B.M.S. COLLEGE OF ENGINEERING

Department of Computer Science and Engineering

Major Project Phase 1

(22CS7PWMP1)

Synopsis Report

On

HoneyGAN Pots: A Deep Learning Approach for Generating Honeypots

Submitted by:

Aastha Priya	**Aneesh K.P**	**Anup Vaidya**	**Araga Laxman Anirudhadithya**
1BM22CS050	1BM22CS050	1BM22CS050	1BM22CS050

Under the Guidance of:

Guide Name: Dr. Asha G.R

Designation: Associate Professor

INRODUCTION

Cybersecurity is an ever-evolving field that requires constant innovation to protect networks and systems from malicious actors. One widely used technique in cyber defense is the deployment of honeypots, which are decoy systems designed to attract, detect, and analyze cyber threats. Honeypots serve as a proactive security measure, providing valuable insights into attacker behavior and aiding in the development of more robust defense strategies. However, selecting and configuring effective honeypots remains a challenge due to the need for realistic and adaptive decoy environments.

This project explores the use of Generative Adversarial Networks (GANs) for dynamically generating realistic honeypot configurations. Traditional methods often rely on static lists or pre-configured images, which lack flexibility and scalability. In contrast, our approach leverages the learning capabilities of GANs to create diverse and authentic decoy configurations tailored to specific cyber defense needs. By training the model on real-world network device configurations, our system can generate honeypots that closely resemble genuine targets, thereby improving their effectiveness in deceiving attackers and enhancing threat intelligence collection.

To the best of our knowledge, this is the first attempt to apply GANs for honeypot generation. Our project aims to bridge the gap between AI-driven automation and cyber deception techniques, offering a novel solution that can dynamically adapt to evolving cyber threats. The findings from this work have the potential to significantly improve cybersecurity defenses by providing organizations with an automated and scalable method for deploying realistic honeypots.

OBJECTIVE

Objective of the PaperCybersecurity experts are continuously threatened by the need to protect networks from attackers. One of the common tactics employed is the implementation of honeypots, which are decoy systems aimed at luring and analyzing attackers. However, standard honeypot implementations call for pre-determined lists or stored images that are not adaptable and efcient.This work introduces a new method that utilizes Generative Adversarial Networks (GANs) to dynamically create honeypot confgurations. The aim is to present cyber defenders with a fexible and automated system that can produce realistic decoy environments without depending on manually maintained databases. In so doing, the research tackles major challenges like decoy choice, fexibility, and detectability resistance.In pursuit of this, the authors create three different kinds of GAN models:Unconditional GAN – Creates decoy confgurations without particular constraints.Conditional O/S GAN – Produces honeypots depending on operating system type (e.g., Windows, Linux, RouterOS).Conditional Device Type (DT) GAN – Creates decoys for particular device functions (e.g., web servers, routers, VPNs).The research assesses the accuracy and recall of the models and quantifes how well generated honeypots perform against detection tools. By making use of real-world device confgurations from Shodan, the method guarantees that decoys resemble actual network devices, thus becoming more attacker-resistant.Ultimately, this research will make the defense of cybersecurity stronger by offering an intelligent, scalable, and automated honeypot generation process, assisting organizations in detecting, deterring, and delaying cyber threats more efciently.

REQUIREMENTS

1. Hardware Requirements:

A. Local (On-Premise) Setup

- CPU: Intel i7/i9 or AMD Ryzen 7/9
- RAM: 16GB or more
- Storage: 500GB SSD
- GPU: NVIDIA RTX 3060+

B. Cloud (AWS/GCP/Azure) Setup

- Compute Instance: 4+ vCPUs, 16GB RAM
- GPU for ML Training: NVIDIA T4 / V100
- Storage: 500GB SSD
- Security: Cloud Firewall, AWS Shield, Cloudflare WAF

2. Software Requirements

A. Honeypot Deployment

- Cowrie – SSH/Telnet Honeypot
- Dionaea – Malware & Exploit Detection
- T-Pot – Multi-honeypot Framework

B. Machine Learning & GANs

- Python (3.8+)
- TensorFlow / PyTorch
- Scikit-learn, Pandas, NumPy
- Jupyter Notebook
- Matplotlib, Seaborn (for visualization)

C. Security & Network Monitoring

- Suricata / Snort – Intrusion Detection System
- Wireshark / Tcpdump – Packet Capture

D. Cloud Services & Deployment

- AWS EC2 / GCP Compute – Cloud Instance for Honeypot
- AWS S3 / GCS – Log Storage
- Kubernetes / Docker – Containerized Deployment

COST ESTIMATE

The cost estimation for the project has been defined in the table below with both the minimal cost required and the maximal cost it may potentially go up to for each category specific to it.

Component	Minimum Cost	Maximum Cost
AWS Cloud Services	₹2,000	₹7,000
Storage	₹1,500	₹5,000
AWS Cloud Computing for GAN	₹1,000	₹2,000
Total	₹4,500	₹14,000

WORKPLAN

The workplan to complete the project has been divided into a total of 7 phases, with the total duration of the project surmounting to an approximate of 8-9 months.

The following phases are mentioned below:

Phase 1: Project Initiation and Planning

Duration: 2 weeks

Objective: Establish the project foundation and gather resources quickly.

Phase 2: Data Collection and Preparation

Duration: 4 weeks

Objective: Collect and preprocess data efficiently.

Phase 3: GAN Architecture Design and Development

Duration: 5 weeks

Objective: Design and implement GAN models rapidly.

Phase 4: Training and Optimization

Duration: 8 weeks

Objective: Train GANs efficiently with real data.

Phase 5: Evaluation and Validation

Duration: 6 weeks

Objective: Assess GAN performance and honeypot quality quickly.

Phase 6: Documentation and Conclusion

Duration: 5 weeks

Objective: Compile results and paper efficiently.

Phase 7: Presentation and Dissemination

Duration: 3 weeks

Objective: Share findings with minimal delay.

REFERENCES

Gabrys, R., Silva, D., & Bilinski, M. (2024b, July 10). HoneyGAN Pots: a deep learning approach for generating honeypots. arXiv.org. https://arxiv.org/abs/2407.07292