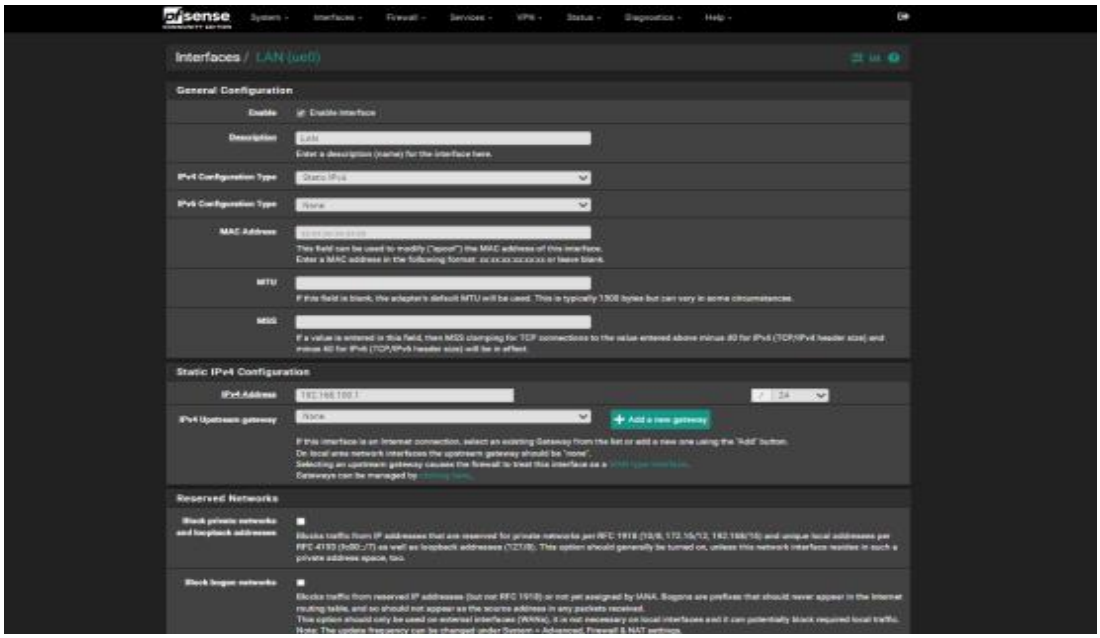



الشكل (5) واجهة عرض بطاقات الشبكة الخاصة بالجدار الناري PfSense

وتظهر الشبكات التي تم اعدادها ضمن التبويب Interfaces كما هو موضح في الشكل التالي:



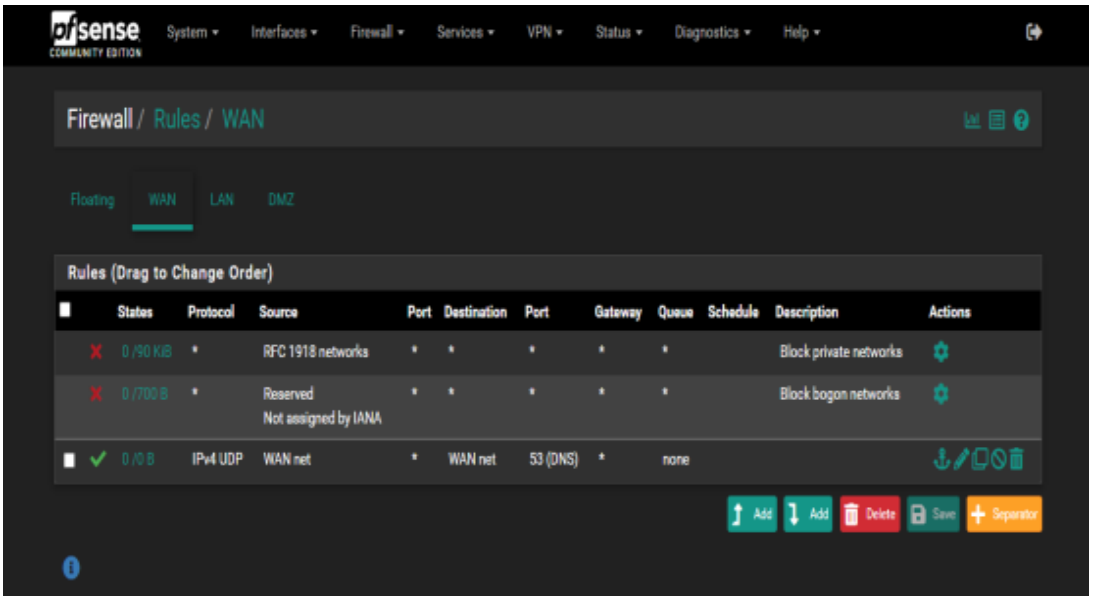
ومن خلال النقر على الشكبة الداخلية مثلاً يتم الانتقال إلى صفحة الاعدادات الخاصة بها والتي من خلالها يمكن اعداد عنوان ip لهذه الشبكة والقناع وتفعيل خدمة DHCP على منفذ بطاقة الشبكة المرتبطة فيها وتحديد مجال العناوين وغير ذلك، كما هو ظاهر بالشكل التالي:




الشكل (6) واجهة اعادات القواعد الخاصة بالجدار الناري PfSense

وهكذا بالنسبة لبقية الشبكات.

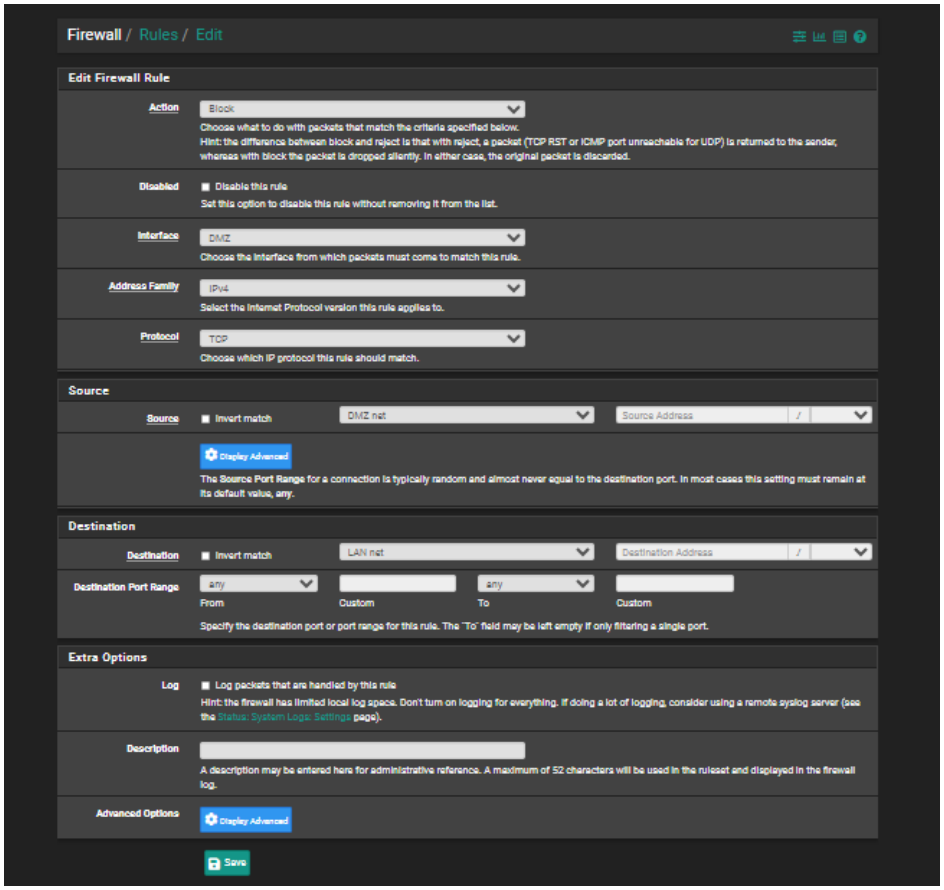
وبالانتقال إلى التبويب Firewall واختيار Rules من القائمة المنسدلة يمكن البدء بانشاء القواعد بعد اختيار الشبكة المطلوبة أولاً كما يلي:



القواعد العامة هي كالتالي:

- ✓ الشبكة LAN تستطيع الوصول إلى الشبكة DMZ والشبكة WAN.
- ✓ الشبكة DMZ تستطيع الوصول إلى الشبكة WAN ولاتستطيع الوصول إلى الشبكة الداخلية LAN.
- ✓ الشبكة الخارجية WAN تستطيع الوصول إلى الشبكة DMZ فقط.

وكمثال نوضح هنا كيفية اعداد قاعدة Rule بحيث أن الشبكة DMZ لاتستطيع الوصول إلى الشبكة الداخلية LAN :

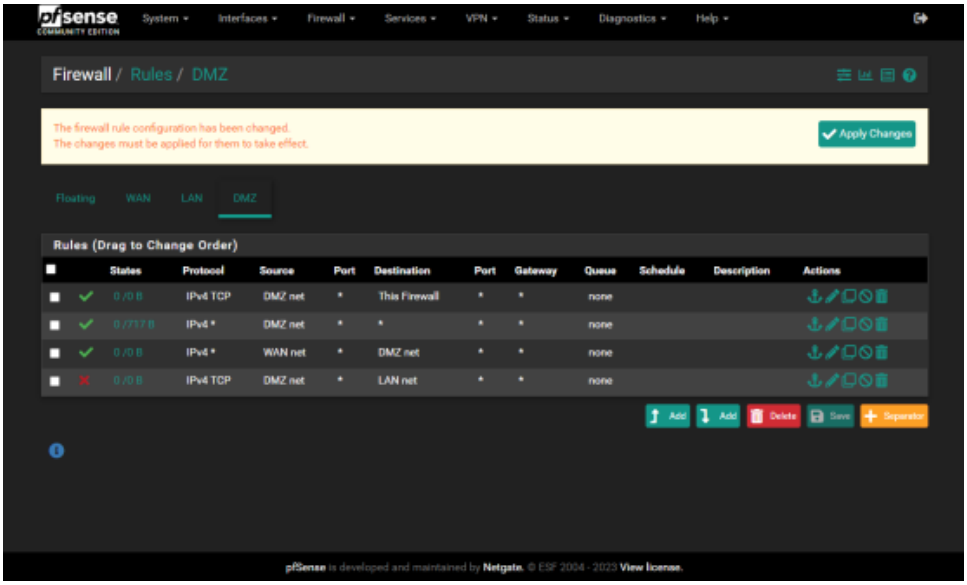



هنا قمنا بتحديد البرامترات التالية:

- الخيار Block لحجب كافة الطلبات.
- منفذ بطاقة الشبكة وهو DMZ.
- اصدار الـip وهنا اخترنا Ipv4.
- البروتوكول الذي نريد منعه وهنا اخترنا البروتوكول TCP.

- العنوان المصدر وهنا تم اختيار كافة عناوين الأجهزة التي تنتمي إلى الشبكة DMZ أي الخيار DMZ net.
- عنوان الوجهة تم اختيار مجال عناوين الشبكة الداخلية LAN net.
- تحديد المنافذ المشمولة بالحجب وهنا تم اختيار جميع المنافذ Any من المصدر والمآل.
- يمكن تفعيل خيار التسجيل لتسجيل جميع الاحداث المتعلقة بهذه القاعدة.
- أخيراً نضغط على SAVE.

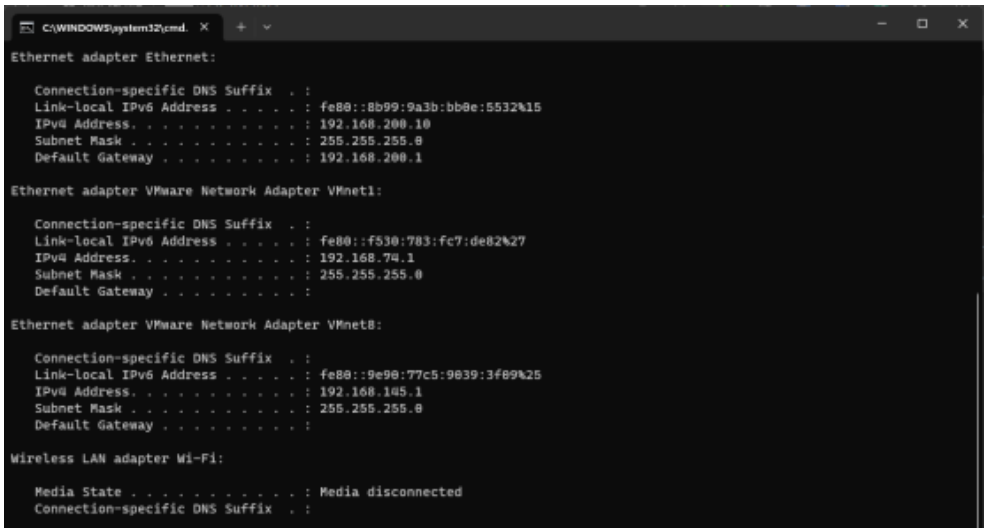
وهكذا بالنسبة لبقية القواعد Rules المراد تطبيقها على هذه الشبكة وبقية الشبكات، حيث تظهر القواعد المطبقة كما هو موضح بالشكل التالي:



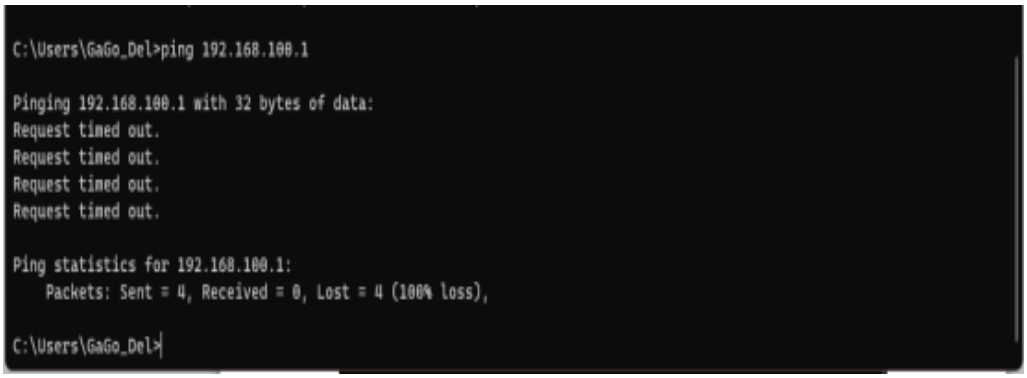

الشكل (7) واجهة اعادات القواعد للشبكة LAN الخاصة بالجدار الناري PfSense

نقوم باختبار الوصول من الشبكة DMZ إلى الشبكة الداخلية فنجد أنه لم يعد بالامكان الوصول كما يظهر في الشكل التالي:

نلاحظ هنا أن الجهاز ينتمي إلى الشبكة 192.168.2.0 أي الشبكة DMZ




حيث نجد أنه لم يعد هناك امكانية للوصول إلى الشبكة 192.168.1.0 أو الشبكة الداخلية كما هو واضح في الشكل التالي:




يمكن أن يتم تنصيب نظام كشف ومنع التدخل Snort ضمن الجدار الناري ولكن لم يتم ذلك لأننا قمنا بتنصيب هذا النظام على جهاز مستقل، وتجدر الاشارة إلى ان الجدار الناري PfSense يؤمن خدمات التوجيه بين الشبكات الثلاث وبالتالي لتجربة مخبرية تمثل النظام الأمني المقترح فيمكن الاستغناء عن جهاز التوجيه أو الراوتر.

وبهذا نكون قد قمنا بتنصيب الجدار الناري PfSense بنجاح.

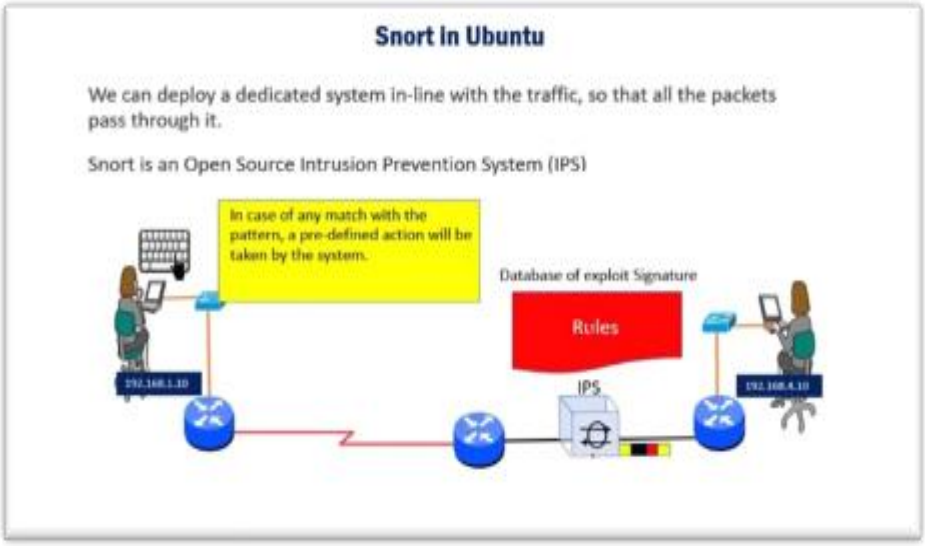
### 3-1-2 المصيدة Snort:

Snort مصيدة مفتوحة المصدر لكشف ومنع الاختراقات في الشبكات. وهو نظام كشف ومنع الاختراقات (IDS/IPS) يمكن استخدامه لتحليل حركة المرور على الشبكة واكتشاف الأنشطة المشبوهة.

يوفر Snort مجموعة واسعة من الميزات، بما في ذلك:

- ✓ تحليل حركة المرور على الشبكة في الوقت الفعلي
- ✓ اكتشاف الأنشطة المشبوهة باستخدام قواعد مخصصة
- ✓ تسجيل حركة المرور على الشبكة
- ✓ التكامل مع أنظمة أمنية أخرى
- ✓ يمكن استخدام Snort على مجموعة متنوعة من الأنظمة، بما في ذلك Linux و Windows و macOS.

كمحصلة إذا كنا نبحث عن نظام IDS/IPS قوي وقابل للاعداد والتخصيص، فإن Snort هو خيار رائع.




الشكل (8) مخطط عمل المصيدة Snort

### 3-1-2-1 خطوات التثبيت:

**أولاً:** يتم التثبيت على حهاز محمل بنظام تشغيل UBUNTU 22.04.

**ثانياً:** نقوم بتحديث مكتبات النظام من خلال الأوامر التالية:

```
$sudo apt-get update
$sudo apt-get upgrade
```

**ثالثاً:** نقوم بتحميل وتنصيب المكتبات والاعتماديات اللازمة لعمل snort من خلال الأمر التالي:

```
sudo apt install build-essential libpcap-dev libpcre3-dev libnet1-dev zlib1g-dev luajit hwloc
libdnet-dev libdumbnet-dev bison flex liblzma-dev openssl libssl-dev pkg-config libhwloc-dev
cmake cpputest libsqlite3-dev uuid-dev libcmocka-dev libnetfilter-queue-dev libmnl-dev
autotools-dev libluajit-5.1-dev libunwind-dev
```

**رابعاً:** تحميل وتنصيب snort كما يلي:

**Download and install latest version of the Snort DAQ (Data Acquisition library LibDAQ)**

```
$sudo mkdir snort-source-files
$cd snort-source-files
$sudo git clone https://github.com/snort3/libdaq.git
$cd libdaq
$sudo ./bootstrap
$./configure
$make
$sudo make install
```

If needed

```
$sudo apt install git
```

**خامساً:** تنصيب Google Thread-caching malloc والتي تفيد في تأمين واجهات برمجة التطبيقات لتخصيص الذاكرة الديناميكية، حيث تقوم بإدارة الذاكرة بشكل أفضل من طلبات الذاكرة الأولية من خلال توفير العديد من التحسينات.

**Download and install google's thread-caching malloc, Tcmalloc, a memory allocator optimized for high concurrency situations**

Tcmalloc -> Thread Cashing Malloc

Malloc -> Memory ALLOCation

Go back to snort-source-files with **cd ..**

```
wget wget https://github.com/gperftools/gperftools/releases/download/gperftools2.8/gperftools-2.8.tar.gz
tar xzf gperftools-2.8.tar.gz
cd gperftools-2.8/
./configure
sudo make install
```

gperftools -> Google performance tools: Fastest malloc

**سادساً:** نقوم بتنفيذ الأوامر التالية لاكمال تنصيب snort كما في الشكل التالي:

**Install Snort 3**

Go back to snort-source-files with **cd ..**

```
git clone git://github.com/snortadmin/snort3.git
cd snort3/
./configure_cmake.sh --prefix=/usr/local --enable-tcmalloc
```

To configure and activate tcmalloc at compile time

```
cd build
make
sudo make install
```

Navigate to the build directory to compile and install Snort 3

```
sudo ldconfig
snort -V
```

Update shared libraries

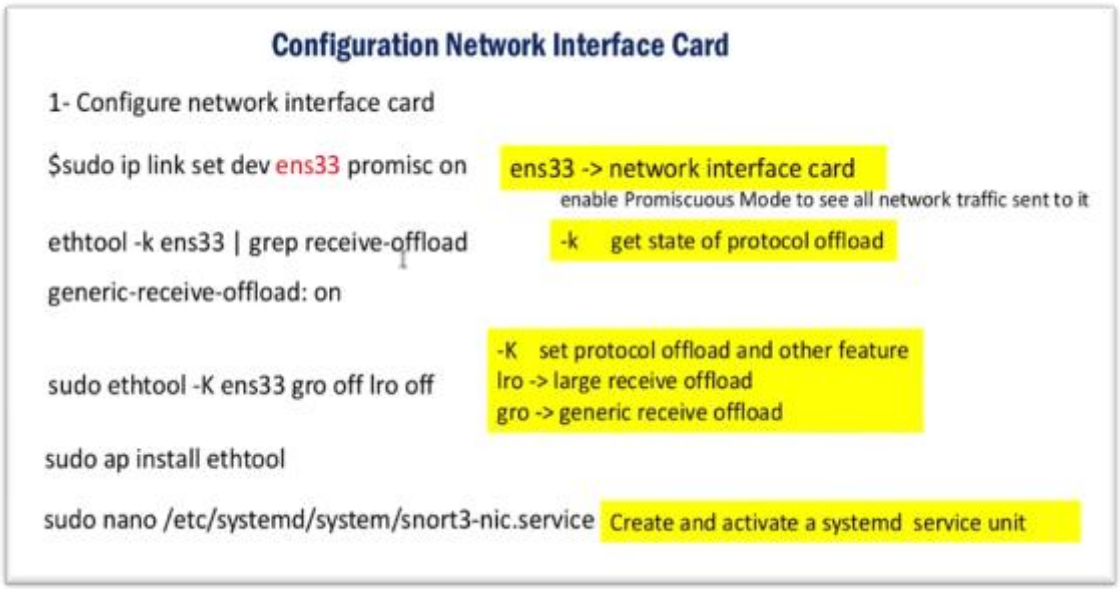
Check the installation of Snort 3

نلاحظ بعد تنفيذ الأمر sudo make install بدء عملية التنصيب.

**سابعاً:** بعد انتهاء عملية التنصيب يمكن التحقق من نسخة snort التي تم تنصيبها من خلال الأمر التالي:

snort -V

نقوم الأن باعداد بطاقة الشبكة لتعمل في النمط promiscuous mode كما يلي:




وللحفاظ على هذه الاعدادات بشكل دائم بحيث يتم تطبيقها عند كل عملية اعادة تشغيل للنظام linux UBUNTU نقوم بتحرير الملف snort3-nic.service باستخدام الأمر التالي:

sudo nano /etc/systemd/system/snort3-nic.service

وتعديل قيمة الخاصية promisc لتصيح true.

نقوم الأن باعادة تحميل وتفعيل الخدمة snort3-nic.service باستخدام الأوامر التالية:

Systemctl daemon-reload

Systemctl enable –now snort3-nic.service

بعد ذلك نقوم ملف الاعدادات الخاص بـ Snort كما يلي:

sudo nano /usr/local/etc/snort/snort.lua

نقوم بتعديل عنوان الشبكة HOME حيث ندخل عنوان ip الجهاز الذي تم تنصيب snort عليه مع القناع، ونحدد الشبكة الخارجية بأنه أي عنوان ip خارج عنوان الشبكة المحلية.

وبهذا نكون قد انتهينا من تنصيب snort بالكامل، ونحتاج الأن لأن نقوم بانشاء مجلد rules ضمن المسار التالي:

```
cd /usr/local/etc/
Sudo mkdir rules
```

نقوم الأن بانشاء ملف rule ضمن المجلد rules نقوم بتسميته باسم icamp.rules على سبيل المثال، ونكتب ضمنه القاعدة التالية:

```
Alert icmp any any -> $HOME_NET any (msg:"The machine with this ip is performing ping command"; sid:1000001; rev:1;)
```

بعد ذلك نقوم بتشغيل snort وذلك بتحديد ملف الاعداد المراد اعتماده وملف القواعد الذي نريد تطبيقه باستخدام الأمر التالي:

```
Sudo snort –c /usr/local/etc/snort/snort.lua  -R /usr/local/etc/rules/icamp.rules –I ens33 –A alert_fast –s 65535 –k none
```

حيث يحدد الأمر السابق ملف الاعدادات الخاص بـ Snort ملف القواعد المعتمد ويحدد بطاقة الشبكة – في حالتنا هنا هي ens33 – ويحدد بأنه لايجب اغفال أي بيانات متراسلة مهما كان طولها والحجم الاقصى هو 65535 بايت، أما القاعدة المستخدمة فتفيد بأنه أي جهاز خارج الشبكة المحلية الخاصة بنا يقوم بتنفيذ أمر ping إلى أي جهاز ضمن الشبكة الخاصة بنا سيتم توليد تنبيه بالرسالة المذكورة وبعرض عنوان ip الخاص بالجهاز الهدف وزالجهاز المآل مع ذكر التاريخ والتوقيت.

يمكن اختيار عدة طرق ضمن ملف القواعد مثل alert، log، pass، activate.

مثال عن قاعدة أخرى يمكن اضافة القاعدة التالية ضمن ملف القواعد السابق كما يلي:

```
Alert tcp $HOME_NET -> any any (content:www.facebook.com; msg:"Hey!Some one is visiting facebook.com at this time"; sid:10000; rev:1;)
```

حيث يفيد هذا الأمر في اظهار تنبيه عندما يقوم أي مستخدم بزيارة موقع Facebook من أي جهاز ضمن شبكتنا الداخلية، ويتم التنفيذ كما في الحالة السابقة.

وبهذا نكون قد قمنا بتنصيب snort وتجربته بنجاح.

### 3-1-3 المصيدة opencanary:

وهي مصيدة مفتوحة المصدر مصممة لجذب المتسللين وتسجيل نشاطهم. حيث تعمل عن طريق محاكاة مجموعة كبيرة من البروتوكولات بحيث تكون جذابة بشكل كبير للمهاجمين، وعندما يقوم المهاجم بمحاولة الدخول إلى هذه المصيدة فهي تقوم بتسجيل نشاطه.

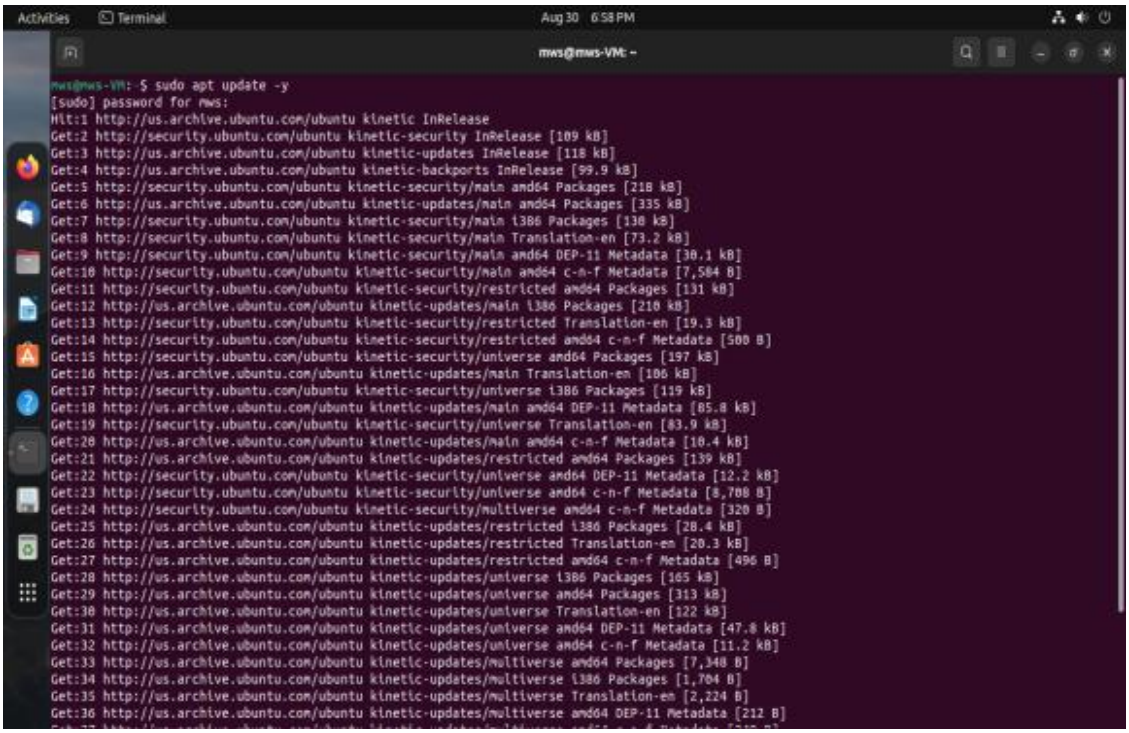
ويمكن استخدام Opencanary لجمع المعلومات عن المتسللين، بما في ذلك بروتوكولات الشبكة التي يستخدمونها، وأدوات القرصنة التي يستخدمونها، وأهدافهم، حيث يمكن استخدام هذه المعلومات لتحسين أمان الشبكة.

### 3-1-3-1 خطوات تثبيت Opencanary:

لتنصيب واعداد Opencanary على جهاز افتراضي يعمل بنظام التشغيل Linux توزيعة Ubuntu 22.04، نتبع الخطوات التالية:

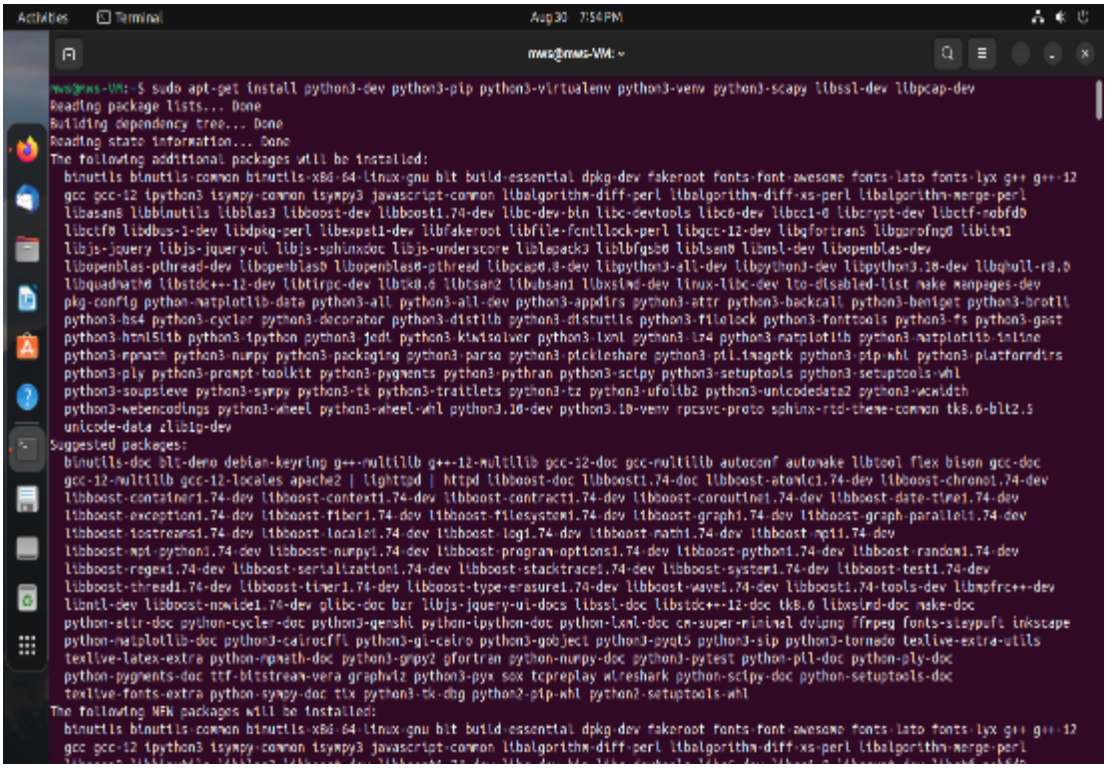
**أولاً:** القيام بتحديث مكتبات النظام:

sudo apt-get update



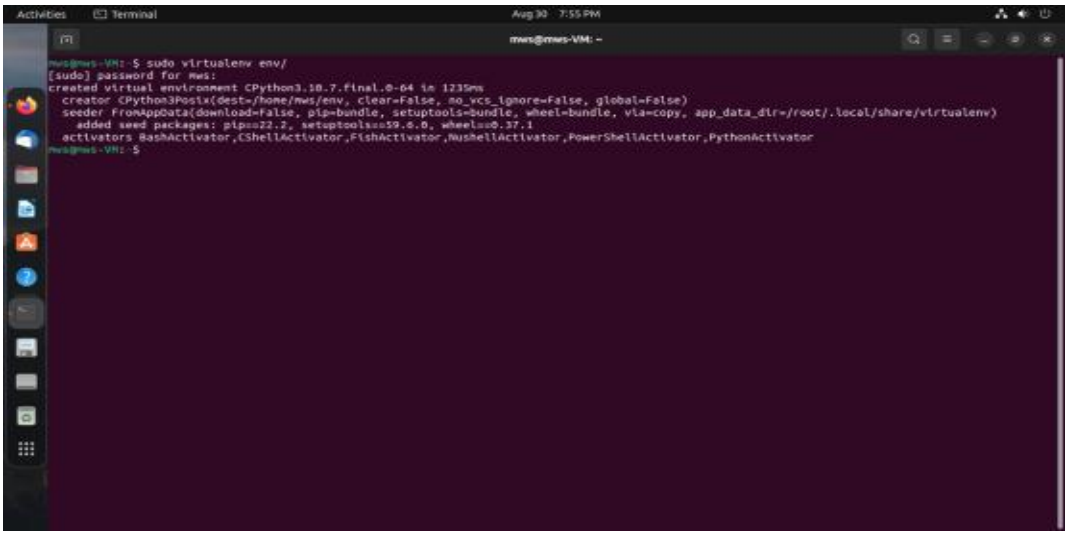
**ثانياً:** تثبيت المكتبات والاعتماديات التالية:

sudo apt-get install python3-dev python3-pip python3-virtualenv python3-venv python3-scapy libssl-dev libpcap-dev



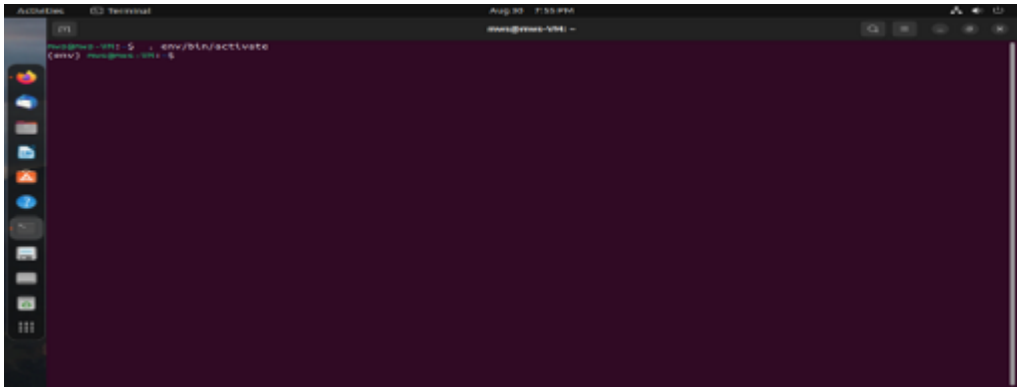
**ثالثاً:** انشاء بيئة افتراضية ضمن المجلد env تفيد هذه البيئة الافتراضية بتكوين حاوية أو Container لجميع المجلدات التي سوف تستخدم لتثبيت موديولات بايثون اللازمة لعمل Opencanary باستخدام الأمر:

/virtualenv env



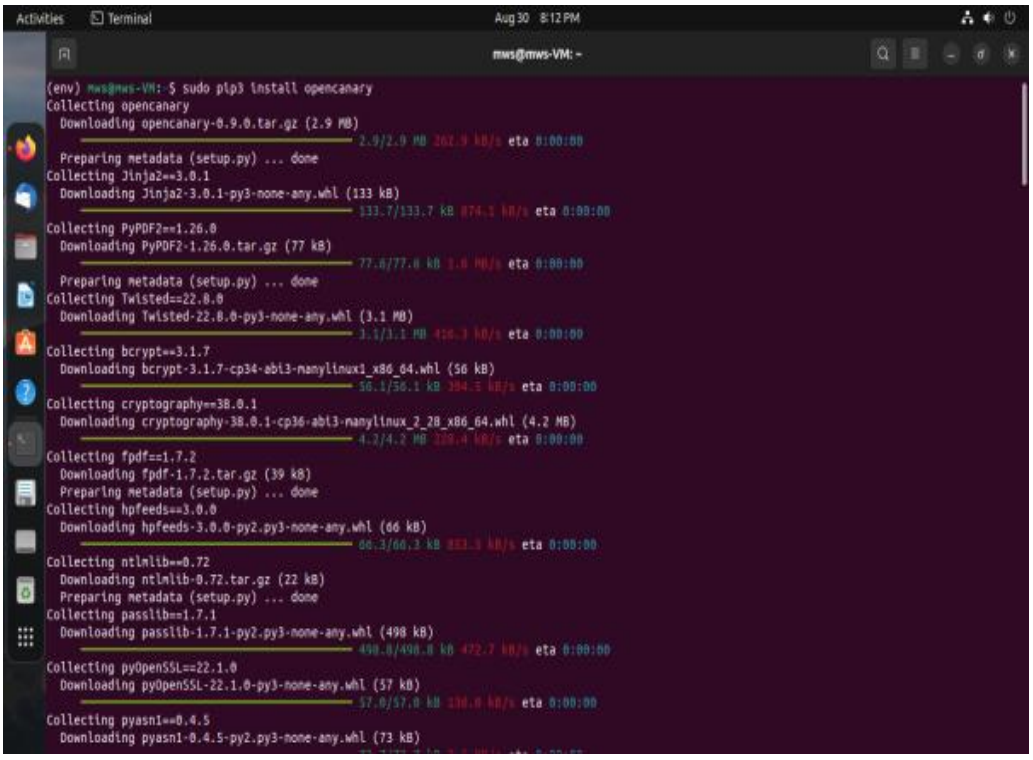
**رابعاً:** تفعيل البيئة الافتراضية باستخدام الأمر:

. env/bin/activate



**خامساً:** تثبيت OpenCanary باستخدام الأمر:

pip3 install opencanary




الشكل (9) بدء عملية تثبيت المصيدة Opencanary

**سادساً:** اعداد Opencanary:

nano opencanaryd/opencanary.conf

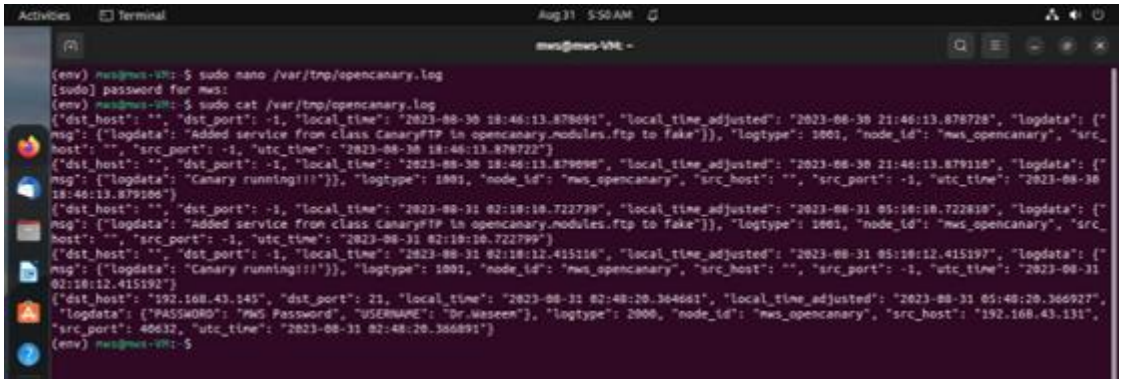
الشكل (10) ملف اعدادات المصيدة Opencanary

في ملف الاعدادات، نقوم بتغيير القيم التالية:

- server_address: عنوان IP أو اسم المجال للجهاز الافتراضي الذي يعمل عليه OpenCanary.
- server_port: المنفذ الذي سيستمع عليه Opencanary.
- log_file: مسار ملف السجل الذي سيتم استخدامه لتخزين نشاط المتسللين.

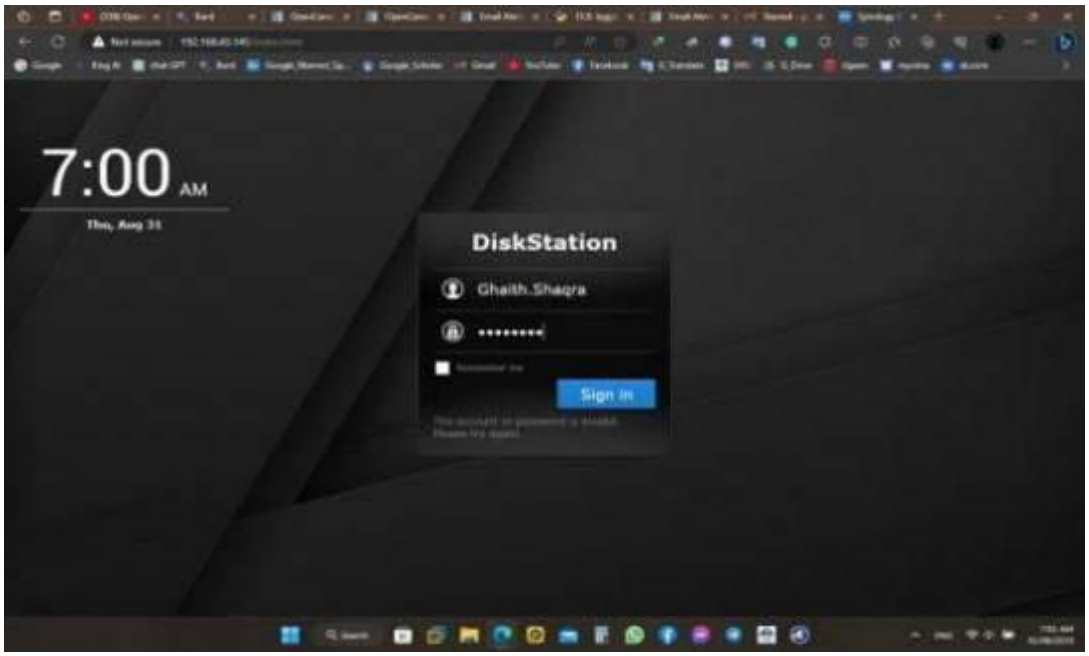
**سابعاً:** تشغيل Opencanary بتنفيذ الأمر:

Opencanaryd –start



الشكل (11) بدء تشغيل المصيدة Opencanary

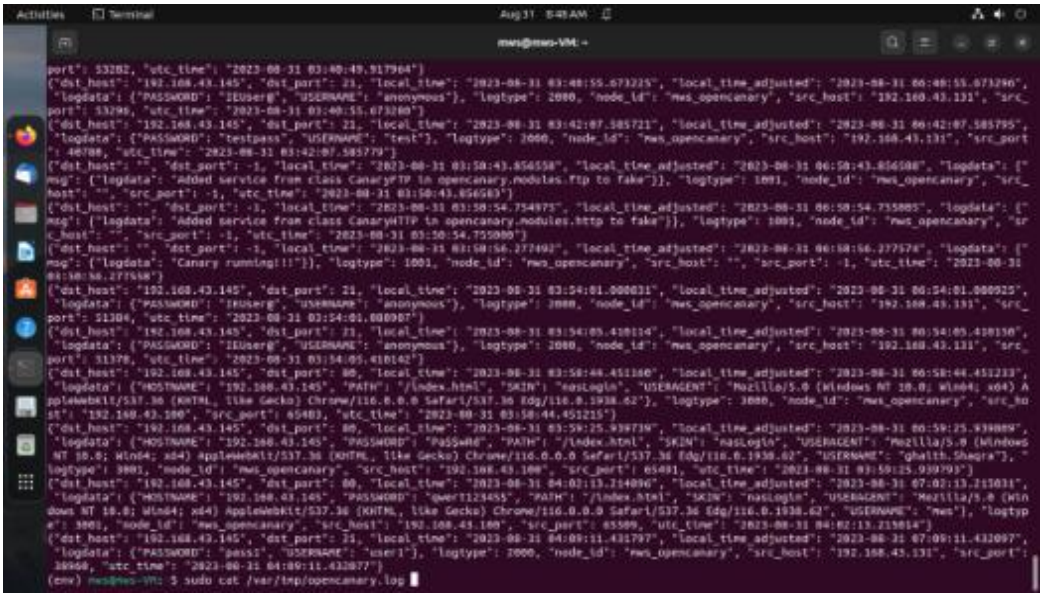
يمكن اختبار تثبيت Opencanary عن طريق محاولة الوصول إلى موقع الويب الخاص به من جهاز آخر. إذا تم اعداد Opencanary بشكل صحيح، فسيتم تسجيل نشاط المهاجم وحفظه في ملف السجل.




الشكل (12) واجهة الويب للمصيدة Opencanary

نقوم بعرض سجل الأحداث في Opencanary من خلال تنفيذ الأمر:

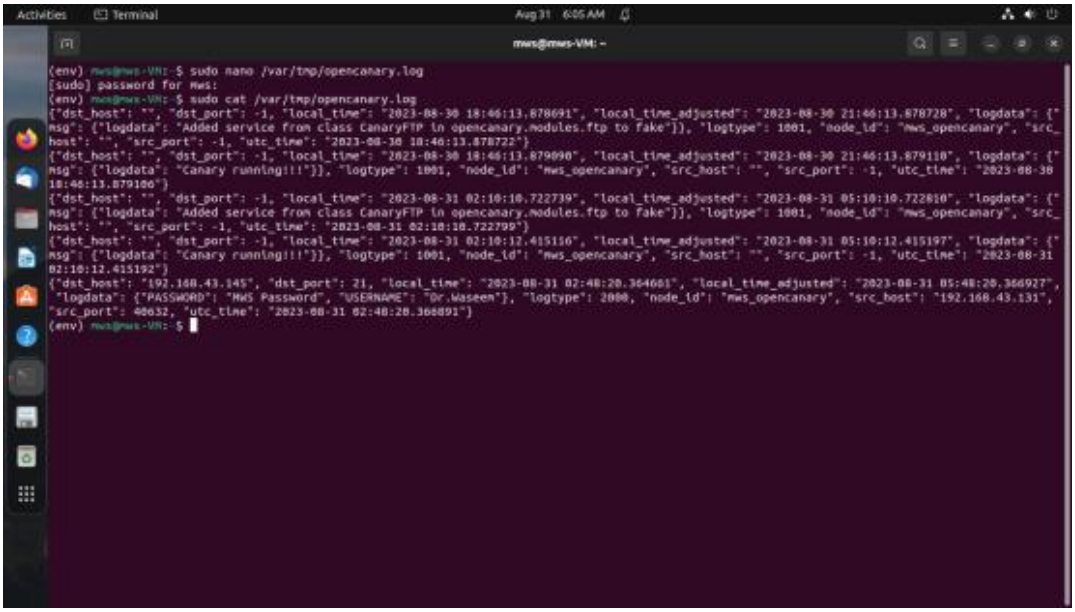
sudo cat /var/tmp/opencanary.log



الشكل (13) ملف تسجيلات المصيدة Opencanary

ويمكن أيضاً الوصول إلى خدمة ftp من خلال جهاز Kali Linux كما هو موضح بالشكل التالي:

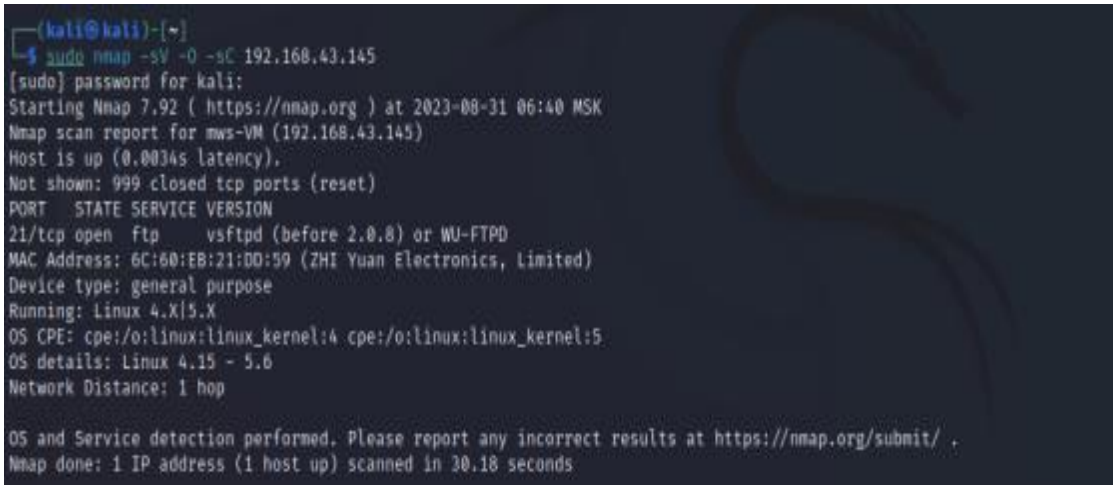

ونلاحظ تسجيل هذا النشاط ضمن ملف التسجيلات opencanary.log كما هو موضح في الشكل التالي:




تجدر الاشارة إلى أن هوية موضع الجذب opencanary يمكن التعديل عليها ضمن ملف الاعدادات opencanary.conf ففي الحالة الافتراضية وباستخدام الأمر:

nmap –sV –O –sC 192.168.2.4

والذي هو عنوان ip الخاص بالمصيدة، حيث يظهر لدينا الخرج التالي:




ولكن يمكن اجراء تعديلات على شكل وهوية المصيدة بحيث تأخذ هوية أحد المخدمات التالية:

## Services

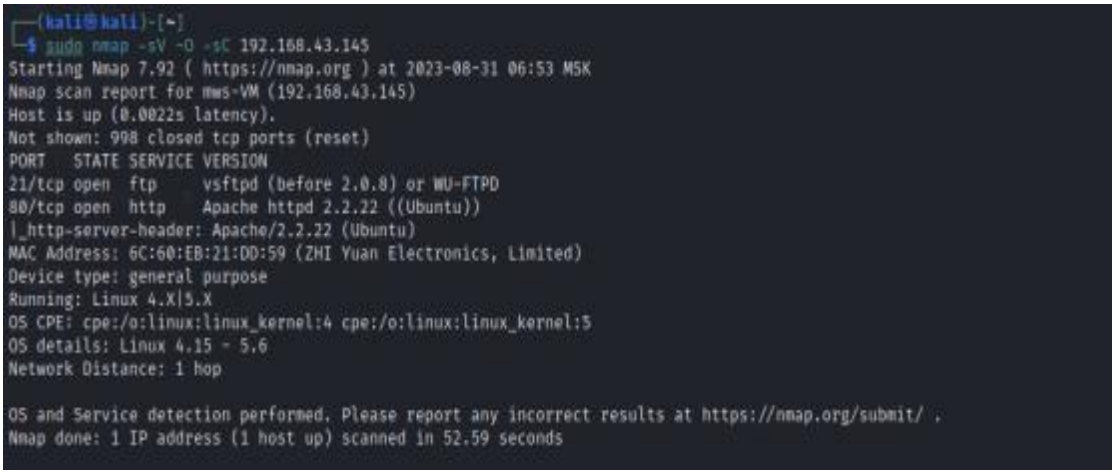
Try these out in the OpenCanary configs for more typical server personalities.

- Linux Web Server
- Windows Server
- MySQL Server
- MSSQL Server

حيث باضافة الاسطر التالية ضمن ملف الاعداد الخاص بـ opencanary ستظهر المصيدة وكأنها مخدم ويب بنظام تشغيل لينوكس كما يلي:

```
{
 "ftp.banner": "FTP server ready",
 "ftp.enabled": true,
 "ftp.port":21,
 "http.banner": "Apache/2.2.22 (Ubuntu)",
 "http.enabled": true,
 "http.port": 80,
 "http.skin": "nasLogin",
 "http.skin.list": [
     {
         "desc": "Plain HTML Login",
         "name": "basicLogin"
     },
     {
         "desc": "Synology NAS Login",
         "name": "nasLogin"
     }
 ],
 "ssh.enabled": true,
 "ssh.port": 8022,
 "ssh.version": "SSH-2.0-OpenSSH_5.1p1 Debian-4",
 // [..] # logging configuration
}
```

وسيظهر الخرج التالي عند محاولة تنفيذ الأمر nmap ضد موضع الجذب Opencanary:




وبذلك نكون قد قمنا بتنصيب واعداد المصيدة Opencanary بشكل صحيح.

# الفصل الرابع: التنفيذ العملي

## 4-1 التنفيذ العملي:

### 4-1-1 بنية الشبكة:

يظهر الشكل (1) المخطط العام لشبكة مكونة من ثلاث مناطق هي:

- الشبكة الداخلية (LAN) : وتحتوي هذه الشبكة على المخدمات الرئيسية للمؤسسة بالاضافة إلى أجهزة العمل الموجودة ضمن المؤسسة ومن بينها جهاز الادارة.
- الشبكة المعزولة (DMZ): وتضم المخدمات المعرضة للوصول من الانترنت والتي تقدم الخدمات للزبائن.
- الشبكة الخارجية (WAN): وهذه الشبكة يمكن اعتبارها شبكة الانترنت.

يربط بين هذه الشبكات جدار ناري مؤلف من ثلاث بوابات أو واجهات واحدة لكل شبكة، وقد تم تطبيق القواعد على هذا الجدار بحيث يتم تحقيق الشروط الأساسية التالية:

- ✓الشبكة الداخلية (LAN): يمكنها الوصول إلى جميع الشبكات وفقط جهاز الادارة ضمن هذه الشبكة يمكنه الوصول إلى اعدادات الجدار الناري.
- ✓الشبكة المعزولة (DMZ): يمكنها الوصول إلى الشبكة الخارجية وبالتالي شبكة الانترنت ولكن لا يمكنها الوصول إلى الشبكة الداخلية.
- ✓الشبكة الخارجية (WAN): يمكن لهذه الشبكة الوصول إلى الشبكة المعزولة فقط بينما لا يمكنها الوصول إلى الشبكة الداخلية.

وقد تم تطبيق نظام العنونة التالي على الشبكات الفرعية السابقة وعلى الجدار الناري وفق التالي:

- ✓عنوان واجهة الشبكة الداخلية (LAN) على الجدار الناري هو: 192.168.1.200 /24
- ✓عنوان الواجهة للشبكة المعزولة (DMZ) على الجدار الناري هو: 192.168.2.200 /24
- ✓عنوان واجهة الشبكة الخارجية (WAN) على الجدار الناري هو: 192.168.43.200 /24

وبالتالي يمكن الاستنتاج بأن عناوين الشبكات الثلاث هي:

- الشبكة الداخلية: 192.168.1.0 /24
- الشبكة المعزولة: 192.168.2.0 /24
- الشبكة الخارجية: 192.168.43.0 /24

## 4-1-2 مكونات النظام:

**أولاً: المصيدة Opencanary:**

مصيدة مفتوحة المصدر تتميز بدعم لعدد من البروتوكولات مثل HTTP, HTTPS, SSH, FTP, TFTP, SNMP, SMTP وغيرها.

**ثانياً: المصيدة SNORT:**

- مصيدة مفتوحة المصدر تتعامل مع كامل حركة المرور ضمن مجال معين يتم تحديده لها من العناوين أو الشبكات الفرعية، وبذلك لا تقوم بالتفاعل فقط مع حركة المرور الموجهة إليها فقط بل تتفاعل مع كامل حركة المرور ضمن الشبكة وبالتالي فهي تعمل كنظام IPS/IDS .
- تم اختيار نظام العمل لهذه المصيدة بحيث تعمل كنظام كشف للتسلل Intrusion Detection System وتم اسناد عناون IP لها هو: 192.168.2.201 /24

تم تثبيت جميع هذه التجهيزات ضمن بيئة افتراضية متكاملة باستخدام برنامج VMWare Workstation 17 Pro ضمن الجهاز الفيزيائي المحتضن لكامل هذا النظام.

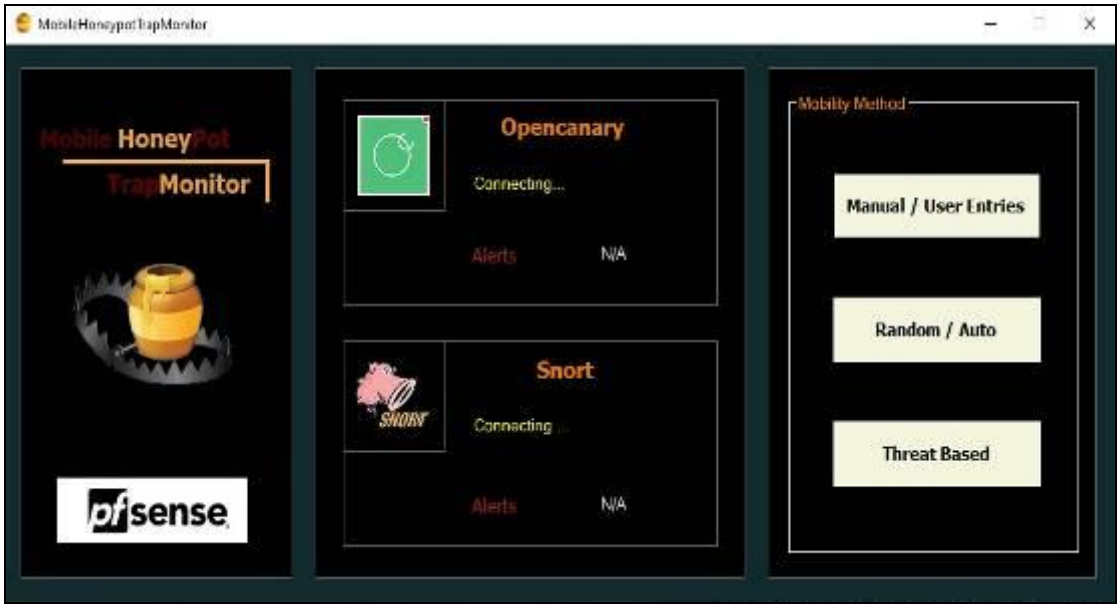
الجهاز الفيزيائي هو مصدر الانترنت ومن خلاله يتم الوصول إلى شبكة الانترنت من خلال بطاقة الشبكة اللاسلكية الخاصة به حيث يملك العنوان IP التالي: 192.168.43.10 /24 عبر شبكة لاسلكية WIFI من خلال مودم لاسلكي له العنوان IP التالي: 192.168.43.1 /24

جهاز الادارة وهو جهاز ضمن الشبكة الداخلية تم اسناد العنوان IP التالي له وهو: 192.168.1.10 /24 وهذا الجهاز يتضمن برنامج الادارة.

تم استخدام بيئة Visual Studio 2019 ولغة البرمجة C# لبرمجة هذا البرنامج بحيث تم التغلب على استراتيجية الكرسي الدوار بحيث يتم الوصول إلى كافة المصائد التي تم نشرها ضمن هذه الشبكة والتعامل مع ملفات التسجيلات المختلفة وملفات الاعدادات والتفاعل التام مع هذه المصائد من خلال منصة موحدة دون الحاجة للتنقل بين هذه المصائد في كل مرة نحتاج فيها للتعديل على الاعدادات أو اعادة التشغيل والنشر.

**ثالثاً: برنامج الادارة:**

عند تشغيل رنامج الادارة تظهر الواجهة الرئيسية الخاصة به كما هو موضح في الشكل التالي:




الشكل (14): الواجهة الرئيسية لبرنامج الادارة

حيث نجد في القسم الأيمن من الواجهة الخيارات التالية:

**Manual/User Entries**: اختيار التفضيلات المطلوبة من قبل مدير النظام من تحديد الخدمات المفعلة وأرقام المنافذ وعنوان المصيدة المنطقي والفيزيائي والدور الذي ستلعبه المصيدة (مخدم، جهاز حاسب، طابعة، هاتف انترنت أو غير ذلك).

**Random/Auto:** هذا الخيار يقوم بتحديد موديل معين للمصيدة يتم اختياره من خلال اجراء تجميع عشوائي لمجموعة من البرامترات التي تمتلك كل منها مجموعة من القيم

الافتراضية يتم اختيار قيمة واحدة فقط لكل برامتر بشكل عشوائي بحيث يتم تكوين الموديل بشكله النهائي ليتم تطبيقه.

**Threat Based:** وهنا يأتي دور نظام كشف التسلل Snort حيث يتم من خلال تحليل ملف التسجيلات الخاص به، تحديد أكثر الأماكن التي تتعرض لهجمات في الشبكة ومنه يتم تحديد الموديل المناسب والمكان الذي يجب أن تنتقل له المصيدة سواء داخل الشبكة LAN أو الشبكة المعزولة DMZ.

## 4-1-3 عملية تنقيل المصيدة:

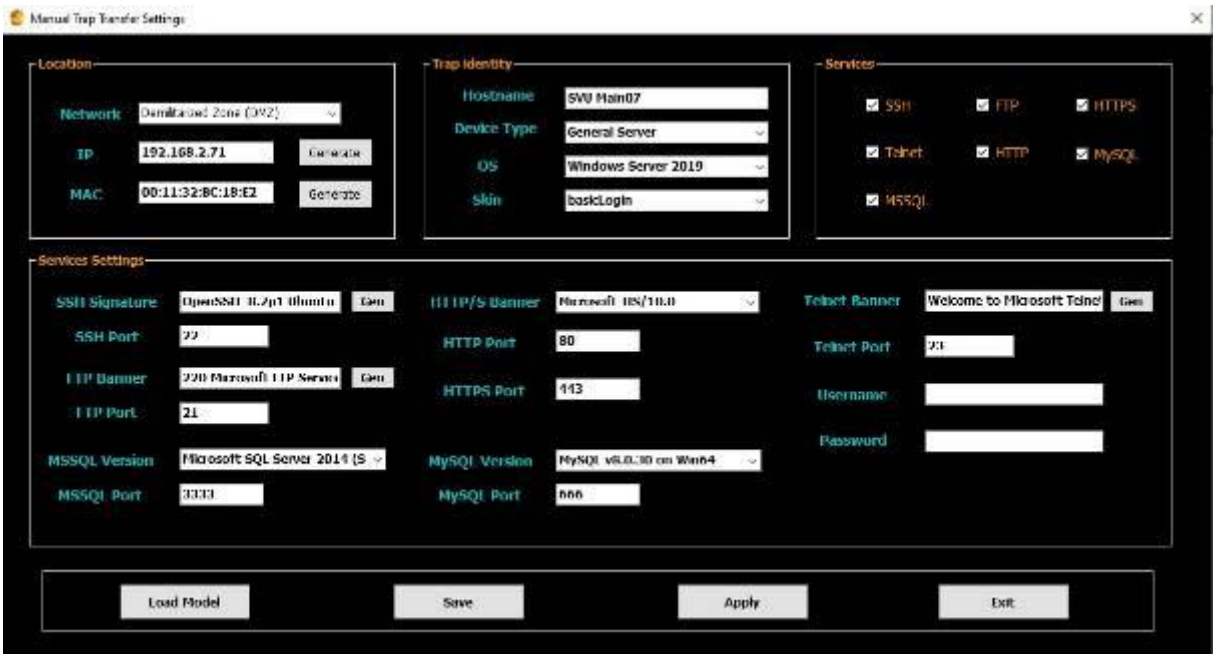
### 4-1-3-1 التنقيل اليدوي للمصيدة:

عند الضغط على الزر Manual تظهر لنا الواجهة التالية:


الشكل (15): واجهة الادخالات اليدوية لبرامترات المصيدة

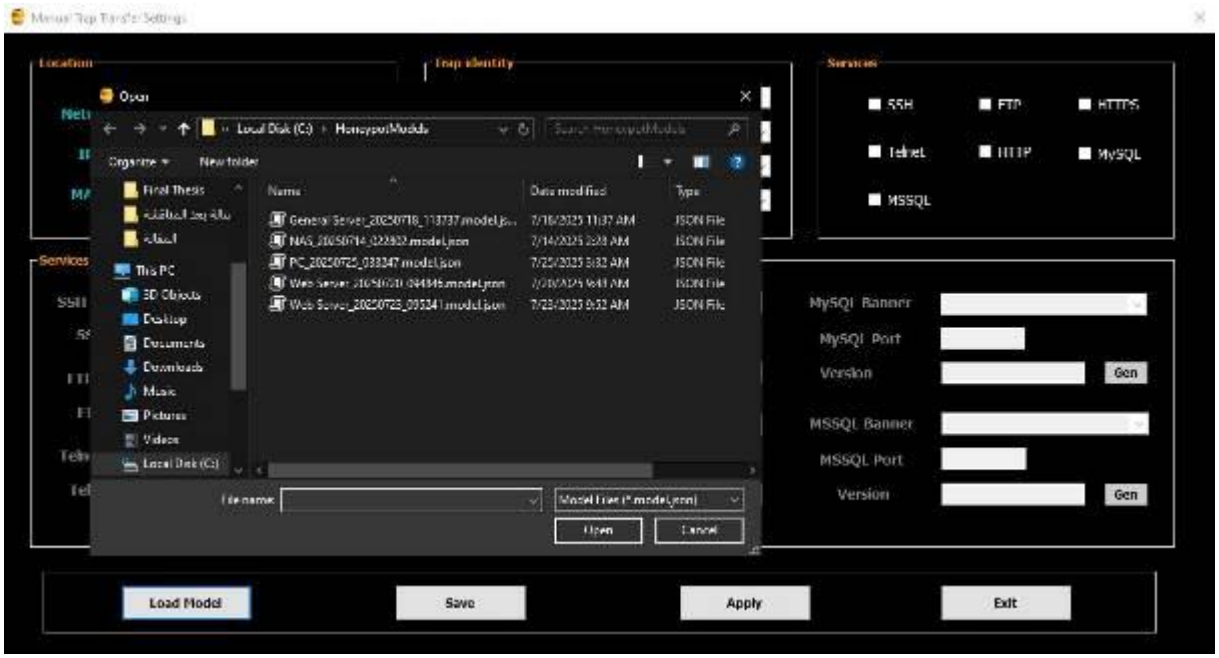
وهنا نجد جميع البرامترات التي يمكن لمدير النظام اختيارها و بالضغط على الزر Apply سوف يتم بناء ملف موديل مخصص وسيتم من خلاله بناء ملف اعدادات قياسي للمصيدة حيث يتم نقله إلى المصيدة وبعد ذلك تطبيقه واعادة تشغيل المصيدة تلقائياً ليتم تطبيق هذه الاعدادات.

يظهر الشكل التالي ادخالات ضمن الواجهة السابقة بحيث تصبح المصيدة عبارة عن مخدم ويب يعمل بنظام التشغيل ويندوز.




الشكل(16): واجهة الادخالات اليدوية من أجل مخدم ويب يعمل بنظاك التشغيل ويندوز

بالضغط على الزر Save يتم حفظ الموديل المحدد وفق الادخالات ضمن مجلد HoneypotModels المحدد مسبقاً، كما يمكن القيام بتخزين العديد من النماذج بحيث يمكن استرجاعها وتحميلها للواجهة من خلال الزر Load Model كما هو موضح في الشكل التالي:



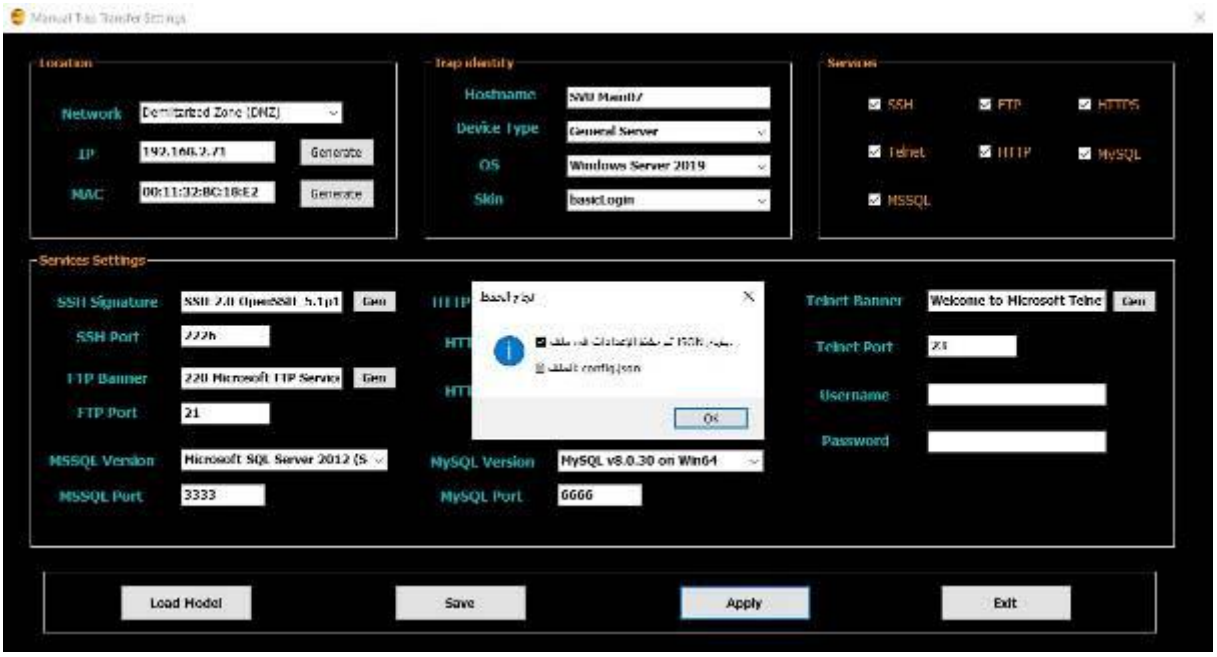

الشكل (17): واجهة اختيار موديل لتحميله إلى المصيدة

تم تصميم حقول الادخالات بحيث يمكن للمستخدم الادخال اليدوي أو اختيار الادخال المطلوب من ضمن مجموعة من الادخالات القياسية التي تتناسب مع الدور المحدد للمصيدة بحيث يكون الموديل الناتج متناسق ومتوافق مع الهوية المحددة لها.

يتم تطبيق الموديل المحدد من خلال الضغط على الزر Apply حيث يتم تنفيذ مجموعة من الخطوات وفق الترتيب التالي:

### أولاً: اختبار الاتصال مع المصيدة Opencanary:

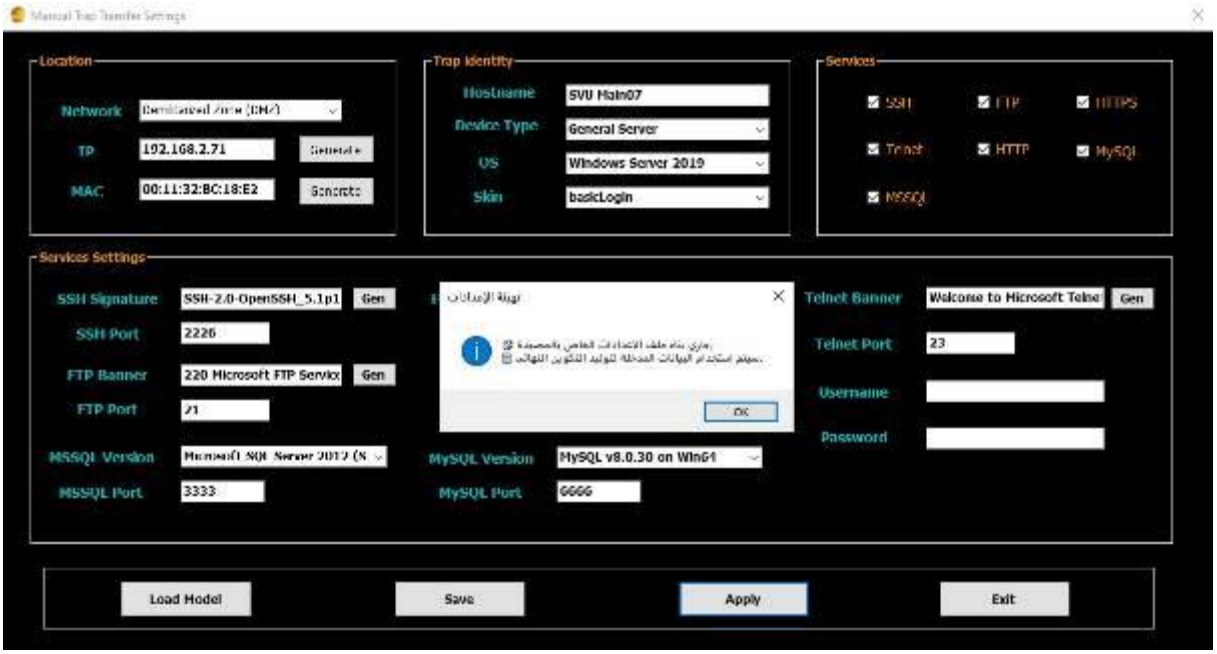
حيث يتم ذلك من أجل العنوان المحدد للبدء وهو العنوان 192.168.2.4/24 وفي حال كان الاتصال محقق يتم الانتقال لمرحلة التنفيذ، حيث يتم حفظ ملف النموذج الجديد ضمن المجلد HoneypotModels كما يتم حفظ اعدادات التنفيذ ضمن المجلد Config.json كماهو موضح بالشكل التالي:




يتم بناء النموذج Model من الادخالات التي تم اختيارها من قبل مدير النظام من خلال بناء ملف النموذج بصيغة Json يتضمن عدد من العقد كل منها تحتوي على اسم الخاصية مرفقة بالقيمة التي تم اختيارها كما هو موضح بالمثال التالي:

```
{
  "Network": "Demilitarized Zone (DMZ)",
  "IP": "192.168.2.50",
  "MAC": "00:16:3E:DA:7D:30",
  "Hostname": "sas",
  "DeviceType": "Web Server",
  "OS": "Windows Server 2008 R2",
  "Skin": "basicLogin",
  "device.node_id": "sas_Web Server",
  "ftp.enabled": true,
  "ftp.port": 21,
  "ftp.banner": "220 Generic FTP Service Ready",
  "http.enabled": false,
  "http.port": 80,
  "http.banner": "Microsoft-IIS/10.0",
  "http.skin": "nasLogin",
  "https.enabled": true,
  "https.port": 443,
  "https.skin": "basicLogin",
  "ssh.enabled": true,
  "ssh.port": 2226,
  "ssh.version": "SSH-2.0-OpenSSH_5.1p1 Debian-4",
  "telnet.enabled": false,
  "telnet.port": 23,
  "telnet.banner": "",
  "mysql.enabled": false,
  "mysql.port": 3306,
  "mysql.banner": "5.5.43-0ubuntu0.14.04.1",
  "mssql.enabled": false,
  "mssql.port": 1433,
  "mssql.version": "2012"
}
```

حيث يتم البدء ببناء ملف الاعدادات من القيم المخزنة في ملف النموذج كما هو موضح بالشكل التالي:



ومن خلال ملف النموذج (trap_config.json) يتم ببناء ملف الاعدادات القياسي الذي تقبله المصيدة Opencanary بحيث ينتج لدينا ملف اعدادات صالح Valid كما هو موضح بالمثال التالي:


```
{
"device.node_id": "sas_Web Server",
"ip.ignorelist": [],
"logtype.ignorelist": [],
"git.enabled": false,
"git.port": 9418,
"ftp.enabled": true,
"ftp.port": 21,
"ftp.banner": "220 Generic FTP Service Ready",
"ftp.log_auth_attempt_initiated": false,
"http.banner": "Microsoft-IIS/10.0",
"http.enabled": false,
"http.port": 80,
"http.skin": "nasLogin",
"http.log_unimplemented_method_requests": false,
"http.log_redirect_request": false,
"https.enabled": true,
"https.port": 443,
"https.skin": "basicLogin",
"https.certificate": "/etc/ssl/opencanary/opencanary.pem",
"https.key": "/etc/ssl/opencanary/opencanary.key",
"httpproxy.enabled": false,
"httpproxy.port": 8080,
"httpproxy.skin": "squid",
"llmnr.enabled": false,
"llmnr.query_interval": 60,
"llmnr.query_splay": 5,
"llmnr.hostname": "DC03",
"llmnr.port": 5355,
"logger": {
"class": "PyLogger",
"kwargs": {
"formatters": {
"plain": {
"format": "%(message)s"
},
"syslog_rfc": {
"format": "opencanaryd[%(process)-5s:%(thread)d]:
%(name)s %(levelname)-5s %(message)s"
}
},
"handlers": {
"console": {
"class": "logging.StreamHandler",
"stream": "ext://sys.stdout"
},
"file": {
"class": "logging.FileHandler",
"filename": "/var/tmp/opencanary.log"
}
}
}
},
```

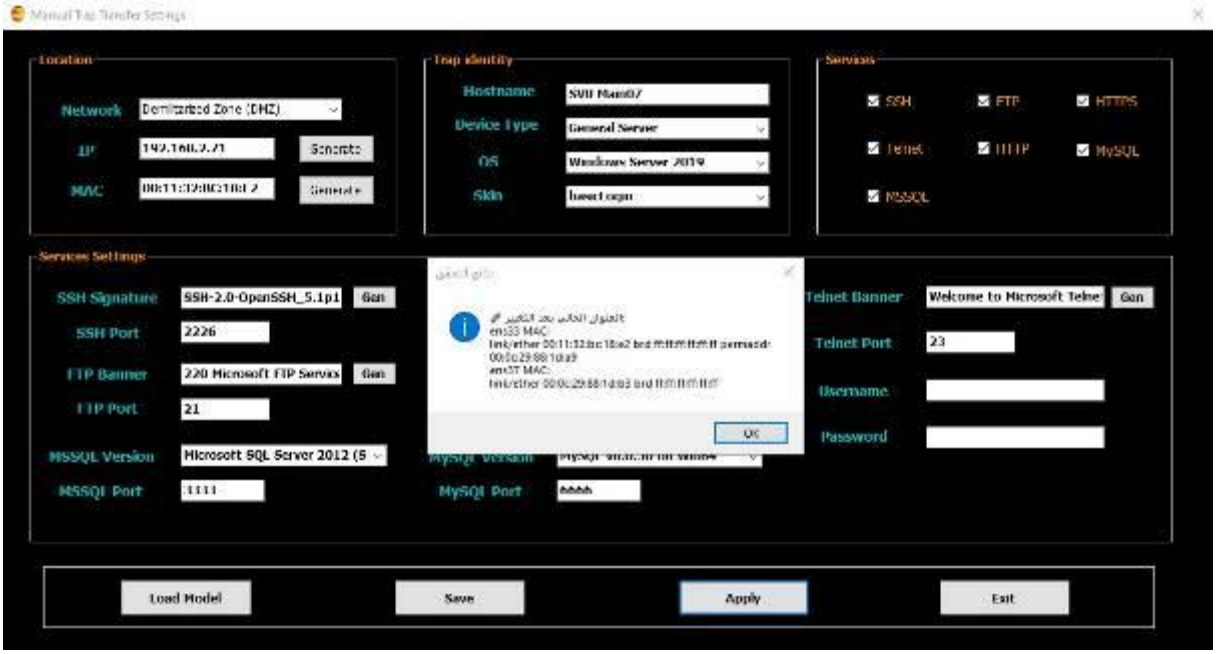


```
"portscan.enabled": false,
"portscan.ignore_localhost": false,
"portscan.logfile": "/var/log/kern.log",
"portscan.synrate": 5,
"portscan.nmaposrate": 5,
"portscan.lorate": 3,
"portscan.ignore_ports": [],
"smb.auditfile": "/var/log/samba-audit.log",
"smb.enabled": false,
"mysql.enabled": false,
"mysql.port": 3306,
"mysql.banner": "5.5.43-0ubuntu0.14.04.1",
"mysql.log_connection_made": false,
"ssh.enabled": true,
"ssh.port": 2226,
"ssh.version": "SSH-2.0-OpenSSH_5.1p1 Debian-4",
"redis.enabled": false,
"redis.port": 6379,
"rdp.enabled": false,
"rdp.port": 3389,
"sip.enabled": false,
"sip.port": 5060,
"snmp.enabled": false,
"snmp.port": 161,
"ntp.enabled": false,
"ntp.port": 123,
"tftp.enabled": false,
"tftp.port": 69,
"tcpbanner.maxnum": 10,
"tcpbanner.enabled": false,
"tcpbanner_1.enabled": false,
"tcpbanner_1.port": 8001,
"tcpbanner_1.datareceivedbanner": "",
"tcpbanner_1.initbanner": "",
"tcpbanner_1.alertstring.enabled": false,
"tcpbanner_1.alertstring": "",
"tcpbanner_1.keep_alive.enabled": false,
"tcpbanner_1.keep_alive_secret": "",
"tcpbanner_1.keep_alive_probes": 11,
"tcpbanner_1.keep_alive_interval": 300,
"tcpbanner_1.keep_alive_idle": 300,
"telnet.enabled": false,
"telnet.port": 23,
"telnet.banner": "",
"telnet.honeycreds": [
{
"username": "admin",
"password": "$pbkdf2-
sha512$19000$bG1NaY3xvjdGyBlj7N37Xw$dGrmBqqWa1okTCpN
3QEmeo9j5DuV2u1EuVFD8Di0GxNiM64To5O/Y66f7UASvnQr8.L
CzqTm6awC8Kj/aGKvwA"
},
{
"username": "admin",
"password": "admin1"
}
],
"telnet.log_tcp_connection": false,
"mssql.enabled": false,
"mssql.version": "2012",
"mssql.port": 1433,
"vnc.enabled": false,
"vnc.port": 5000
}
```

**ثانياً: تغيير العنوان الفيزيائي للمصيدة MAC Address:**

حيث يتم ارسال أمر مؤجل للمصيدة بحيث يتم التنفيذ وانتظار انهاء العملية واعادة الاتصال مع المصيدة باستخدام اعدادات الشبكة الجديدة لتجنب انقطاع الاتصال مع المصيدة كما هو موضح بالشكل التالي:


حيث يظهر لمدير النظام سير عملية التغيير للتأكد من أن كل الأوامر يتم تنفيذها بشكل صحيح وبالتالي لن يتم فقدان التحكم بالمصيدة أثناء عملية التغيير كما هو موضح في الشكل التالي:



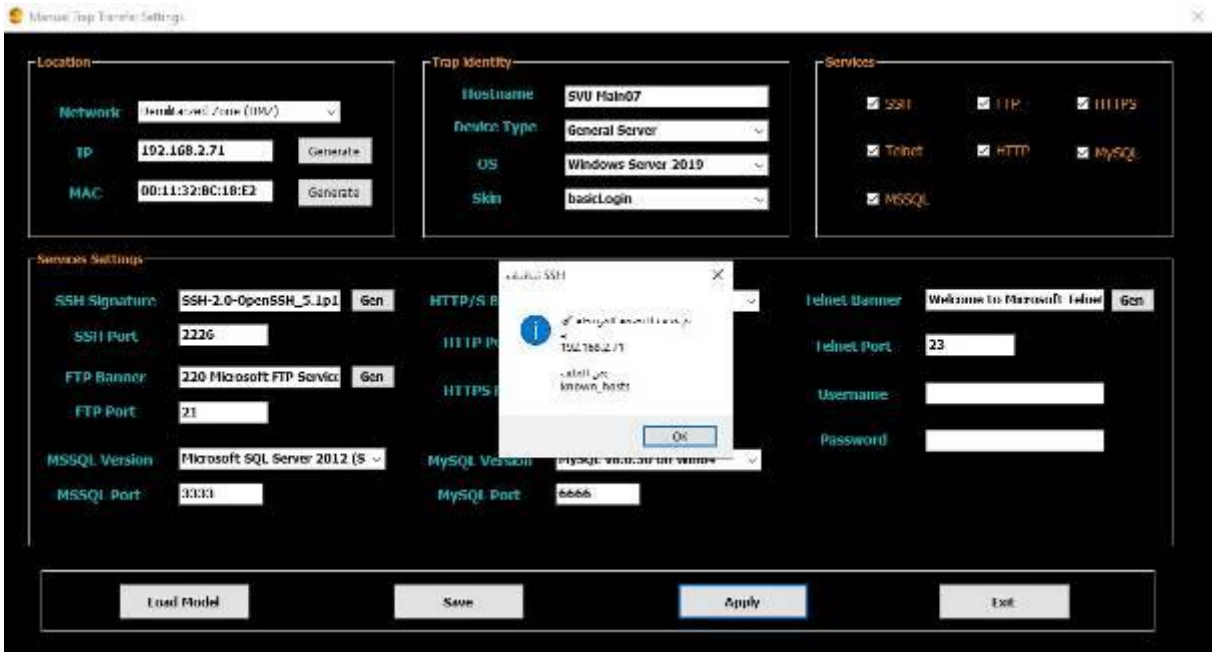
### ثالثاً: تغيير العنوان المنطقي للمصيدة IP Address وعنوان البوابة الافتراضية Default Gateway:

حيث يتم ذلك اعتماداً على المنطقة التي سيتم النقل اليها والمحددة ضمن النموذج، حيث تتم العملية بشكل مشابه للعملية السابقة كما هو موضح بالشكل التالي:

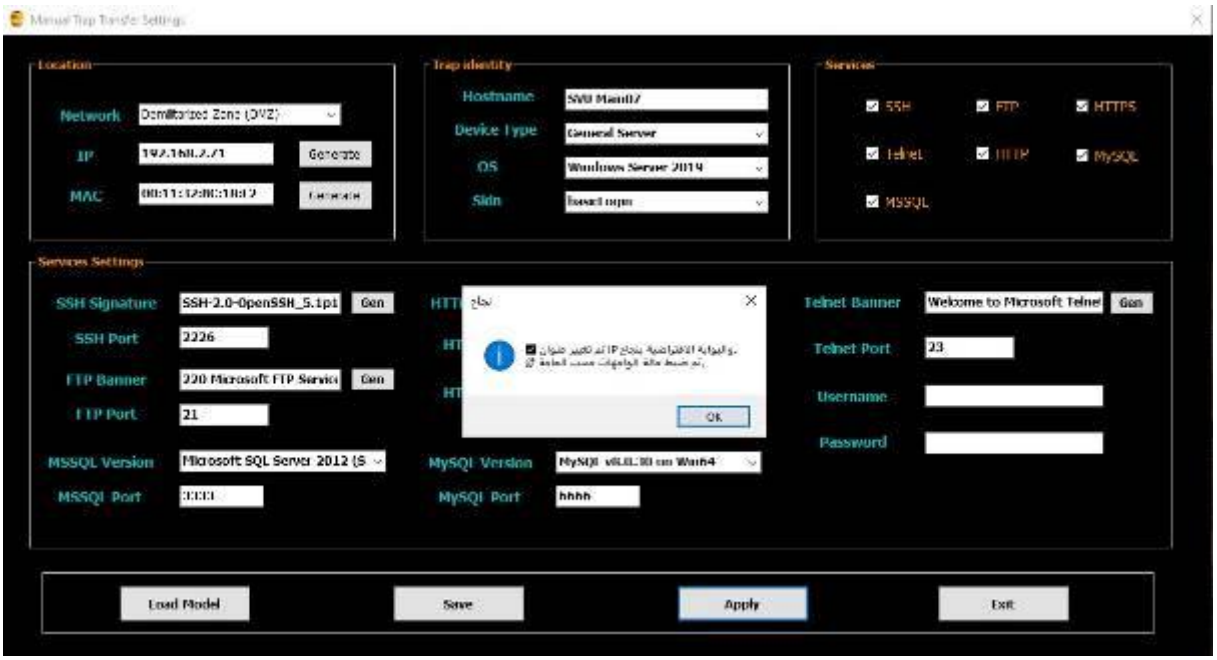

وهنا يجب التنويه بأنه يتم حذف(Fingerprint) المرتبطة بالعنوان IP القديم للمصيدة من جهاز الادارة

كما هو موضح بالشكل التالي:

وأيضاً حذف البصمة (Fingerprint) المرتبطة بالعنوان الجديد ان كانت موجودة من الملف Known Hosts ضمن جهاز الادارة وذلك ليتم استكمال عملية تأسيس الاتصال بالمصيدة وفق العناوين الجديدة دون مشاكل كما هو موضح في الشكل التالي:




وعند اكتمال التنفيذ تظهر النتيجة لمدير النظام كما هو موضح في الشكل التالي:



### رابعاً: تغيير اسم الألة الافتراضية المثتبت عليها المصيدة Hostname:

يعتبر هذا الاجراء ضروري من أجل التمويه الكامل للهوية الجديدة للمصيدة حيث تبدأ العملية كما هو موضح بالشكل التالي:


و يظهر لمدير النظام تقدم عملية التنفيذ مع معلومات عن حالة الاسم الجديد للمصيدة وفق الشكل التالي:

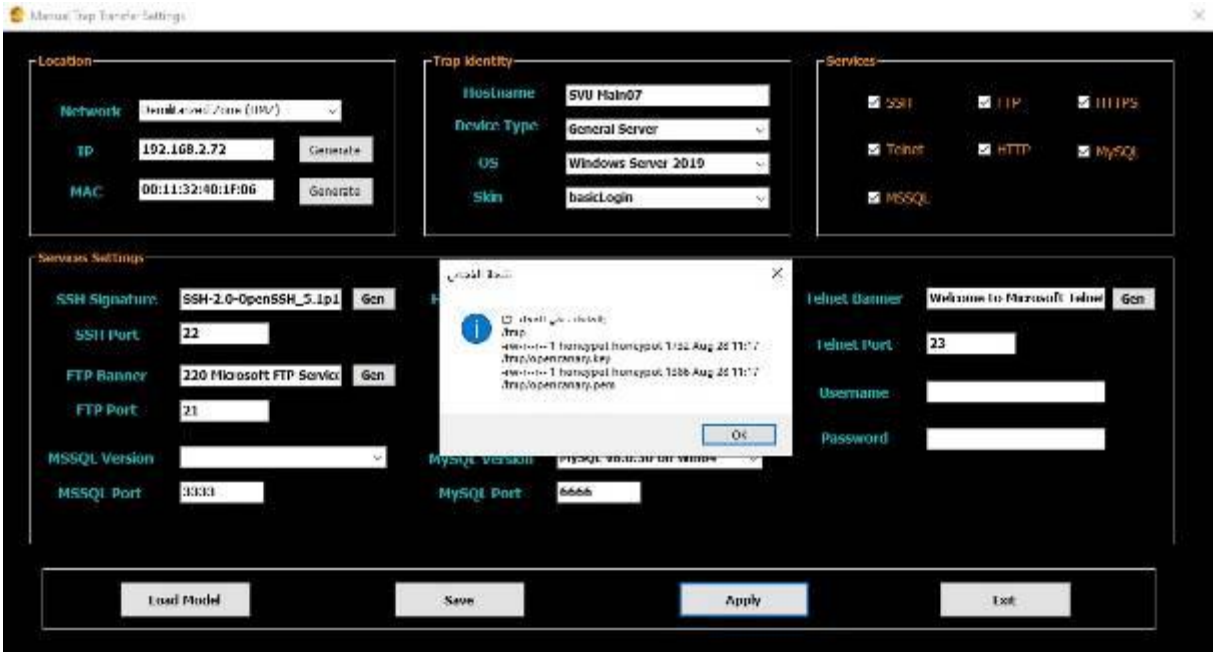
وعند اكتمال التنفيذ بنجاح يظهر لمدير النظام رسالة توضح نتيجة هذا التنفيذ كماهو موضح بالشكل التالي:


## خامساً: تغيير شهادة SSL Certificate للمصيدة:

حيث يتم انشاء شهادة SSL جديدة للمصيدة بحيث تظهر المصيدة كجهاز جديد لأي مهاجم محتمل عند محاولته تسجيل الدحول للمصيدة في حالة كونها تلعب دور مخدم ويب على سبيل المثال كما هو موضح بالشكل التالي:

وتظهر نتيجة التنفيذ لمدير النظام كما هو موضح بالشكل التالي:

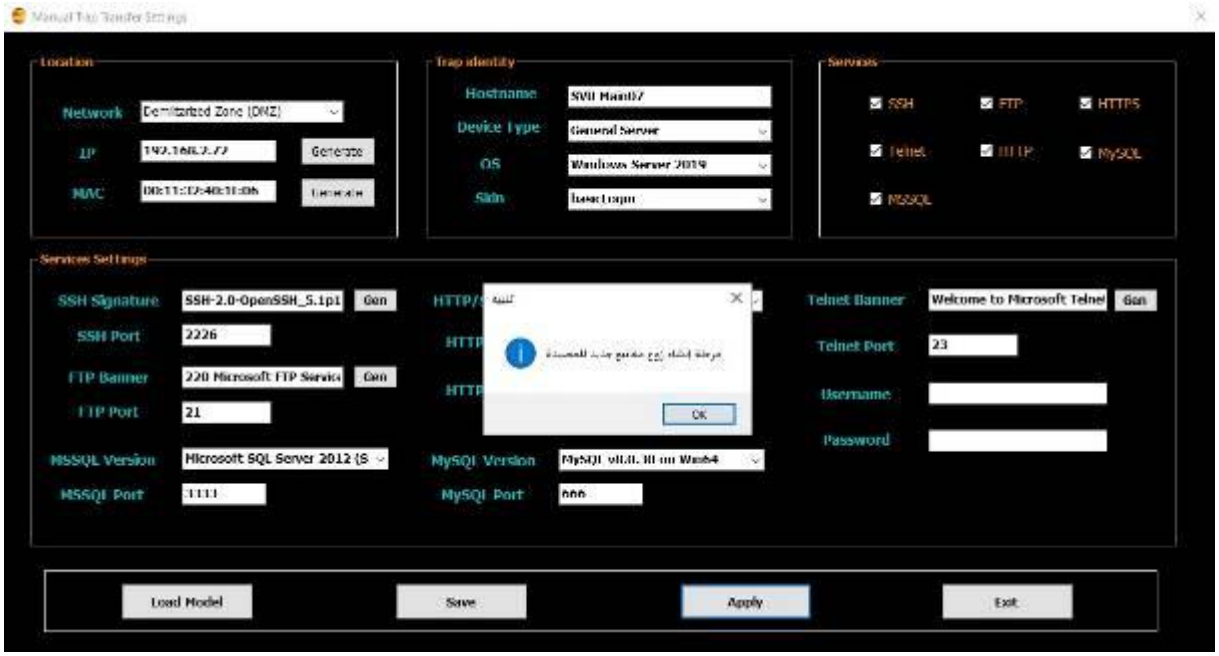

وهنا يتم رفع ملف المفتاح وملف الشهادة إلى المصيدة للمجلد tmp وبعد ذلك يتم عرض هذه الملفات بعد تحديث صلاحيات الوصول اليها كما هو موضح بالشكل التالي:



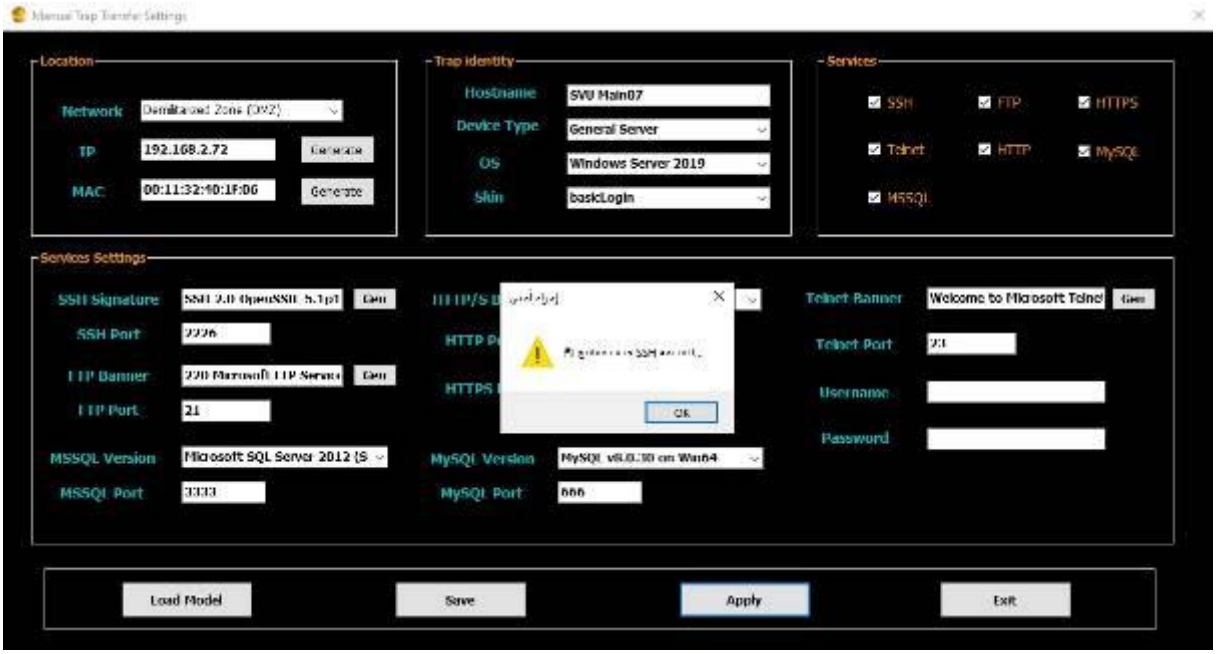
وبعد ذلك يتم عرض رسالة تظهر الملفات النهائية لهذه المرحلة على المصيدة كما هو موضح بالشكل التالي:


## سادساً: تغيير زوج المفاتيح الخاص والعام للمصيدة:

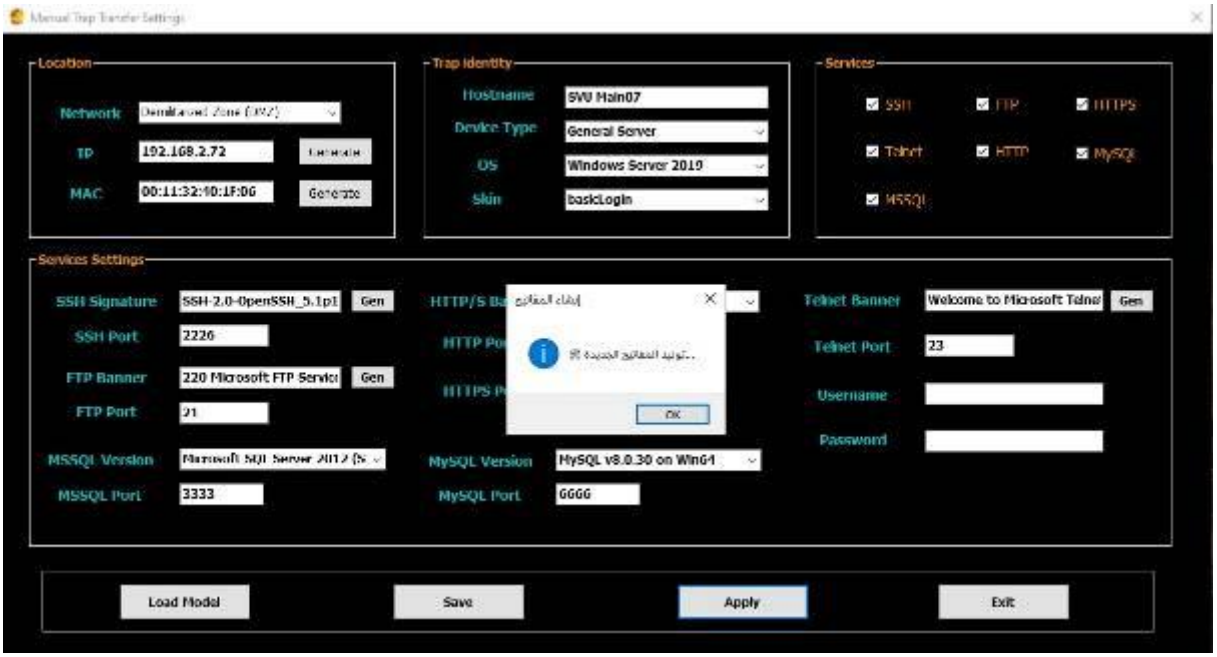
حيث تبدأ بعد ذلك مرحلة انشاء زوج مفاتيح جديد للمصيدة كما هو موضح بالشكل التالي:



حيث يتم بداية حذف زوج المفاتيح القديم من المصيدة كما هو موضح بالشكل التالي:



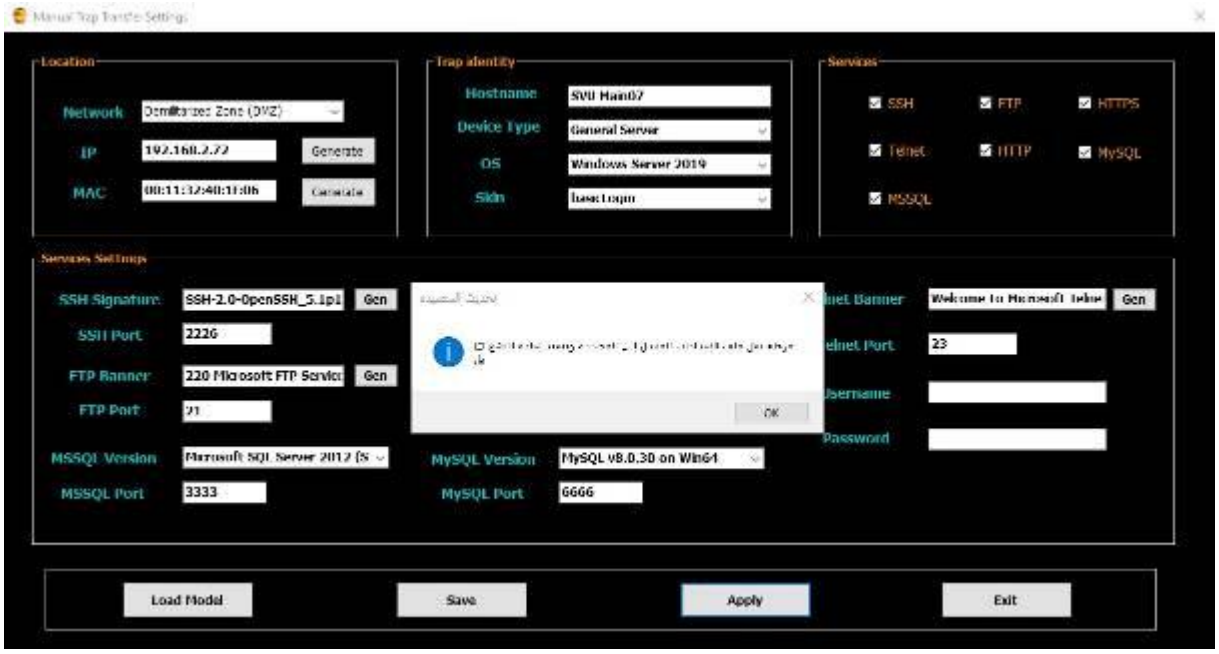

بعد ذلك يتم اعلام مدير النظام عن بدء توليد المفاتيح الجديدة كما هو موضح بالشكل التالي:



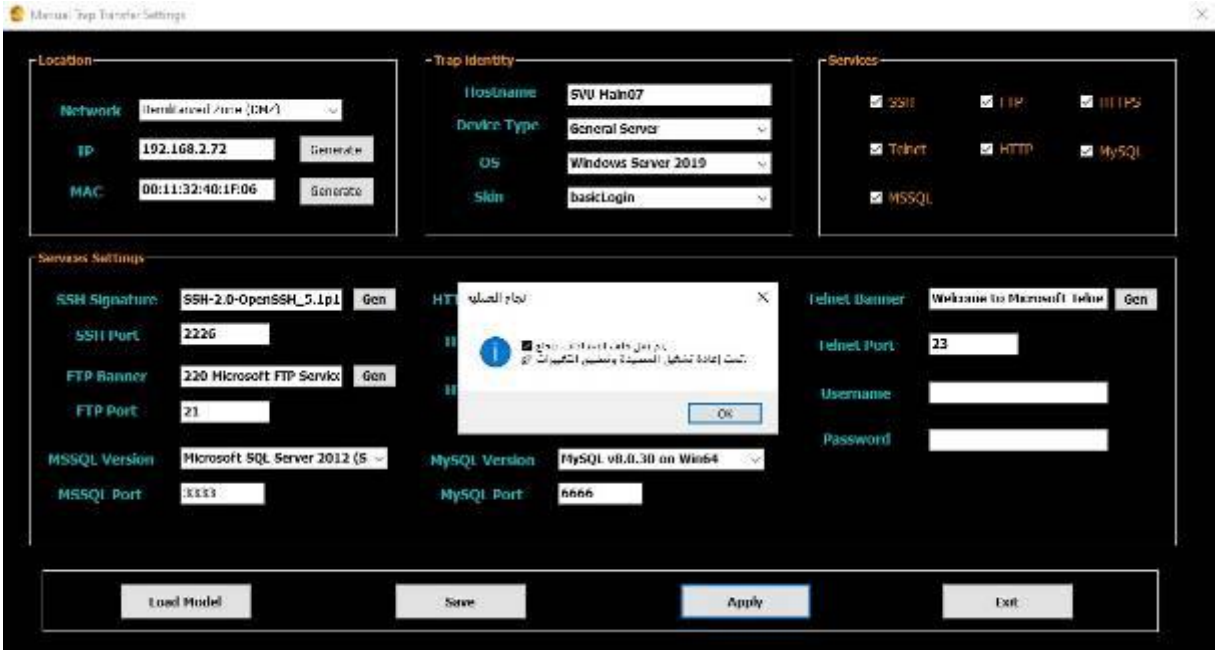

يتم بعد ذلك عملية اعادة تشغيل الخدمة SSH على الجهاز المضيف للمصيدة كما هو موضح بالشكل التالي:


وبعد ذلك يتم اعلام مدير النظام بنتائج تنفيذ هذه المرحلة كما هو موضح بالشكل التالي:

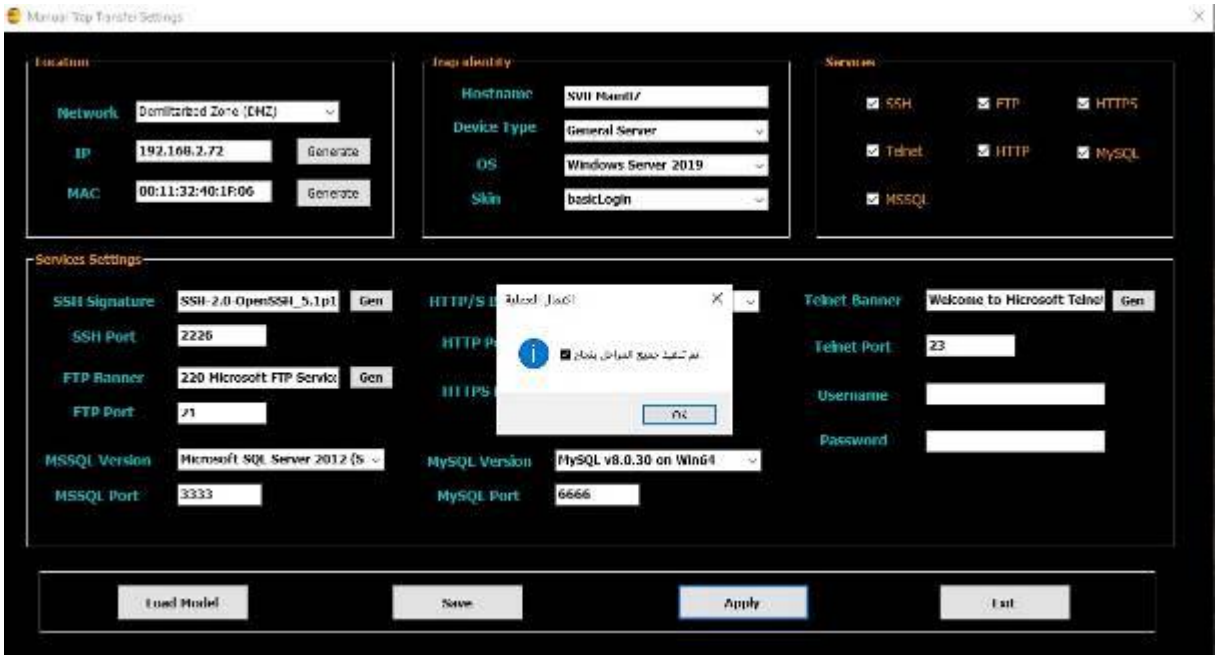
وبعد الانتهاء من تنفيذ جميع العمليات السابقة بنجاح يتم الانتقال لمرحلة نقل ملف الاعدادات المعدل للمصيدة واعادة تشغيلها كما هو موضح بالشكل التالي:



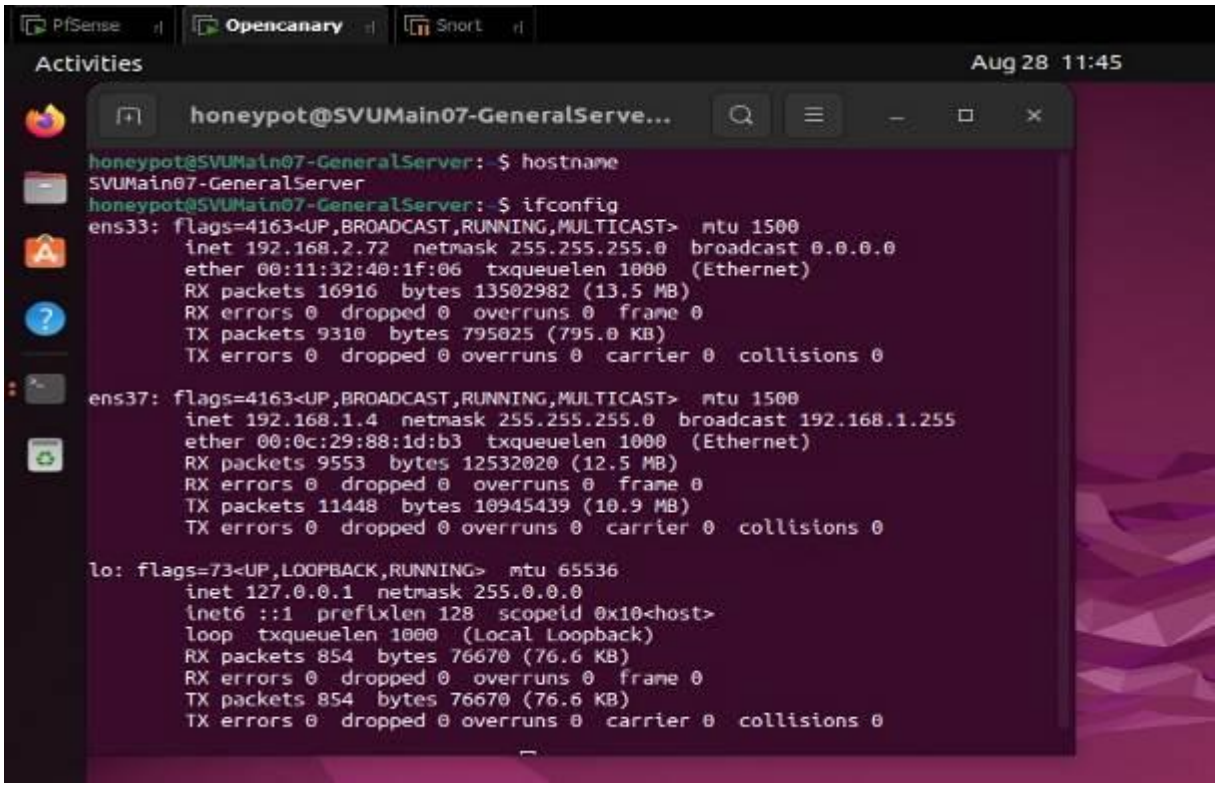

حيث يتم نقل هذا الملف من جهاز الادارة عبر يرنامج الادارة إلى المصيدة وتحديداً إلى المسار /etc/opencanaryd/opencanary.conf باستخدام الروتوكول SSH، ويتم اعادة تشغيل المصيدة Opencanary ومراقبة ملف التسجيلات للتأكد من أن كل العمليات السابقة قد تم تنفيذها بشكل صحيح دون أخطاء كما هو موضح بالشكل التالي:



وعند اكتمال تنفيذ كامل العمليات السابقة بنجاح يتم اظهار نتيجة التنفيذ الكلي لمدير التظام كما هو موضح بالشكل التالي:



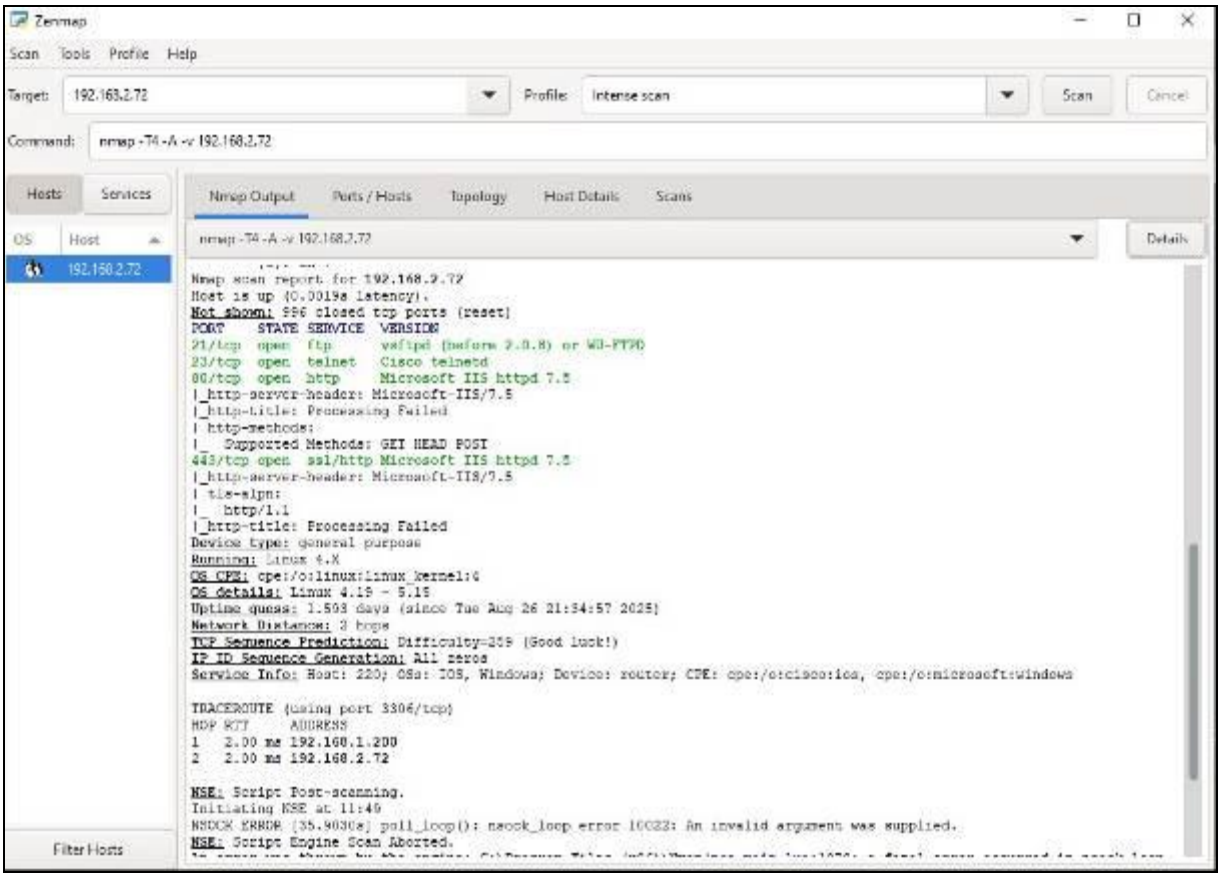

حيث يمكن التأكد من تطبيق جميع التغييرات السابقة على المصيدة وسنلاحظ نجاح التطبيق، حيث يمكن الاطلاع على العنوان الجديد للمصيدة والتأكد من أن العنوان الجديد قد تم تغييره وكذلك اسم الألة الافتراضية وباقي الاعدادات كما هو موضح بالشكل التالي:



كما يمكن تتبع صحة تنفيذ هذه الاجراءات على المصيدة من جهاز المهاجم من خلال القيام باجراء مسح باستخدام الأمر التالي:

Nmap -T4 -A -v 192.168.2.72

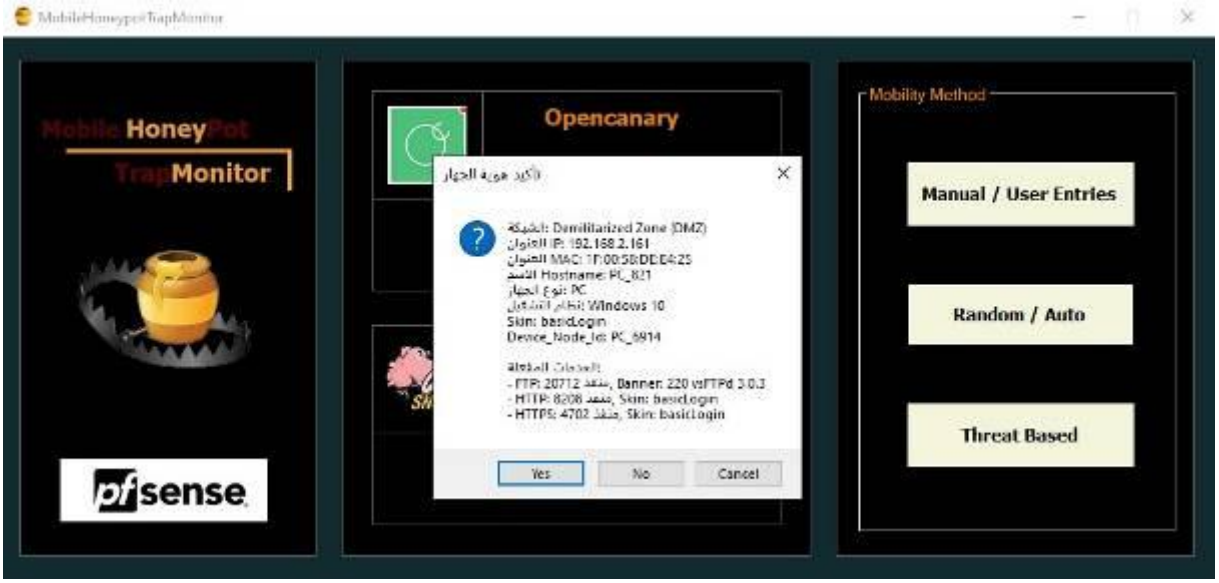
وسنجد أن المصيدة قد تم تغيير هويتها بما يتوافق مع الادخالات التي قام بها مدير النظام كما هو موضح بالشكل التالي:



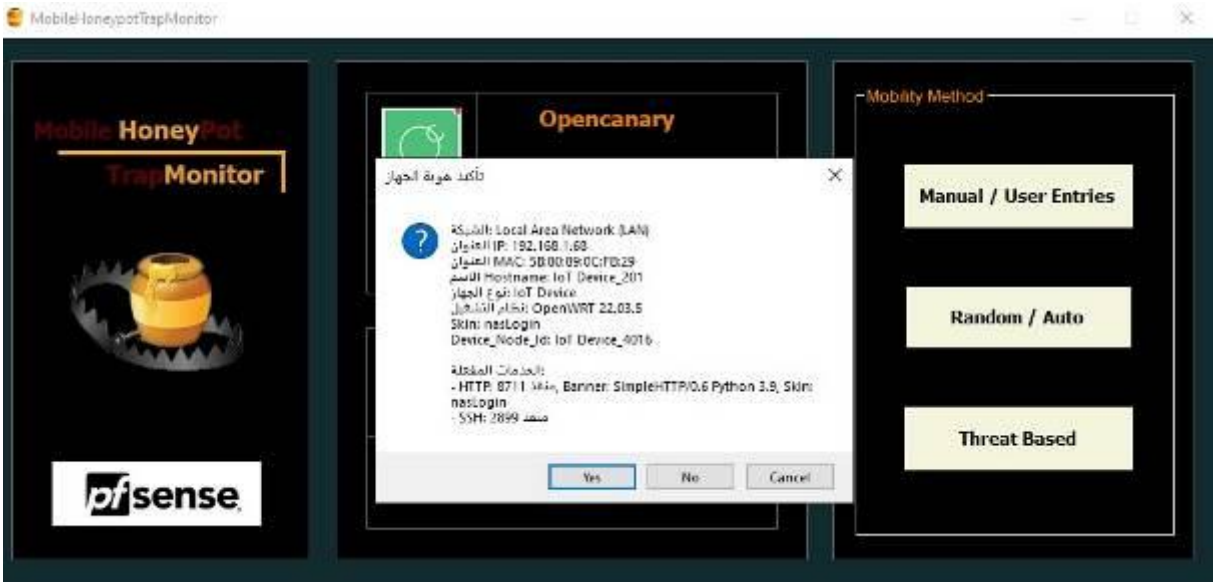

وبذلك نكون قد نجحنا في تغيير الهوية بشكل كامل للمصيدة Opencanary بنجاح وفق ادخالات مدير النظام بحيث تظهر المصيدة وكأنها جهاز مختلف تماماً بالنسبة للمهاجم المحتمل.

### 4-1-3-2 التنقيل العشوائي للمصيدة:

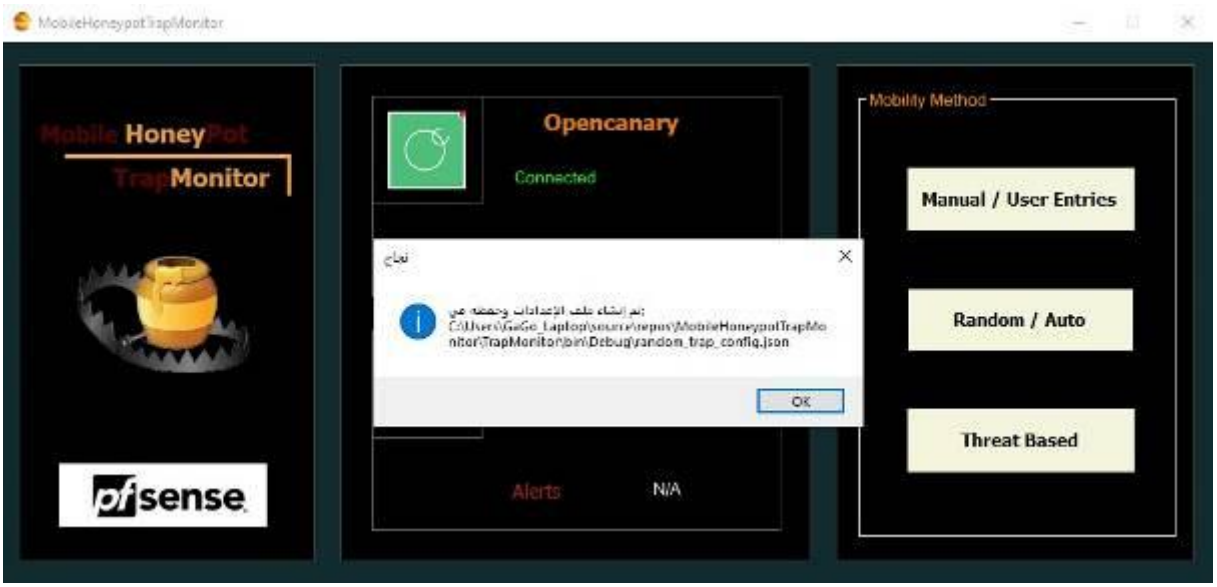
عند الضغط على الزر Random/Auto يتم تشكيل هوية جديدة للمصيدة Opencanary بشكل عشوائي دون الاعتماد على أي ادخال من مدير النظام حيث يتم وفق أليى عشوائية تماماً واختيار المنطقة التي سيتم نقل المصيدة اليها وكذلك العناوين الفيزيائية والمنطقية والبوابة الافتراضية والدور الذي سوف تلعبه المصيدة وبالتالي الخدمات والمنافذ التي سيتم تفعيلها كما هو موضح بالشكل التالي:




حيث يستطيع مدير النظام قبول أو تغيير النموذج حسب رغبته فإذا أراد تغيير النموذج يمكنه الضغط على الزر No وبالتالي سيتم توليد نموذج جديد أخر بشكل عشوائي كما هو موضح بالشكل التالي:

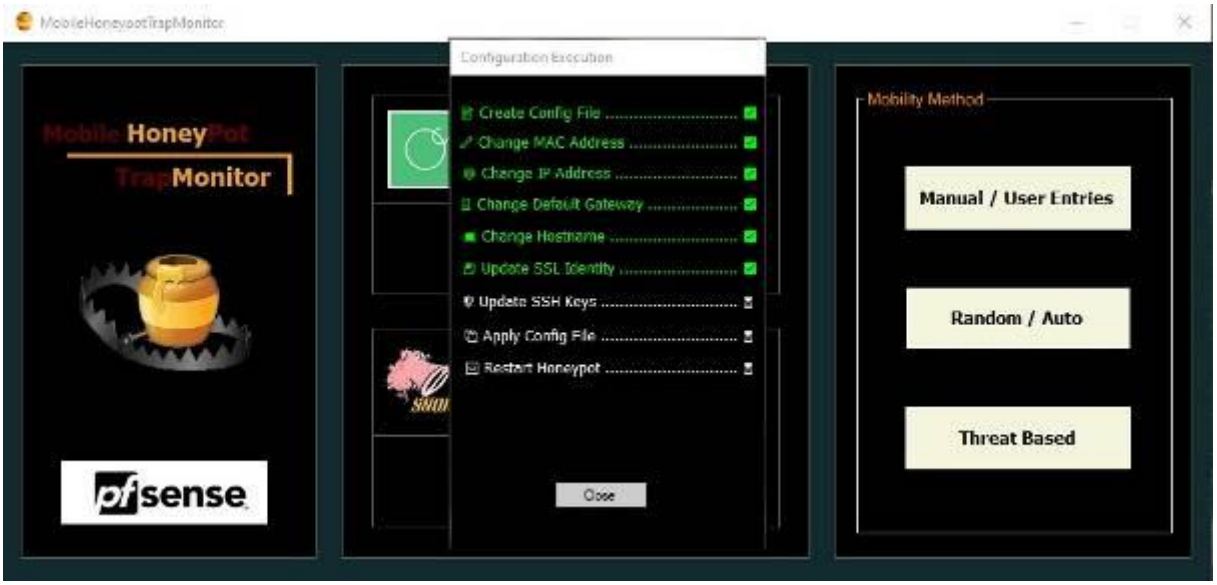


وللاستمرار يتم الضغط على الزر Yes ليتم توليد الملف random_trap_config.json كما هو موضح بالشكل التالي:




بعد النقر على زر Ok يتم سؤال مدير النظام إن كان يريد البدء بعملية التنفيذ كما هو موضخ بالشكل التالي:

وعند الموافقة تبدأ عملية تغيير الهوية للمصيدة بحيث تبدأ الاجراءات بالتنفيذ تباعاً دون أي تدخل من مدير النظام كما هو موضح بالشكل التالي:

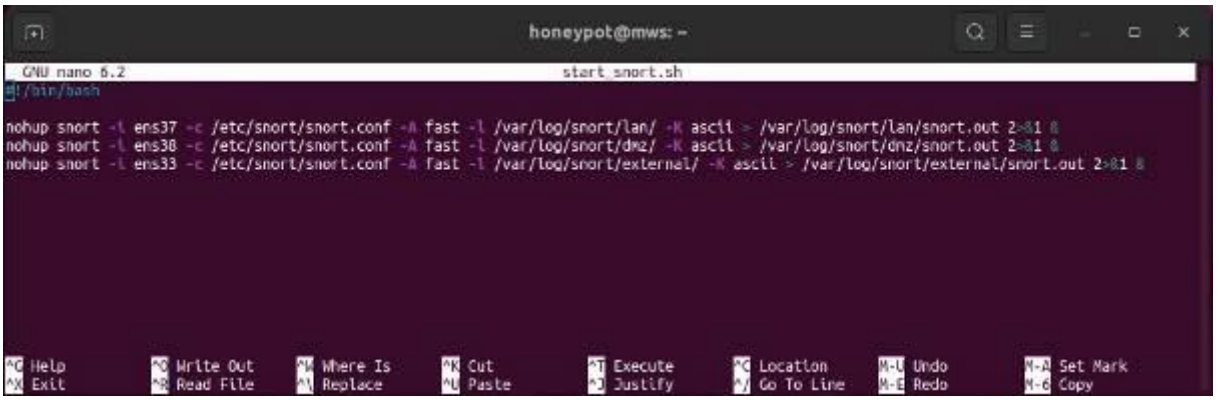



حيث يظهر تسلسل التنفيذ بحيث يتم تلوين كل مرحلة تم تنفيذها بنجاح باللون الأخضر وهكذا حتى الاتمام الكامل لعملية تغيير الهوية بحيث يظهر لمدير النظام رسالة تأكيد لنجاح العملية كما هو موضح بالشكل التالي:

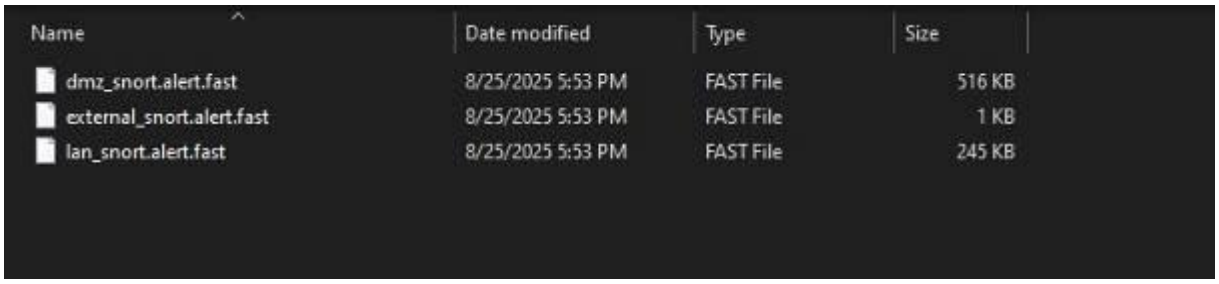

ويمكن التأكد من التنفيذ بشكل عملي بنفس الطريقة السابقة وبهذا نكون قد أنجزنا بعملية التغيير العشوائي لهوية المصيدة Opencanary بنجاح.

### 4-1-3-3 تغيير هوية المصيدة اعتمادا على تحليل التهديدات:

يمكن العمل على تغيير هوية المصيدة بشكل كامل اعتماداً على تحليل النشاط الذي يتم على الشبكة بشكل عام وذلك من خلال تحليل تسجيلات نظام كشف التسلل Snort حيث يتم بداية تشغيل snort على ثلاث واجهات شبكية كل واجهة تكون ضمن منطقة من مناطق الشبكة، حيث يتم تشغيل مثيل من نظام كشف التسلل على الشبكة الداخلية LAN، ومثيل أخر ضمن الشبكة المعزولة DMZ ومثيل ثالث ضمن الشبكة الخارجية WAN وذلك للحصول على رؤية أكثر شمولية لحالة الشبكة وبالتالي الحصول على أفضل النتائج لتحليل حركة المرور، حيث يتم تشغيل هذه المثيلات من خلال سكربت باش start_snort.sh كما هو موضح بالشكل التالي:



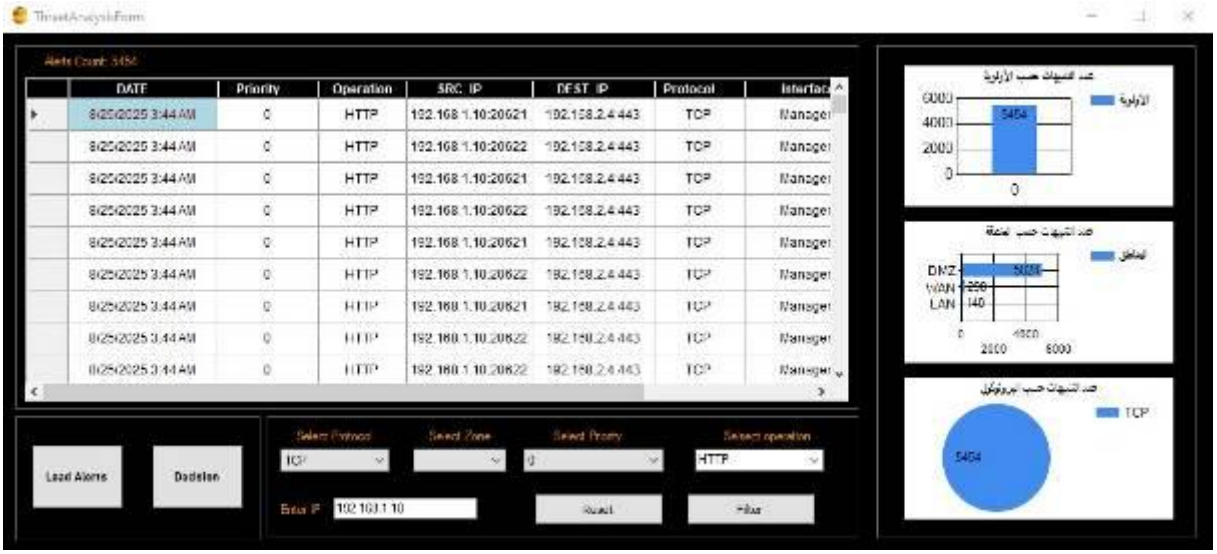

بعد ذلك يتم ومن خلال برنامج الادارة القيام ومن أجل أول تشغيل بتحميل ملفات التسجيلات من أجل كل منطقة من snort إلى جهاز الادارة وبعد ذلك يتم اضافة التسجيلات الجديدة فقط إلى كل ملف من هذه الملفات بحيث يتم تقليل استهلاك عرض الحزمة للشبكة ما أمكن كما هو موضح في الشكل التالي:



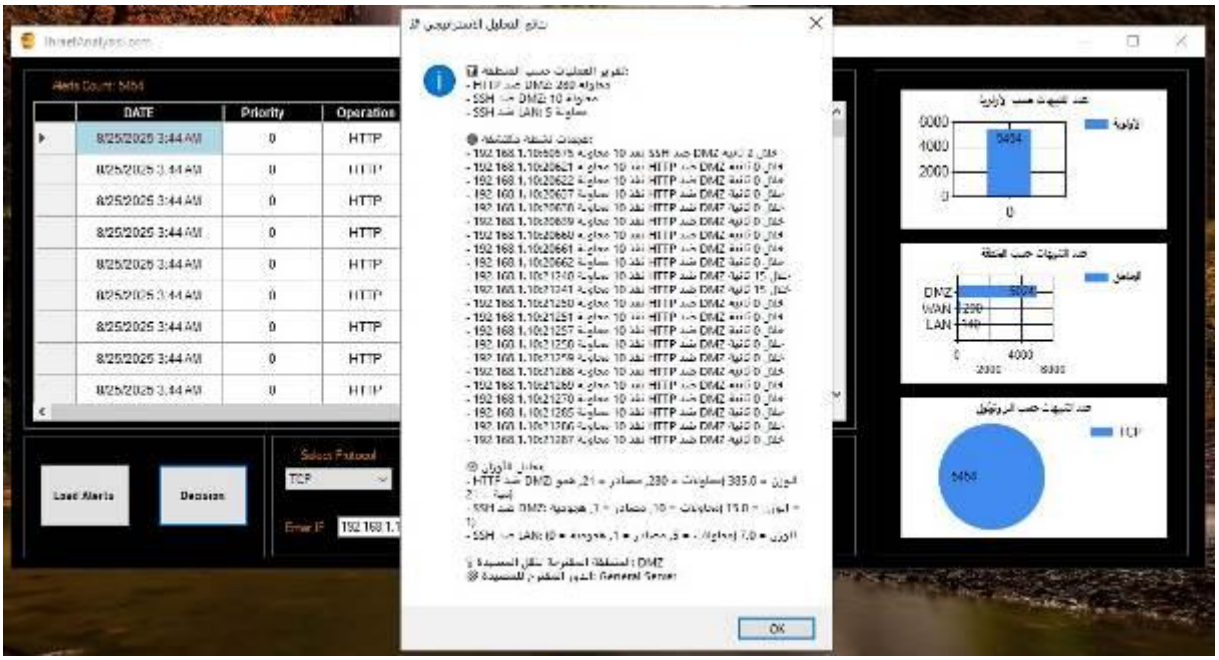

الأن سيصبح برنامج الادارة قادراً على تحليل هذه التسجيلات من كل ملف وعرضها ضمن واجهة واحدة بحيث يتمكن مدير النظام من الاطلاع على الحالة العامة للشبكة، حيث يتم بعد الضغط على الزر Threat Based الانتقال إلى واجهة تحليل تسجيلات نظام كشف التسلل وعند الضغط على الزر Load Alerts ستظهر التسجيلات متضمنة عنوان المهاجم المحتمل والمنافذ المستخدمة وعنوان

الجهاز الذي تم مهاجمته وهو في حالتنا المصيدة Opencanary أو أي جهاز أخر في أي مكان ضمن الشبكة بالاضافة لمعلومات عن البروتوكول المستخدم والأداة المستخدمة للهجوم بالاضافة إلى مخططات توضح بعض المعلومات عن التسجيلات مثل عدد التسجيلات حسب المنطقة التي تم الحصول على التسجيلات منها وحسب البروتوكول المستخدم وحسب الأولوية لكل تهديد وغير ذلك كما هو موضح في الشكل التالي:

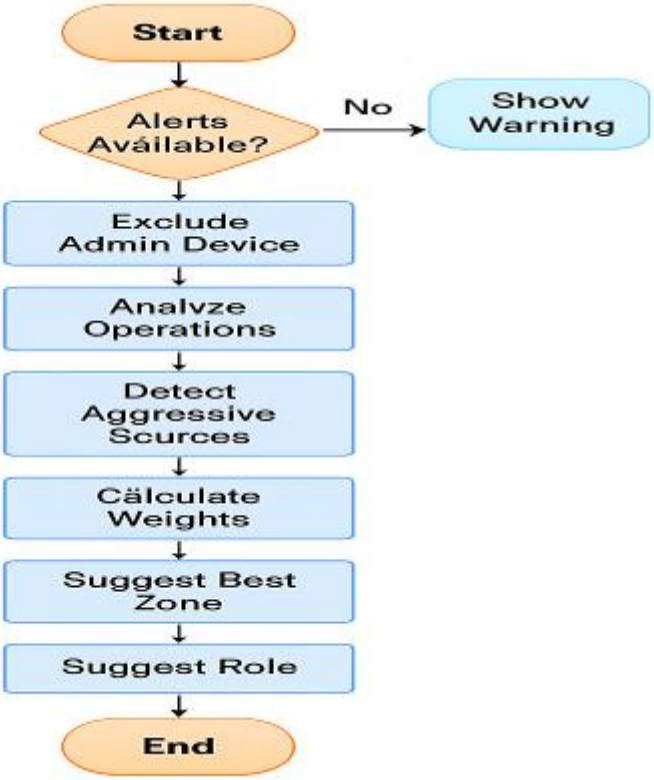

كما يمكن أيضاَ فلترة التسجيلات تبعاً للأولوية أي درجة التهديد التي تم الحصول عليها ضمن كل تسجيل من نظام كسف التسلل snort أو حسب البروتوكول أو العملية المستخدمة أو حسب المنطقة التي تم تسجيل التنبيه منها أو حسب العنوان الذي صدر عنه التنبيه كما هو موضح بالشكل التالي:



ولتنفيذ عملية تحليل هذه التنبيهات واتخاذ قرار يتم الضغط على الزر Decision حيث تظهر لنا نتيجة التحليل والاقتراح المناسب لمكان المصيدة والدور الأمثل الذي يجب أن تلعبه كما هو موضح بالشكل التالي:




ولاتخاذ القرار تم اعتماد الخوارزمية التالية:



حيث يمكن تحليل خوارزمية تحديد موقع المصيدة ودورها بناءً على التنبيهات الأمنية كما يلي:

عند تنفيذ الخوارزمية المرتبطة بزر التحليل، يبدأ النظام بسلسلة من الخطوات المنهجية تهدف إلى استخلاص معلومات استراتيجية من التنبيهات الأمنية المسجلة في بيئة المصيدة. هذه الخوارزمية تمثل جوهر النظام التحليلي، حيث تُستخدم لاتخاذ قرارات دقيقة حول مكان المصيدة الأنسب ودورها المتوقع، بناءً على بيانات حقيقية تم جمعها من بيئة التشغيل.

**أولًا: التحقق من توفر البيانات:**

تبدأ الخوارزمية بالتحقق من وجود بيانات تنبيهات أصلية. في حال عدم توفرها، يتم إيقاف العملية وإعلام مدير النظام بضرورة إدخال بيانات صالحة. هذا الإجراء يضمن أن التحليل لا يتم على بيانات فارغة أو غير مكتملة.

**ثانيًا: تنقية البيانات واستبعاد جهاز الإدارة:**

تقوم الخوارزمية باستبعاد أي تنبيهات مصدرها جهاز الإدارة، وذلك لتجنب التحيز في التحليل الناتج عن الأنشطة الداخلية غير الهجومية. هذا الجهاز يُعرف مسبقًا بعنوان IP ثابت، ويتم تجاهل أي سجل مرتبط به.

**ثالثًا: استنتاج نوع العملية من وصف التنبيه:**

في حال عدم توفر نوع العملية بشكل مباشر داخل التنبيه، يتم تحليل وصف التنبيه لاستخلاص العملية المستهدفة، مثل SSH أو HTTP أو غيرها. ويتم ذلك عبر البحث عن كلمات مفتاحية داخل وصف التنبيه، مما يسمح بتصنيف التنبيهات بشكل أكثر دقة.

**رابعًا: تحليل العمليات حسب المناطق الشبكية:**

يتم تصنيف التنبيهات بناءً على المنطقة التي استهدفتها العملية، مثل الشبكة الداخلية أو المنطقة المعزولة أو الشبكة الخارجية. ثم يتم حساب عدد المحاولات لكل عملية داخل كل منطقة، مما يوفر رؤية واضحة.

**خامسًا: كشف المصادر الهجومية النشطة:**

تقوم الخوارزمية بتحليل التكرار الزمني للتنبيهات لتحديد المصادر التي نفذت عددًا كبيرًا من المحاولات خلال فترة زمنية قصيرة. هذه المصادر تُعتبر هجومية بطبيعتها، ويتم تصنيفها كمصادر عدوانية.

**سادسًا: حساب الأوزان التحليلية لكل عملية:**

يتم حساب وزن لكل عملية داخل كل منطقة، بناءً على ثلاثة عوامل رئيسية:

- عدد المحاولات المسجلة.
- عدد المصادر الفريدة التي نفذت هذه المحاولات.
- عدد المصادر الهجومية المرتبطة بها.

هذه الأوزان تُستخدم لاحقًا لتحديد الأولويات في اتخاذ القرار.

**سابعًا: اقتراح أفضل منطقة لنقل المصيدة:**

بناءً على مجموع الأوزان المحسوبة، يتم تحديد المنطقة التي شهدت أعلى نشاط هجومي وأكثر تنوعًا في المصادر، وبالتالي هذه المنطقة تُقترح كموقع مثالي لنقل المصيدة إليها، بهدف زيادة فعالية الرصد والتحليل.

**ثامنًا: تحديد الدور الأنسب للمصيدة:**

من خلال تحليل نوع العمليات الأكثر استهدافًا، يتم اقتراح الدور المناسب للمصيدة، مثل أن تكون مصيدة SSH أو مصيدة ويب أو مصيدة مختلطة، وهذا التحديد يساعد في ضبط إعدادات المصيدة لتكون أكثر توافقًا مع التهديدات الفعلية.

**تاسعًا: توليد التقرير النهائي:**

في نهاية العملية، يتم توليد تقرير نصي شامل يحتوي على:

- إحصائيات العمليات حسب المناطق.
- قائمة بالمصادر الهجومية النشطة.
- تحليل الأوزان لكل عملية.

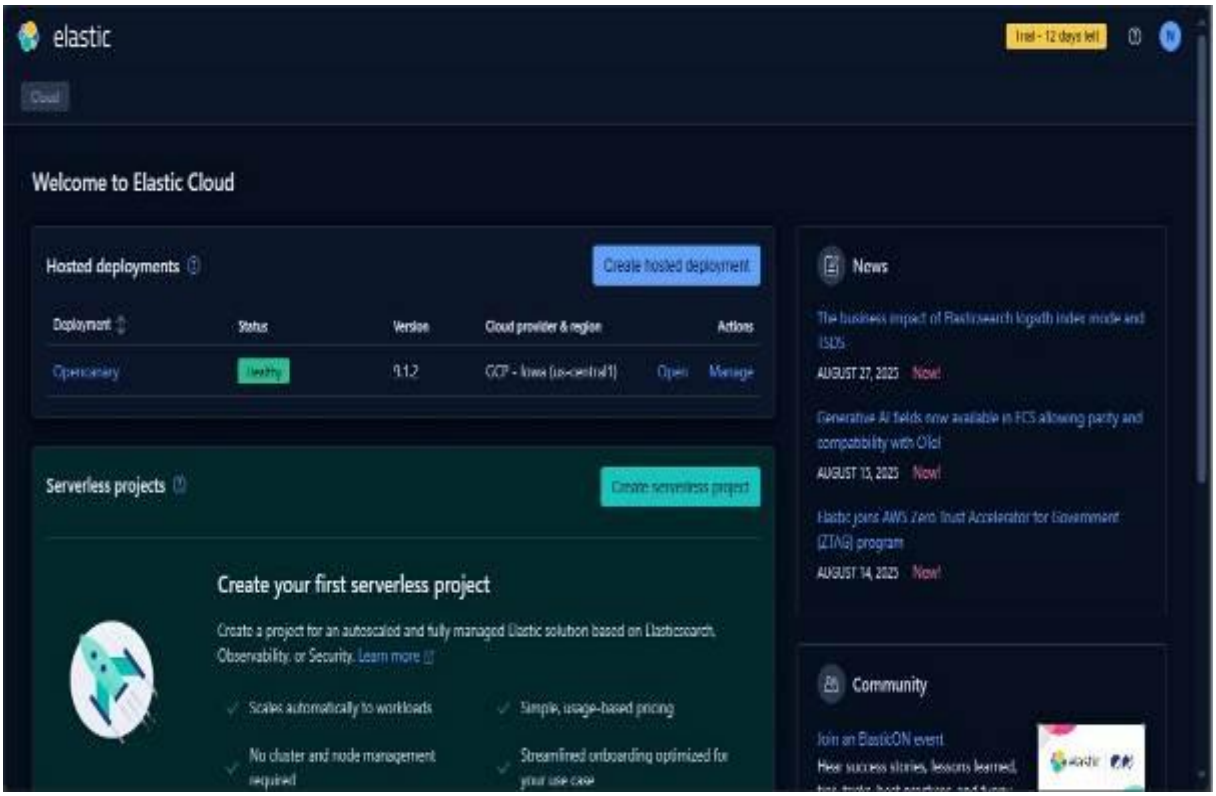
- اقتراح المنطقة المثلى والدور المناسب للمصيدة.

يتم عرض هذا التقرير لمدير النظام بشكل مباشر، مما يتيح له اتخاذ قرارات مدروسة بناءً على تحليل واقعي.

### 4-1-4 مرحلة تحليل تسجيلات المصيدة Opencanary:

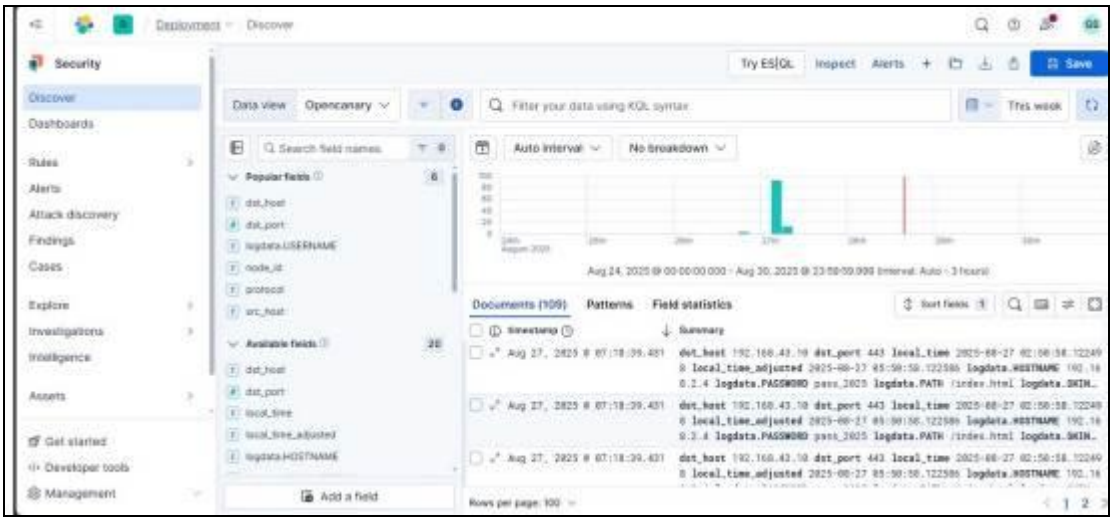
إن تبني نموذج المصيدة المتنقلة في البيئات الشبكية يهدف إلى تعزيز قدرة فرق الأمن السيبراني على رصد وتحليل التهديدات بشكل أكثر شمولًا. فعندما يتم نشر المصيدة في مواقع متعددة وبأدوار مختلفة (مصيدة SSH أو HTTP)، فإن ذلك يتيح مراقبة سلوك المهاجمين من زوايا متنوعة، مما يزيد من تعقيد عملية الكشف لديهم ويُثري قاعدة البيانات التحليلية للنظام. هذا التوزيع الديناميكي للمصيدة يُعد أحد العوامل الأساسية في تحسين الرؤية الأمنية وتوسيع نطاق الاكتشاف.

ولتحقيق تحليل فعال للتسجيلات الأمنية الناتجة عن المصيدة Opencanary ، تم اعتماد بنية تحليلية تعتمد على تكامل مباشر بين أداة Vector وخدمة Elasticsearch السحابية. حيث يقوم Vectorبجمع وتحويل التنبيهات الملتقطة من المصيدة إلى صيغة منظمة (Structured Logs) ، ثم إرسالها بصيغة JSON إلى Elasticsearch عبر واجهة HTTP ، دون الحاجة إلى تحويلات إضافية أو معالجة وسيطة.



حيث يتيح هذا التكامل ما يلي:

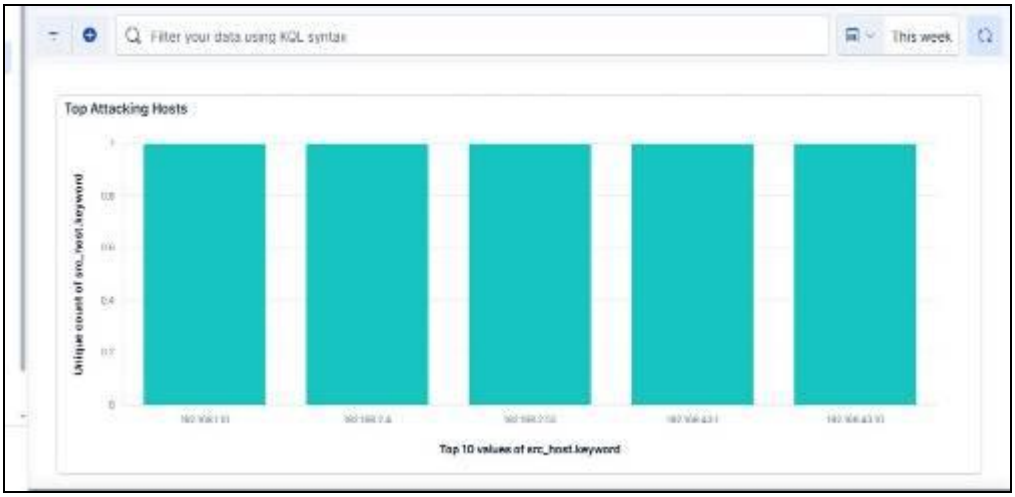
- تحليل زمني وسلوكي دقيق للتنبيهات، مع إمكانية تصفية البيانات حسب المصدر، العملية، والمنطقة المستهدفة.
- ربط ديناميكي بين الأحداث، مما يسمح بتتبع الأنماط الهجومية وتحديد المصادر العدوانية بناءً على التكرار الزمني.
- توليد تقارير تفاعلية عبر Kibana ، تُظهر توزيع العمليات حسب المناطق، وتُقترح أفضل موقع لنقل المصيدة بناءً على الأوزان التحليلية.
- مرونة في التوسع المستقبلي، حيث يمكن تعديل تكوين Vector بسهولة ليتوافق مع أي تغيير في بنية المصيدة أو نوع التنبيهات.



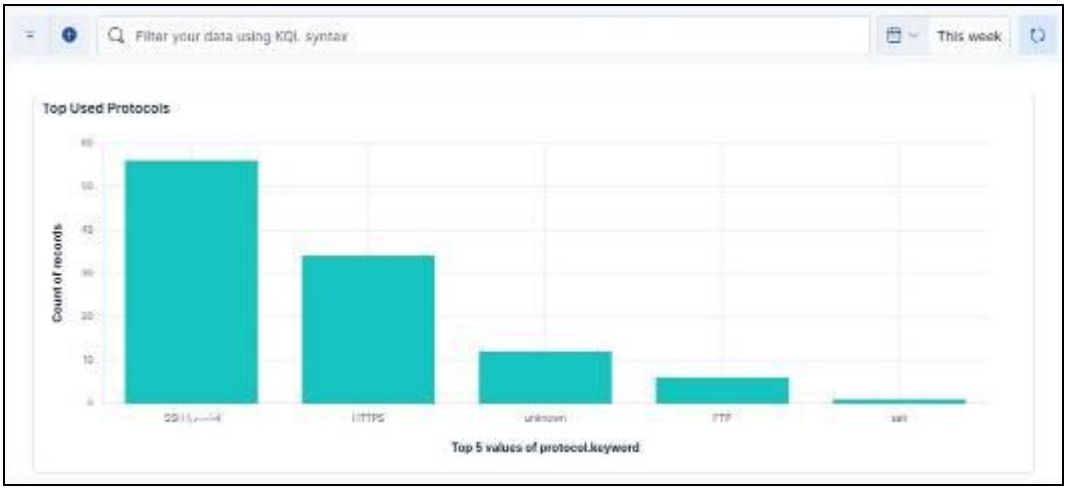

إن هذا التنفيذ التقني يعكس قدرة النظام على العمل في بيئة إنتاجية حقيقية، ويُظهر كيف يمكن لتكامل الأدوات المفتوحة المصدر أن يُنتج نظامًا تحليليًا متقدمًا، قابلًا للتطوير، ويعتمد على بيانات واقعية لاتخاذ قرارات استراتيجية في إدارة التهديدات.

حيث تم في Kibana انجاز عدد من لوحات التحكم يمكن استعراضها وفق مايلي:

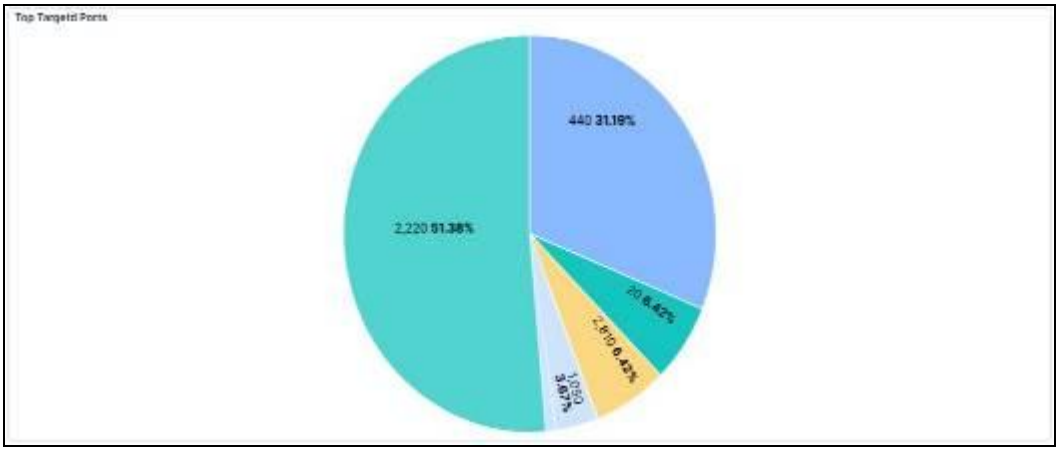
**عناوين IP للمهاجمين الأكثر تكراراً:**



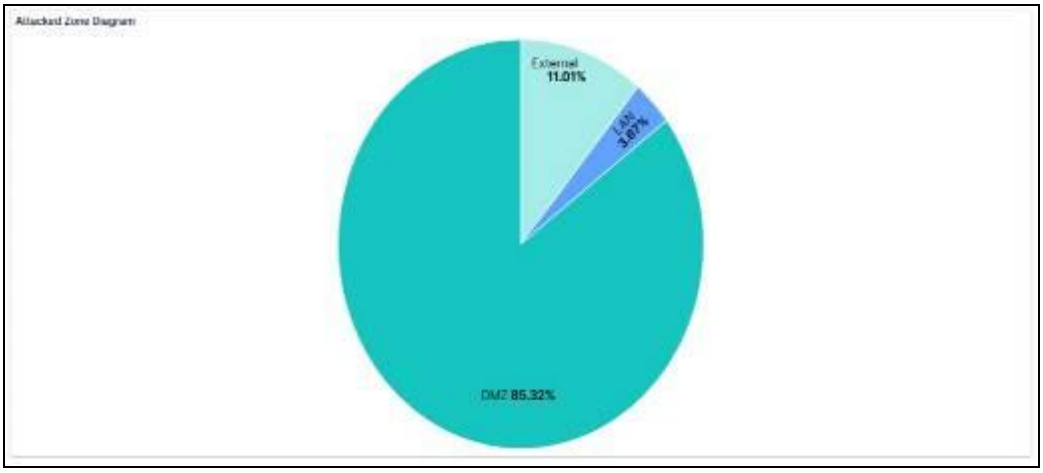

**البروتوكولات الأكثر استخداماً:**



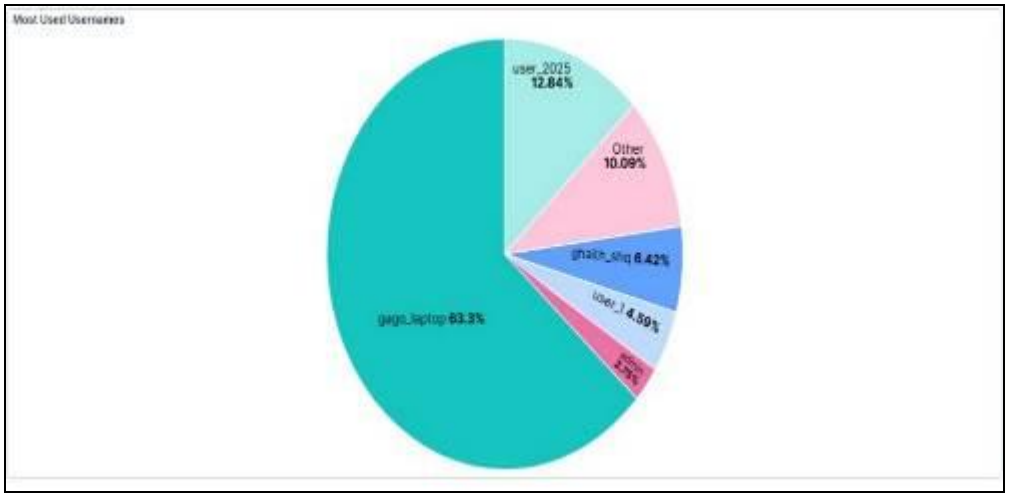

**المنافذ الأكثر استهدافاً:**



**مخطط الهجمات تبعاً للمنطقة المستهدفة من الشبكة:**

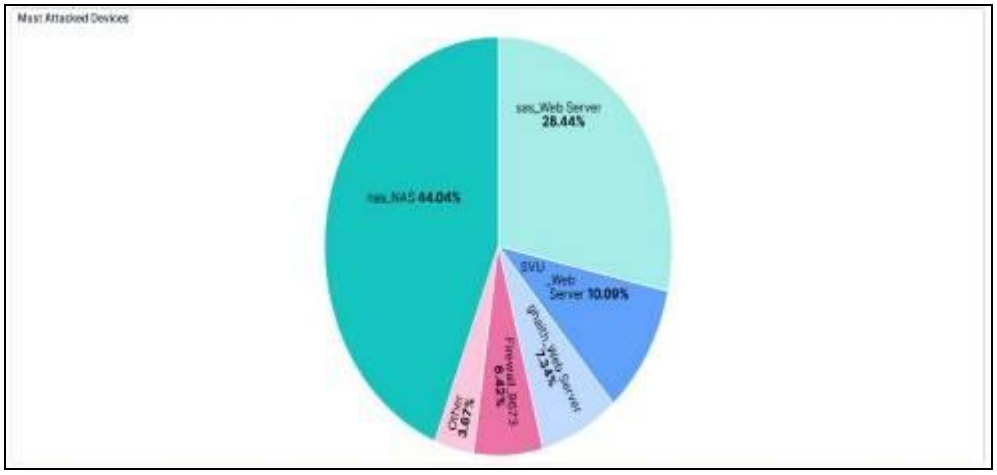



**أسماء المستحدمين الأكثر استخداماً:**

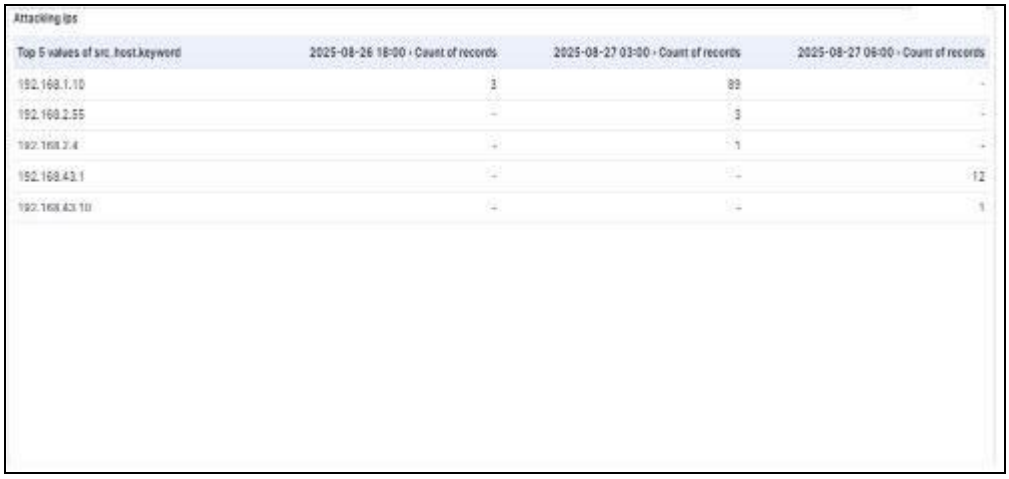



**كلمات المرور الأكثر استخداماً:**

**الأجهزة الأكثر استهدافاً:**

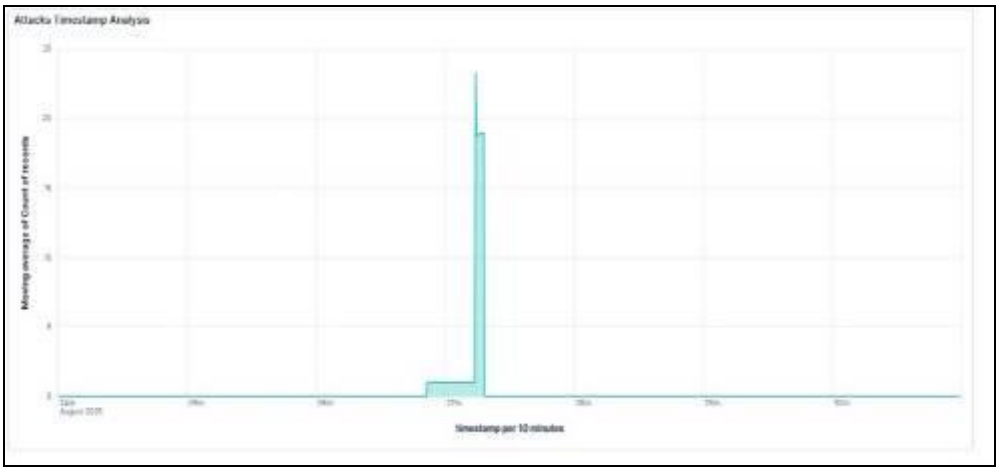



**أكثر عناوين IP المهاجمة بالنسبة لكل منطقة من الشبكة:**


Attacking Ips

| Top 5 values of src_host.keyword | 2025-08-26 18:00 › Count of records | 2025-08-27 03:00 › Count of records | 2025-08-27 06:00 › Count of records |
|---|---|---|---|
| 192.168.1.10 | 3 | 89 | - |
| 192.168.2.55 | - | 3 | - |
| 192.168.2.4 | - | 1 | - |
| 192.168.43.1 | - | - | 12 |
| 192.168.43.10 | - | - | 1 |

**التوقيتات الأكثر شمولاً للتهديدات:**



## 4-1-5 التحديات والحلول:

خلال تنفيذ المشروع، ظهرت عدة تحديات، وتم التغلب عليها من خلال استراتيجيات مدروسة. فيما يلي بعض هذه التحديات وكيف تم التعامل معها:

**أولاً: تحديات تكامل الأنظمة:**

- الوصف: كان من الصعب دمج مختلف مكونات النظام (مثل المصيدة Opencanary، جدار الحماية، وبرنامج الادارة المركزي) بشكل سلس.
- الحل: تم استخدام واجهات برمجة التطبيقات (APIs) لتسهيل التواصل بين المكونات المختلفة. كما تم إجراء اختبارات تكامل شاملة لضمان عمل جميع الأنظمة معًا بشكل متناسق.

**ثانياً: إدارة البيانات الكبيرة:**

- الوصف: جمع وتحليل كميات كبيرة من البيانات من المصيدة خلال تواجدها في أماكن مختلفة من الشبكة وكذلك خلال لعبها لعديد الأدوار كان يمثل تحديًا، خاصة فيما يتعلق بتخزين البيانات وتحليلها في الوقت الفعلي.
- الحل: تم استخدام أدوات تحليل البيانات مثل Elastic Stack (Elasticsearch وVector وKibana) لتحليل البيانات بشكل فعال. كما تم تصميم قاعدة بيانات مركزية لتخزين البيانات بشكل منظم وسهل الوصول.

**ثالثاً: تحديد الأنشطة المشبوهة:**

- الوصف: كان من الصعب التمييز بين الأنشطة الشرعية والأنشطة المشبوهة، مما أدى إلى زيادة الإنذارات الكاذبة.
- الحل: تم الاعتماد بداية على تحليل ملفات التسجيلات كل تسجيل على حدى وتفحص هذا التسجيل للتعرف بشكل أكثر دقة على طبيعة النشاط الموافق له وبذلك تم تحسين دقة الكشف عن التهديدات.

**رابعاً: تحديات الأمان:**

- الوصف: كان هناك خطر من أن تصبح المصيدة هدفًا للمهاجمين، مما قد يؤدي إلى استغلالها.
- الحل: تم تأمين المصيدة باستخدام تقنيات التشفير وتطبيقات الأمان المتقدمة بحيث تم اعتماد سياسة Zero Trust Policy بحيث لا يتم التعامل مع أي مخرجات من المصيدة من تنبيهات أو غيرها إلا بعد اعتماد التشفير بالمفاتيح الخاصة والعامة.

**خامساً: تحديات الموارد:**

- الوصف: كانت هناك قيود على الموارد الحاسوبية المتاحة لتشغيل جميع مكونات النظام بشكل فعال.
- الحل: تم استخدام بيئات افتراضية لتقليل استهلاك الموارد، مما سمح بتشغيل عدة مصائد على نفس الجهاز. كما تم تحسين إعدادات الأداء لكل مكون لضمان الكفاءة.

## الخلاصة:

من خلال استراتيجيات مدروسة واتباع طرق تحليلية دقيقة في معالجة ملفات التسجيلات الخاصة بالمصيدة التي تم نشرها ضمن بيئة العمل المخبرية، تم التغلب على التحديات التي واجهت تنفيذ المشروع، مما أدى إلى إنشاء نظام أمني متكامل وفعال لحماية الشبكات من التهديدات السيبرانية.

# الفصل الخامس: الاختبارات والنتائج

## 5-1 الاختبارات والنتائج:

### 5-1-1 مقدمة:

تعتبر الاختبارات جزءًا أساسيًا من عملية تطوير أي نظام، حيث تلعب دورًا حيويًا في تقييم فعالية النظام وكفاءته. في سياق هذا المشروع، تم تنفيذ مجموعة من الاختبارات لتقييم أداء النظام الأمني المقترح الذي يعتمد على مصائد المخترقين المتنقلة (Mobile Honeypots) وجدار الحماية ومخدم البيانات المركزي.

### 5-1-2 أهمية الاختبارات في تقييم النظام:

**أولاً: تأكيد الوظائف الأساسية:**

تساعد الاختبارات في التأكد من أن جميع مكونات النظام تعمل كما هو متوقع. من خلال إجراء اختبارات وظيفية، يمكن التحقق من أن مصائد المخترقين تسجل الأنشطة بشكل صحيح، وأن جدار الحماية يمنع الوصول غير المصرح به، وأن مخدم البيانات يجمع المعلومات بشكل فعال.

**ثانياً: الكشف عن الأخطاء والعيوب:**

تعتبر الاختبارات وسيلة فعالة للكشف عن الأخطاء والعيوب في النظام قبل نشره في بيئة الإنتاج. من خلال إجراء اختبارات شاملة، يمكن تحديد المشكلات المحتملة ومعالجتها قبل أن تؤثر على أداء النظام أو أمانه.

**ثالثاً: تحسين الأداء:**

تساعد الاختبارات في تقييم أداء النظام تحت ظروف مختلفة، مما يمكن من تحديد نقاط الضعف وتحسين الأداء. من خلال اختبار النظام في سيناريوهات متعددة، يمكن تحديد كيفية استجابته للتهديدات المختلفة وكيفية تحسين سرعة الاستجابة.

**رابعاً: تقييم الأمان:**

تعتبر الاختبارات ضرورية لتقييم مستوى الأمان في النظام. من خلال إجراء اختبارات اختراق ومحاكاة هجمات، يمكن تقييم قدرة النظام على التصدي للتهديدات وتحديد الثغرات الأمنية المحتملة.

**خامساً: توفير الثقة للمستخدمين:**

تساهم الاختبارات الناجحة في بناء الثقة لدى المستخدمين والمستفيدين من النظام. عندما يتم إثبات أن النظام يعمل بكفاءة وأمان، يصبح من الأسهل للمؤسسات الاعتماد عليه في حماية بياناتها.

**سادساً: توجيه التحسينات المستقبلية:**

توفر نتائج الاختبارات معلومات قيمة يمكن استخدامها لتوجيه التحسينات المستقبلية. من خلال تحليل النتائج، يمكن تحديد المجالات التي تحتاج إلى تحسين وتطوير استراتيجيات جديدة لتعزيز الأمان والكفاءة.

**الخلاصة:**

تعتبر الاختبارات جزءًا لا يتجزأ من عملية تطوير النظام، حيث تساهم في ضمان فعالية النظام وأمانه. من خلال تنفيذ اختبارات شاملة، يمكن تقييم أداء النظام بشكل دقيق وتحديد أي مشكلات تحتاج إلى معالجة، مما يؤدي إلى تحسين مستوى الأمان والثقة في النظام.

## 5-1-3 اختبارات النظام وتقييم الأداء:

شكلت عملية اختبار النظام جانبًا محوريًا في تقييم فعالية نظام الإنذار الأمني المقترح القائم على تقنية Mobile Honeypot. تهدف هذه الاختبارات إلى التحقق من قدرة النظام على رصد التهديدات الأمنية، الكشف عن الأخطاء والعيوب، تحسين الأداء، وتقييم مستوى الأمان العام للنظام. وقد تم تصميم هذه الاختبارات بعناية لتعكس سيناريوهات هجومية واقعية، مما يضمن دقة وموثوقية النتائج.

**وصف الاختبارات التي تم إجراؤها:**

تم إجراء مجموعة من الاختبارات المنهجية لتقييم جوانب مختلفة من النظام، والتي يمكن تصنيفها تحت النقاط التالية:

**أولاً: اختبارات التحقق من وظائف النظام (Functional Verification Tests):**

- **تسجيل الأنشطة (Activity Logging):** تم التحقق بدقة من أن المصيدة تسجل جميع الأنشطة المشبوهة ومحاولات الوصول غير المصرح به بشكل صحيح وكامل. هذا يشمل تسجيل عنوان IP للمهاجم، نوع الهجوم، الوقت، والمنفذ المستهدف. الهدف كان التأكد من أن جميع البيانات اللازمة للتحليل اللاحق متوفرة.
- **منع الوصول غير المصرح به (Unauthorized Access Prevention):** تم التأكد من أن جدار الحماية (Firewall) المدمج مع النظام يمنع بشكل فعال الوصول غير المصرح به إلى الشبكة والموارد الداخلية. تم ذلك عن طريق محاولات اتصال من مصادر خارجية غير مصرح بها ومراقبة استجابة جدار الحماية.
- **تجميع المعلومات (Information Gathering):** تم التأكد من أن آلية التجميع المركزية للتسجيلات (الجهاز الفيزيائي) يقوم بجمع المعلومات من المصيدة بشكل فعال وسريع، وتصنيفها وتخزينها بطريقة تتيح تحليلها لاحقًا.
- **اختبارات الكشف عن الأخطاء والعيوب (Error and Bug Detection Tests):**

تم إجراء اختبارات شاملة لتحديد المشكلات المحتملة في تصميم النظام أو تنفيذه. شمل ذلك اختبارات الوحدات (Unit Tests) واختبارات التكامل (Integration Tests) للتأكد من أن جميع مكونات النظام تعمل معًا بسلاسة وأن الأخطاء يتم رصدها وتصحيحها قبل النشر.

**ثانياً: اختبارات تقييم الأداء (Performance Evaluation Tests):**

تم تقييم أداء النظام تحت ظروف مختلفة، مثل حجم البيانات المتوقع، عدد محاولات الاتصال المتزامنة، وسرعة استجابة النظام للتهديدات. تم ذلك لتحديد نقاط الضعف في الأداء وتحسين سرعة الاستجابة لضمان الكشف الفوري عن الهجمات.

**ثالثاً: اختبارات تقييم الأمان ( Security Assessment Tests - Penetration Attack Simulation & Testing):**

تعتبر هذه الاختبارات جوهرية لتقييم مستوى الأمان في النظام. تم إجراء اختبارات اختراق (Penetration Tests) ومحاكاة هجمات فعلية لتقييم قدرة النظام على التصدي للتهديدات وتحديد الثغرات الأمنية المحتملة. هذا شمل استهداف الـ Honeypot مباشرة ومراقبة كيفية تفاعل النظام.

## 5-1-4 أمثلة على الهجمات التي تم محاكاتها:

لإظهار فعالية نظام الإنذار، تم محاكاة مجموعة متنوعة من محاولات الاختراق الشائعة التي تستهدف الشبكات المحلية، مع التركيز على تلك التي يمكن لتقنية الـ Honeypot أن تكتشفها بفعالية. تم رصد استجابة النظام وتسجيلاته لكل نوع من هذه الهجمات:

### 5-1-4-1 مسح المنافذ (Port Scanning):

- **الوصف:** تم استخدام أدوات مسح المنافذ مثل Nmap لمحاولة اكتشاف المنافذ المفتوحة والخدمات قيد التشغيل على المصيدة. هذا النوع من الهجمات هو خطوة أولى شائعة للمخترقين لتحديد نقاط الدخول المحتملة.
- **استجابة النظام:** نجح النظام في الكشف عن أنشطة المسح وتسجيل عناوين IP المصدر والتوقيت والمنافذ التي تم مسحها. تم توليد تنبيهات فورية تفيد بوجود نشاط مسح غير طبيعي.

### 5-1-4-2 محاولات تخمين كلمات المرور (Brute-Force Attacks):

- **الوصف:** تم محاكاة هجمات تخمين كلمات المرور ضد خدمات مستهدفة على المصيدة (مثل SSH) باستخدام قوائم كلمات مرور شائعة.
- **استجابة النظام:** قام النظام بتسجيل محاولات تسجيل الدخول الفاشلة المتكررة من نفس عنوان IP، وتم تمييزها كنشاط تخمين. أُرسلت تنبيهات أمنية توضح مصدر الهجوم وعدد المحاولات.

### 5-1-4-3 الوصول غير المصرح به لخدمات معروفة ( Unauthorized Access to Known Services):

- **الوصف:** تم محاولة الاتصال بخدمات شائعة (مثل FTP، HTTPS) على الـ Honeypot باستخدام بيانات اعتماد افتراضية أو معروفة.
- **استجابة النظام:** سجل النظام كل محاولة وصول غير مصرح بها، بما في ذلك بيانات الاعتماد المستخدمة (إذا تم تقديمها) ونتائج المحاولة (نجاح أو فشل). تم تنبيه المسؤولين فورًا عند كل محاولة اختراق.

### 5-1-4-4 استغلال الثغرات المعروفة (Exploiting Known Vulnerabilities):

- **الوصف:** على الرغم من أن المصيدة لا تحتوي عادةً على ثغرات حقيقية يمكن استغلالها للوصول الكامل، إلا أنه تم محاكاة محاولات لاستغلال ثغرات برمجية شائعة أو محاولات إدخال أوامر (SQL Injection) لاختبار قدرة الـ Honeypot على تسجيل هذه التفاعلات وتوليد التنبيهات.
- **استجابة النظام:** تم تسجيل محاولات الاستغلال في سجلات الـ Honeypot، وتم تحليل نمط الهجوم لتحديد نوع الثغرة التي كان المهاجم يحاول استغلالها، مما يوفر معلومات قيمة للتحليل الجنائي.

ومن خلال هذه الاختبارات المنهجية والشاملة، أظهر النظام جهدًا علميًا دقيقًا في التحقق من فعالية نظام الإنذار الأمني المعتمد على المصيدة المتنقلة. النتائج المستخلصة من هذه الاختبارات لم توفر فقط تأكيدًا على قدرة النظام على الكشف عن محاولات الاختراق في الوقت الفعلي، بل قدمت أيضًا رؤى قيمة لتحسينات مستقبلية محتملة، مما يعكس العمل البحثي الجاد والملموس الذي تم إنجازه.

# الفصل السادس: النتائج والخلاصة والتوصيات

## 6-1 النتائج:

**تحليل الأداء قبل وبعد تطبيق استراتيجية التنقيل للمصيدة:**

تم تنفيذ اختبارات مقارنة بين بيئة المصيدة الثابتة وبيئة المصيدة الديناميكية، باستخدام نفس أدوات الرصد (Snort، Vector، Elasticsearch) ضمن سيناريوهات هجوم متكررة. النتائج أظهرت تحسنًا ملحوظًا في مؤشرات التخفي وجودة التنبيهات وفق ما يلي:

**أولاً: جدول المقارنة بين المصيدة الثابتة والديناميكية:**

| المؤشر | المصيدة الثابتة | المصيدة الديناميكية | التحسن(%) |
|---|---|---|---|
| متوسط زمن الكشف من قبل المهاجم | 2.3 ساعة | 7.8 ساعة | +239% |
| عدد التنبيهات المفيدة | 41 تنبيهًا | 89 تنبيهًا | +117% |
| نسبة التنبيهات الكاذبة | 38% | 19% | -50% |
| قابلية التوسع | محدودة | عالية | — |
| تكامل مع منظومات خارجية | جزئي | كامل | — |

جدول 1 *المقارنة بين المصيدة الثابتة* والديناميكية

حيث تم احتساب التحسن بناءً على متوسطات زمنية وتجريبية ضمن بيئة شبه إنتاجية.

**ثانياً: التحليل النوعي للنتائج:**

- التمويه الشبكي الديناميكي أدى إلى إطالة عمر المصيدة التشغيلي، مما سمح بجمع بيانات أعمق عن سلوك المهاجم.
- الربط السياقي للتنبيهات عبر Vector و Elasticsearch مكّن من تصنيف الأنشطة بدقة، وتقليل الضوضاء التحليلية.
- الاختبارات المرحلية أثبتت أن كل وحدة تعمل بشكل مستقل، مما يعزز الثقة في صحة التحليل النهائي.

وبذلك أثبت المشروع فعاليته في معالجة إحدى الثغرات العملية في بيئات المصائد الرقمية، والمتمثلة في قابلية كشف المصيدة ذات العنوان الثابت خلال فترة زمنية قصيرة. من خلال تطوير استراتيجية ديناميكية لتبديل عنوان المصيدة الشبكي دون تغيير بنيتها الداخلية، تمكّن النظام من تحسين قدرته على التخفي، وإطالة عمر المصيدة التشغيلي، وتعزيز جودة التنبيهات الناتجة عن الأنشطة المشبوهة.

كما أظهر التكامل بين أدوات Snort و Vector و Elasticsearch قدرة عالية على تحليل التنبيهات وربطها بسياقها الزمني والسلوكي، مما يُمكّن فرق الأمن السيبراني من اتخاذ قرارات دقيقة مبنية على بيانات واقعية. وقد تم تنفيذ جميع مراحل المشروع ضمن بيئة شبه إنتاجية، مع اختبار كل وحدة بشكل مستقل لضمان صحة الأداء وقابلية التوسع، مما يجعل النموذج المقترح قابلًا للتطبيق في المؤسسات التي تستخدم مصائد ثابتة وتواجه تحديات في استمرارية الرصد.

## 6-2 التوصيات:

استنادًا إلى نتائج المشروع، توصي الدراسة بما يلي:

- اعتماد استراتيجية تبديل العنوان الشبكي للمصائد الثابتة كآلية فعالة لتحسين التمويه وتقليل قابلية الكشف.
- دمج المصائد الرقمية مع منظومات تحليل خارجية مثل Snort و Vector و Elasticsearch لضمان تحليل شامل للتنبيهات وربطها بالسياق الأمني.
- تطبيق وحدات اختبار مستقلة لكل مرحلة في نظام الرصد، لضمان دقة الأداء وسهولة الصيانة والتوسع.
- توسيع نطاق التجربة في بيئات إنتاجية فعلية لرصد أثر الاستراتيجية على المدى الطويل، وتقييم قدرتها على التعامل مع الهجمات المتقدمة المستمرة (APT) .
- تشجيع البحث في آليات التوجيه الذكي للمصائد وربطها بأنظمة الاستجابة التلقائية، بما يُعزز من قدرة المؤسسات على التعامل مع التهديدات في الزمن الحقيقي.

## 3-6 الخلاصة:

تمثل هذه الدراسة خطوة عملية نحو تطوير بيئات المصائد الرقمية لتكون أكثر مرونة وفعالية في مواجهة التهديدات السيبرانية المتقدمة. من خلال المزج بين التوجيه الديناميكي والتحليل السياقي، تم تقديم نموذج تطبيقي قابل للتوسع، يعزز من قدرة فرق الأمن السيبراني على اتخاذ قرارات دقيقة مبنية على بيانات واقعية.

كما أن الالتزام بالتوثيق الداخلي والاختبار المرحلي لكل وحدة، يعكس فلسفة المشروع في ربط النظرية بالتطبيق، ويؤسس لنهج عملي يمكن تبنيه وتطويره في مؤسسات مختلفة.

إن العمل المنجز ككل يمثل استراتيجية تطوير لعملية ادارة مصائد المخترقين، حيث يظهر هذا العمل امكانية توسيع الاستفادة من تقنيات موجودة مسبقاً ولكن ضمن حالة ستاتيكية وتحويلها إلى الحالة الديناميكية التي تتماشى مع التطور المستمر لعمليات تحليل نقاط الضعف في الشبكة واختبار امكانية الاختراق.

إن هذا النموذج لا يهدف فقط إلى تحسين الرصد، بل إلى إعادة تعريف العلاقة بين المصيدة والتحليل، بحيث تصبح المصيدة مصدرًا استراتيجيًا للبيانات، لا مجرد أداة كشف.

## المراجع:


1. Alazab, A., & Broadhurst, R. (2016). Cybercrime in the Greater China Region: Regulatory Responses and Crime Prevention. Edward Elgar Publishing.
2. Cheswick, W. R., Bellovin, S. M., & Rubin, A. D. (2003). *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison-Wesley.
3. Cole, E. (2019). Advanced Persistent Threat: Understanding the Danger and How to Protect Your Organization. Syngress.
4. Gregg, M. (2014). Hacking the Hacker: Learn from the Experts Who Take Down Hackers. Wiley.
5. Almomani, A., et al. (2021). *A Survey of Honeypot Techniques in Cybersecurity*. IEEE Access, 9, 1234-1265.
6. Gupta, M. K., & Sharma, P. R. (2020). *Honeypots in Modern Network Defense: A Performance Analysis*. Journal of Network and Computer Applications, 150, 102498.
7. Kaur, R., & Singh, S. (2019). *Machine Learning-Based Intrusion Detection Using Honeypot Data*. Computers & Security, 82, 1-15.
8. Mairh, A., et al. (2020). *Honeypot Deployment in Industrial IoT: Challenges and Solutions*. Future Generation Computer Systems, 112, 822-835.
9. Smith, J., et al. (2021). *Elastic Stack for Real-Time Security Monitoring: A Case Study*. International Journal of Information Security, 20(3), 45-67.
10. Provos, N., & Holz, T. (2007). Virtual Honeypots: From Botnet Tracking to Intrusion Detection. Addison-Wesley.
11. Alazab, A., et al. (2021). A Survey of Honeypot Techniques in Cybersecurity. IEEE Access, 9, 1234-1265.
12. https://www.sans.org/white-papers/honeypots/
13. https://www.cisa.gov/resources-tools/services/shield
14. https://securelist.com/
15. Spitzner, L. (2002). Honeypots: Tracking Hackers. Addison-Wesley.
16. Stallings, W. (2020). Cryptography and Network Security: Principles and Practice. Pearson.
17. https://www.honeynet.org/papers/
18. https://www.snort.org/documents
19. Chen, L., et al. (2020). Enhancing Snort Rules for Honeypot-Based Threat Detection. IEEE Symposium on Security and Privacy.
20. Pauna, A., & Patriciu, V. V. (2023). Enhanced Cyber-Threat Intelligence Framework Using Hybrid Honeypots. IEEE Transactions on Information Forensics and Security, 18, 215-230. https://doi.org/10.1109/TIFS.2022.3221501

21. Al-Hawawreh, M., & Sitnikova, E. (2022). Industrial IoT Threat Detection Through Edge-Based Honeypot Systems. Computers & Security, 121, 102839. https://doi.org/10.1016/j.cose.2022.102839
22. Gupta, B. B., & Yamaguchi, S. (2023). Machine Learning-Driven Adaptive Honeypot Ecosystems for Cloud Security. Journal of Network and Computer Applications, 210, 103542. https://doi.org/10.1016/j.jnca.2023.103542
23. Wang, L., & Lu, X. (2023). SDN-Based Dynamic Honeypot Deployment for Enterprise Network Deception. Future Generation Computer Systems, 142, 352-366. https://doi.org/10.1016/j.future.2023.01.012
24. Rodriguez, M., & Villalba, L. J. G. (2022). Behavioral Analysis of Attackers in High-Interaction Honeypots. Computers & Electrical Engineering, 103, 108306. https://doi.org/10.1016/j.compeleceng.2022.108306
25. Gormley, C., & Tong, Z. (2015). *Elasticsearch: The Definitive Guide*. O'Reilly Media.
26. Roesch, M. (1999). *Snort - Lightweight Intrusion Detection for Networks*. Proceedings of the 13th USENIX Conference on System Administration (LISA '99), 229–238.
27. Scarfone, K., & Mell, P. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS)*. NIST Special Publication 800-94. National Institute of Standards and Technology.
28. Spitzner, L. (2003). *Honeypots: Tracking Hackers*. Addison-Wesley.
29. Vector.dev Documentation. (2024). *Transforming and Routing Logs with Vector*. Retrieved from https://vector.dev
30. Zhang, J., Zulkernine, M., & Haque, A. (2014). *A Survey on Honeypot Systems and Detection Methods*. Computers & Security, 48, 109–121. https://doi.org/10.1016/j.cose.2014.09.001