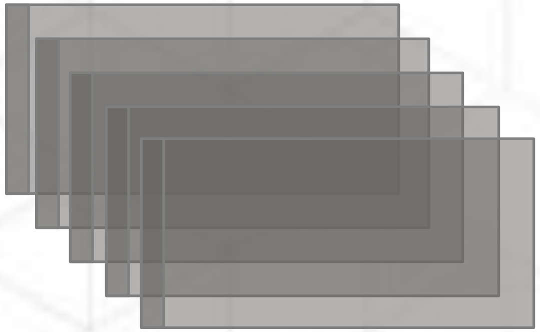
PESIDIOUS - Create Mutated Evasive Malware Using Artificial Intelligence

Bedang Sen
Incident Response Consultant, X-Force IRIS, IBM

Chandni Vaya
Incident Response Consultant, X-Force IRIS, IBM

HITB LOCKDOWN 002 livestream

Contents of this Presentation



Contents of this Presentation

Who are we?

Pesidious

Implementation

Project **Demo**

Future Work

Who are we?

Bedang Sen
Incident Response Consultant
X-Force IRIS, IBM

https://www.linkedin.com/in/bedangsen/

Chandni Vaya
Incident Response Consultant
X-Force IRIS, IBM

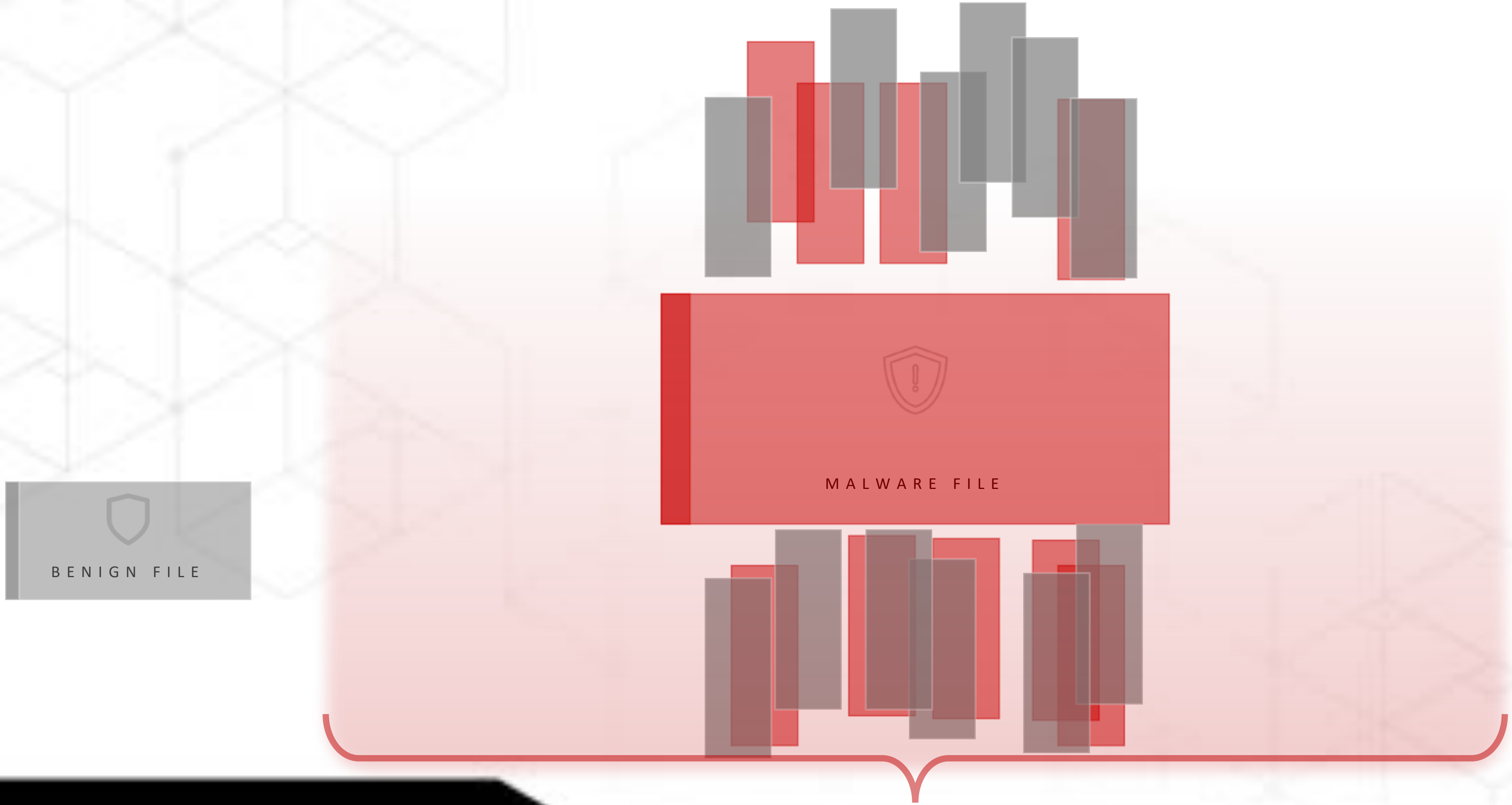
https://www.linkedin.com/in/chandni-vaya-519a05137/

Why Pesidious?

PESIDIOUS AI MUTATION SOLUTION

MALWARE FILE

BENIGN FILE


BENIGN FILE
MALWARE FILE

MALWARE FILE

BENIGN FILE

Benign looking
malware file

What is | Reinforcement Learning

Agent

Environment

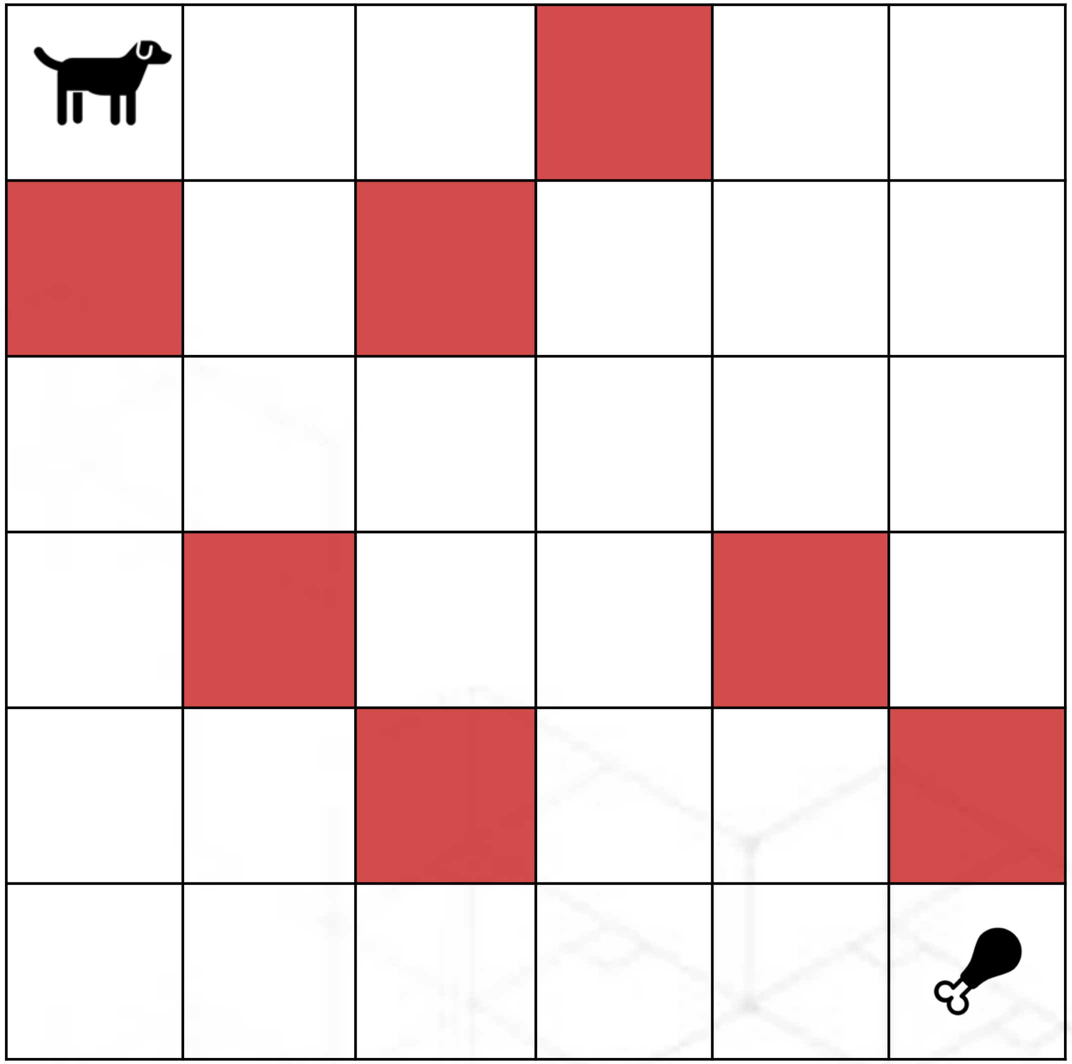
Goal

Q-Value (+ve)

State

Q-Value (-ve)

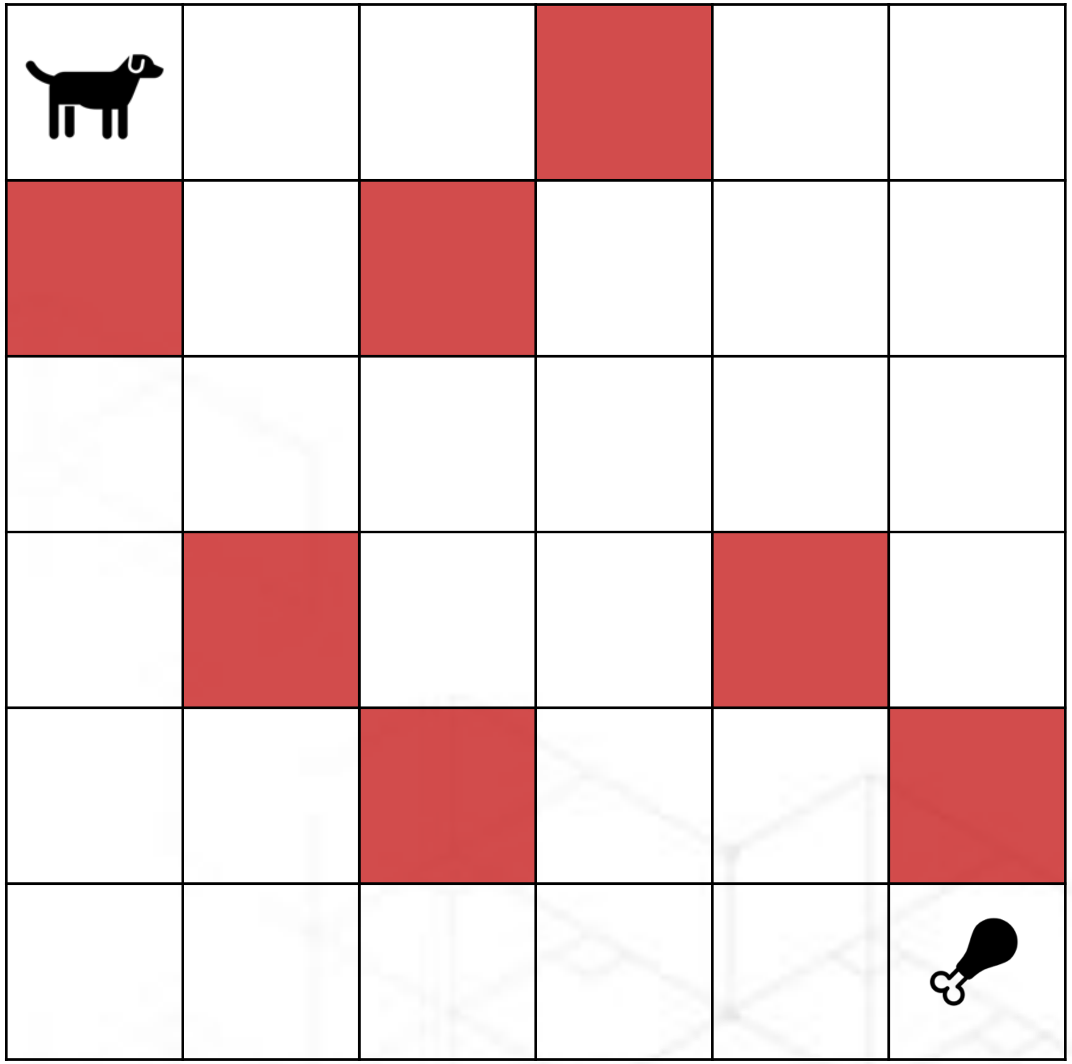
Actions



What is | Reinforcement Learning

	↑	↓	→	←
1	0	-0.51	0.7	0
2	0	0.84	0.5	0.2
4	0	0.87	0.64	0.05
34	0.34	0	0.86	0.21
35	0.55	0	1	0.31

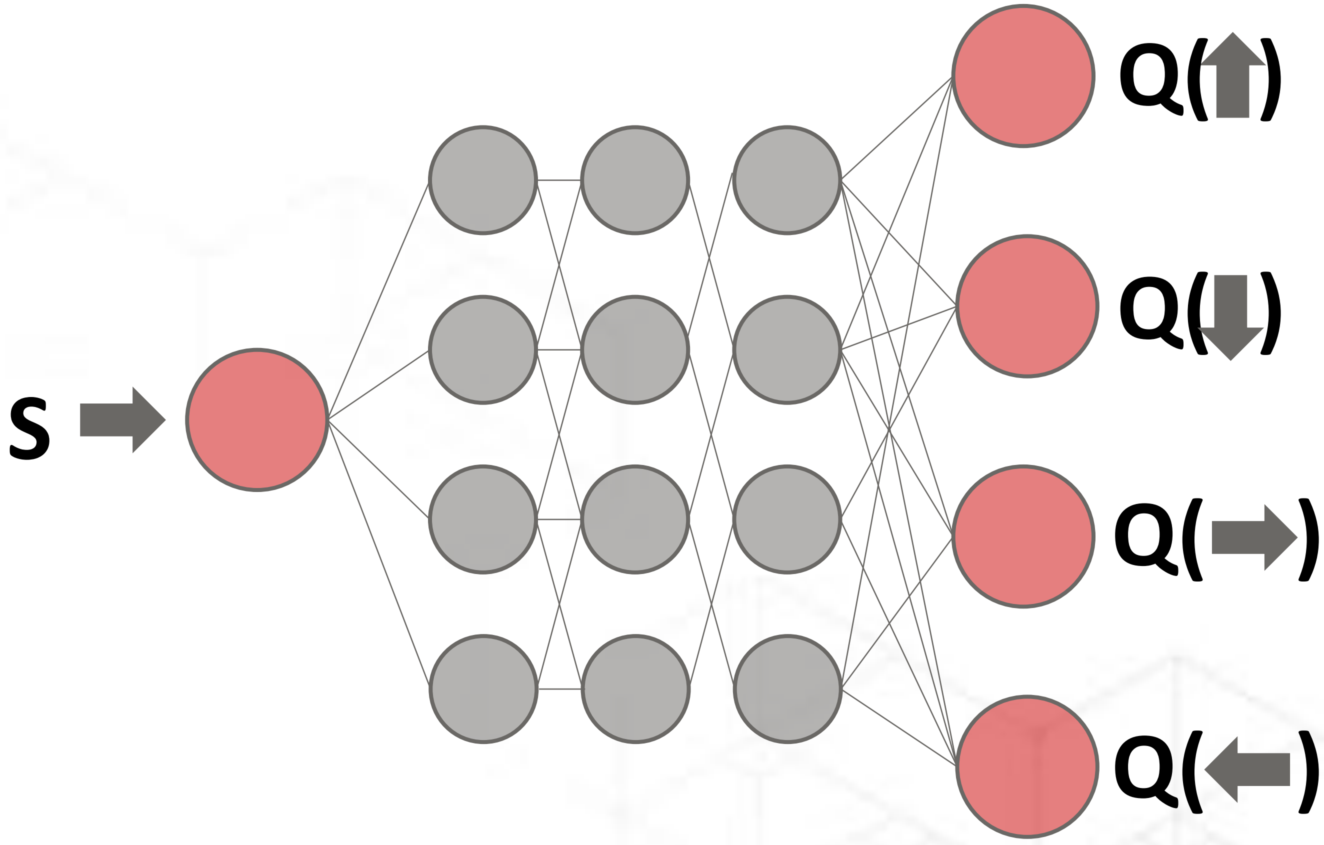
Q-TABLE



What is | Deep Reinforcement Learning

	↑	↓	→	←
1	0	-0.51	0.7	0
2	0	0.84	0.5	0.2
4	0	0.87	0.64	0.05
34	0.34	0	0.86	0.21
35	0.55	0	1	0.31

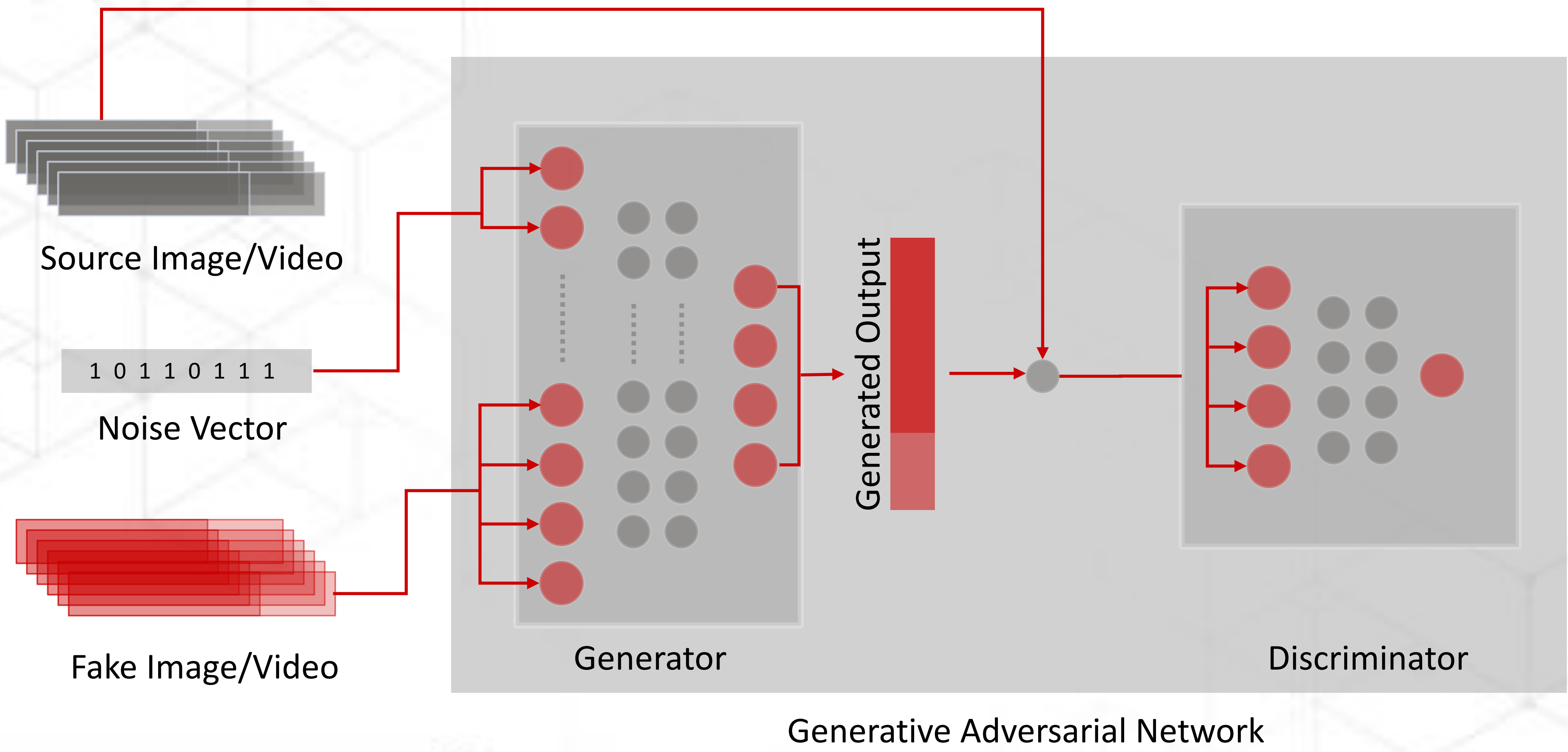
Q-TABLE



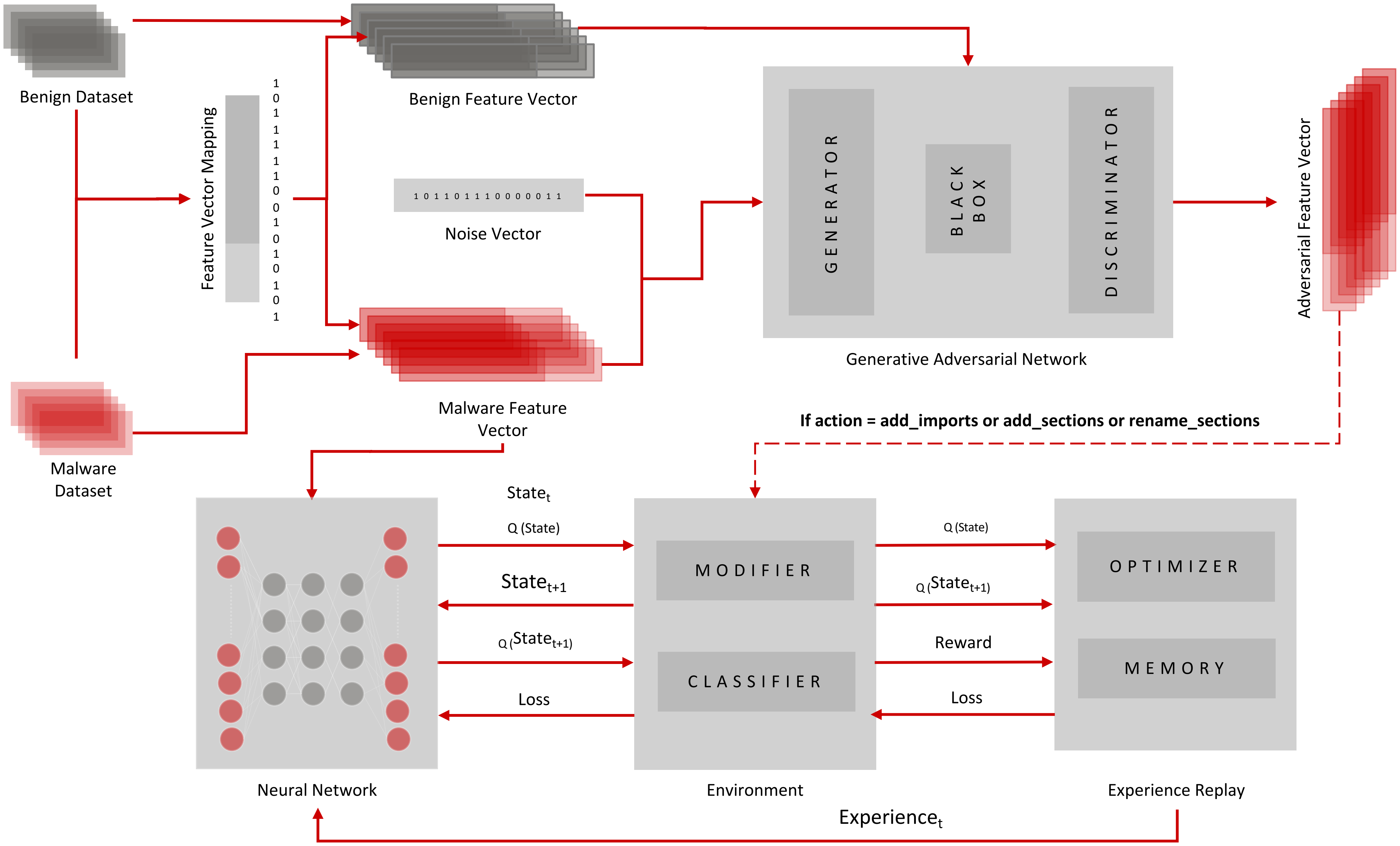
Neural Network

What is | Generative Adversarial Networks

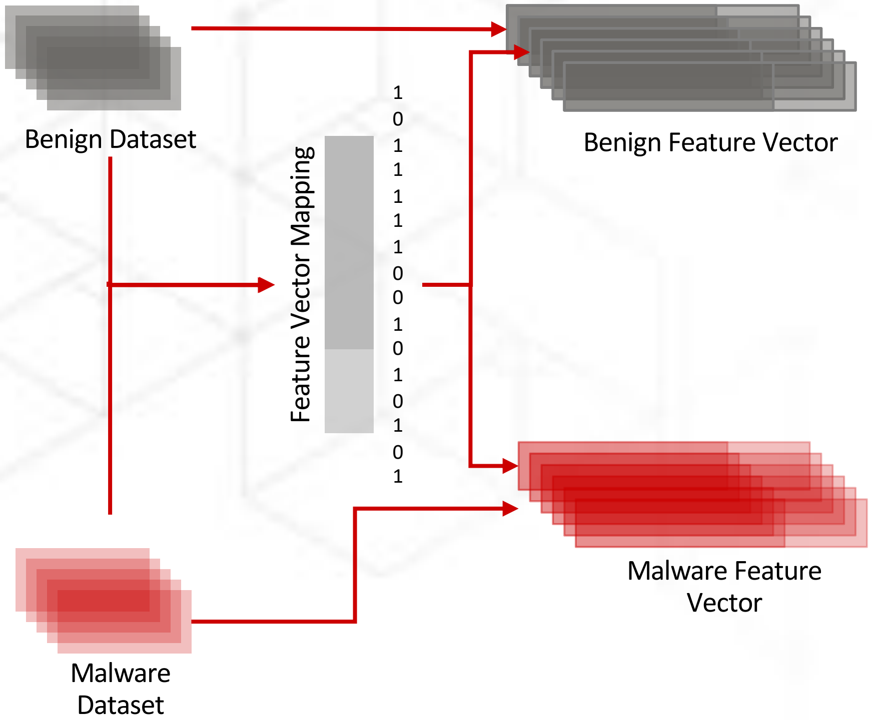
What is | Generative Adversarial Networks



PESIDIOUS | Malware Mutation Using **Reinforcement Learning** and **Generative Adversarial Networks**



Implementation | Extracting Features into Feature Vector Maps



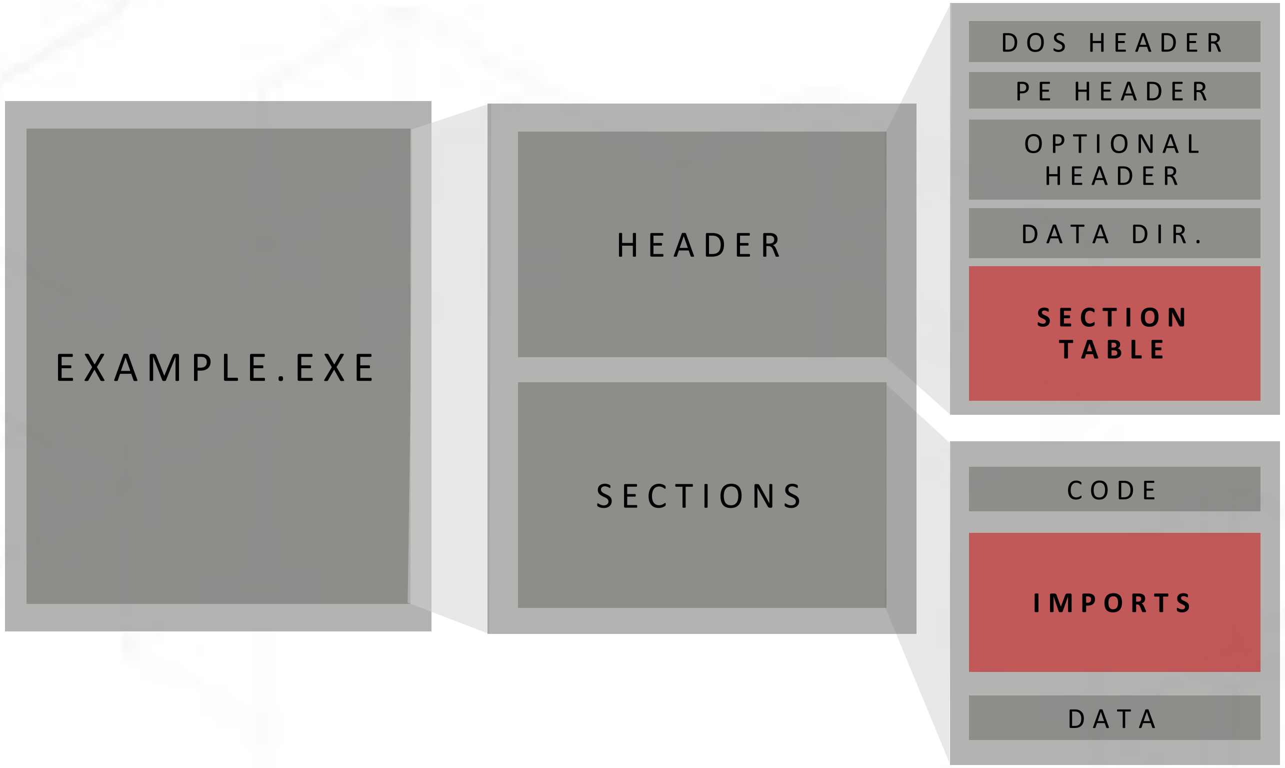
1. Collect the malicious and benign binary dataset.

2. Extract all the features into a single feature vector map

3. Generate feature vectors for each binary data using the feature vector map

1

Implementation | Extracting Features into Feature Vector Maps



1

Implementation | Extracting Features into Feature Vector Maps

BENIGN FILE

- SECTION INFORMATION
  - Number of sections
  - Number of sections with non zero value or empty name
  - No of sections with different characteristics
- PROPERTY OF ENTRY POINTS
- IMPORTS INFORMATION
  - DLL
  - Functions
- EXPORT INFORMATION
- MACHINE ARCHITECTURE OS
- OTHER HEADER INFORMATION
  - Virtual size
  - Debug flag
  - Relocation
  - Resources
  - Signature

Feature Vector Mapping

1
0
1
1
1
1
1
0
0
1
0
1
0
1
0
1

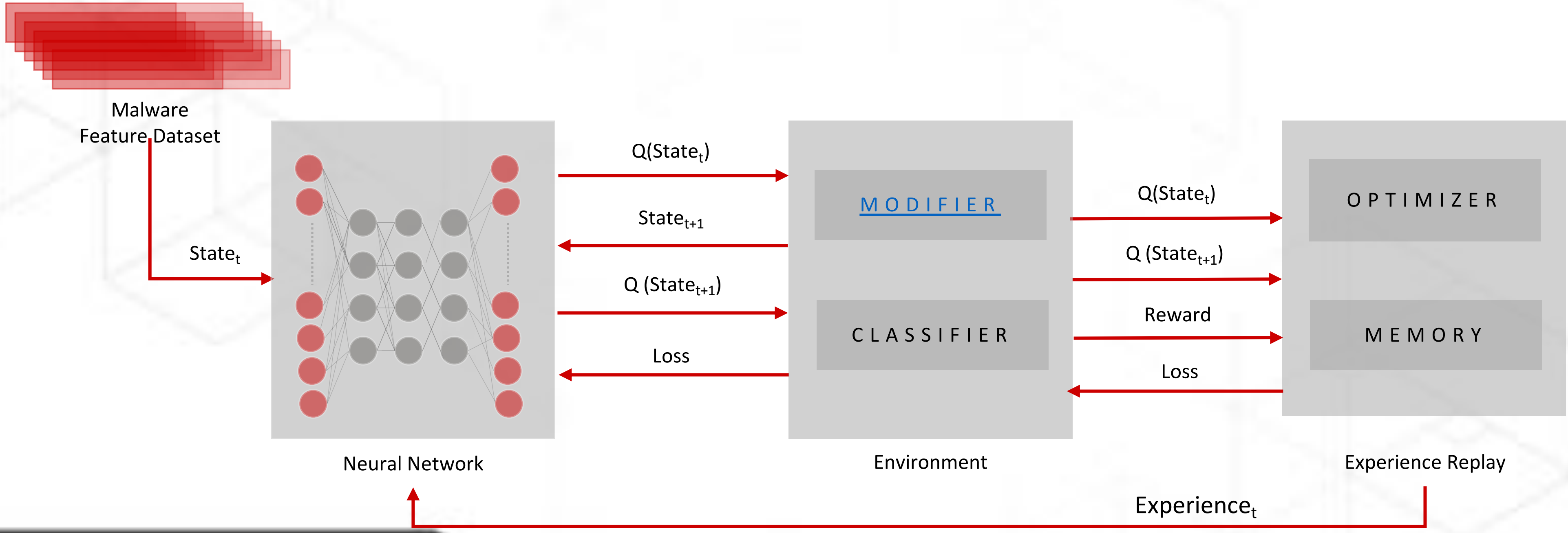
1

Implementation | Training a Deep Reinforcement Learning Agent

1. Implement the environment for the agent to learn.

2. Design a Deep learning model to select the actions based on the current state of the malware.

3. Use experience replay with prioritized replay buffer.



2

Implementation | Training a Deep Reinforcement Learning Agent

MODIFIER

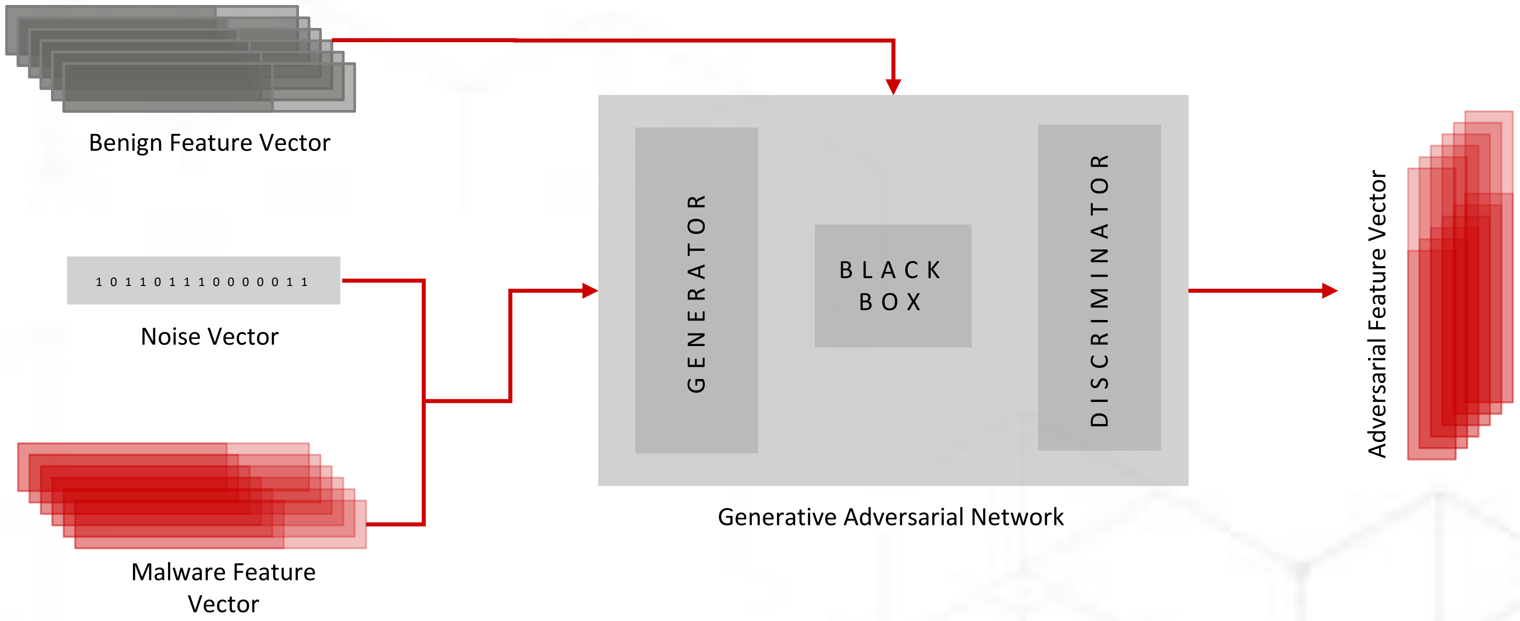
- RANDOMLY ADDING IMPORT FUNCTIONS AND DLLS
- RANDOMLY ADDING SECTIONS AND
- RENAMING SECTIONS
- APPENDING TO EXSITING SECTION
- APPENDING RANDOM BYTES
- REMOVING/ADDING SIGNATURE
- REMOVING DEBUG FLAG
- UPX PACK/UNPACK

2

Implementation | Generating Adversarial Feature Samples with Generative Adversarial Networks

1. Feature vectors are concatenated with noise and fed to the GAN.

2. The GAN generates adversarial feature vectors.



Implementation | Generating Adversarial Feature Samples with Generative Adversarial Networks

DECISION TREE

LOGISTIC REGRESSION

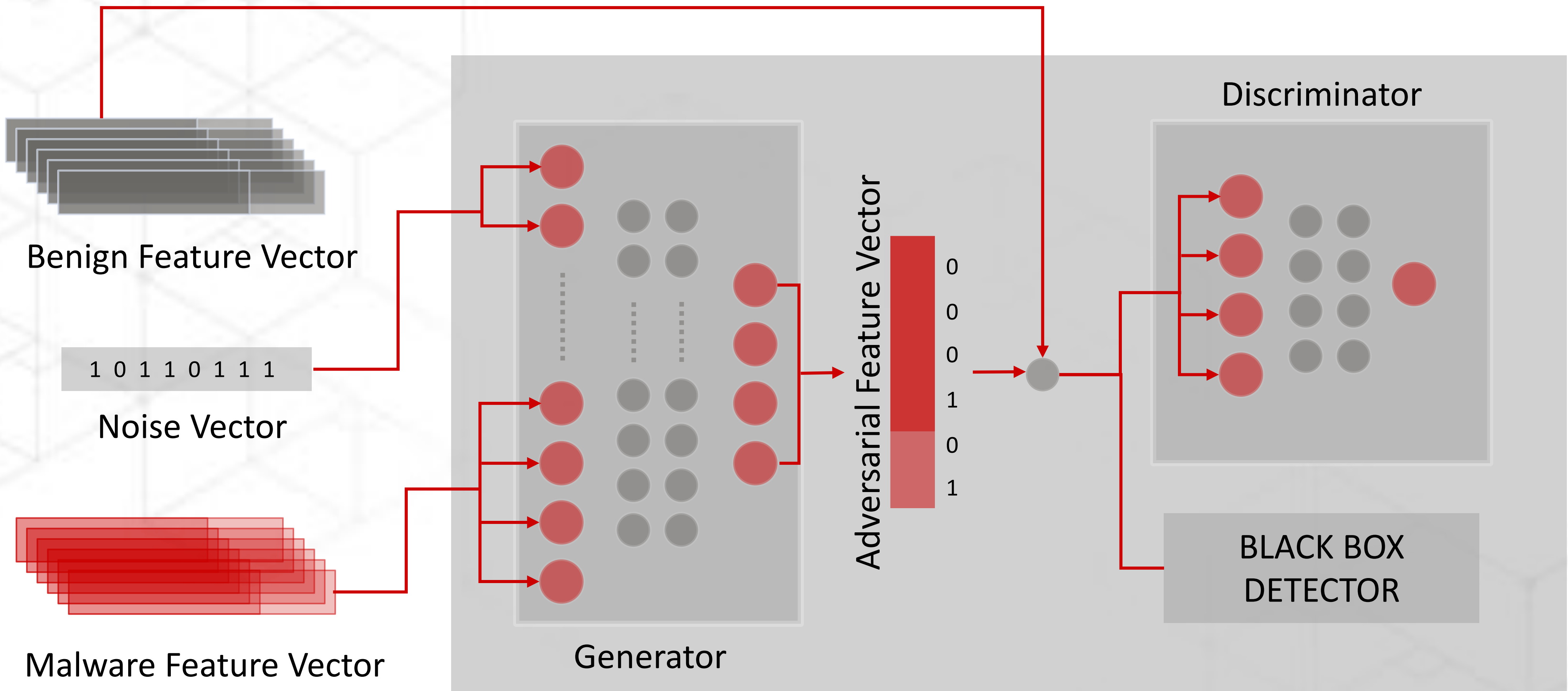
MULTI LAYERPERCEPTRON

RANDOM FOREST

SVM

BLACK BOX

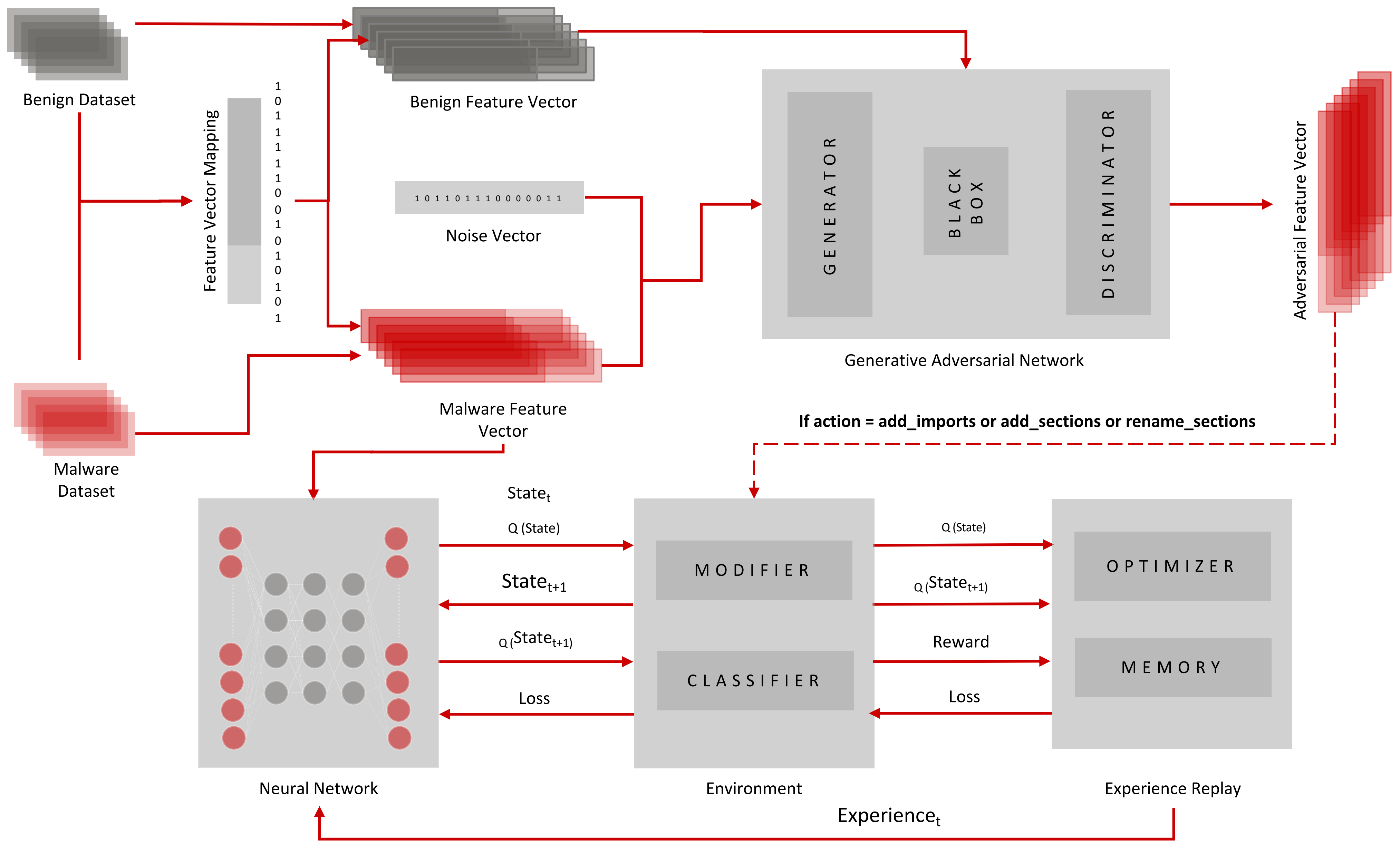
Implementation | Understanding the Generative Adversarial Network



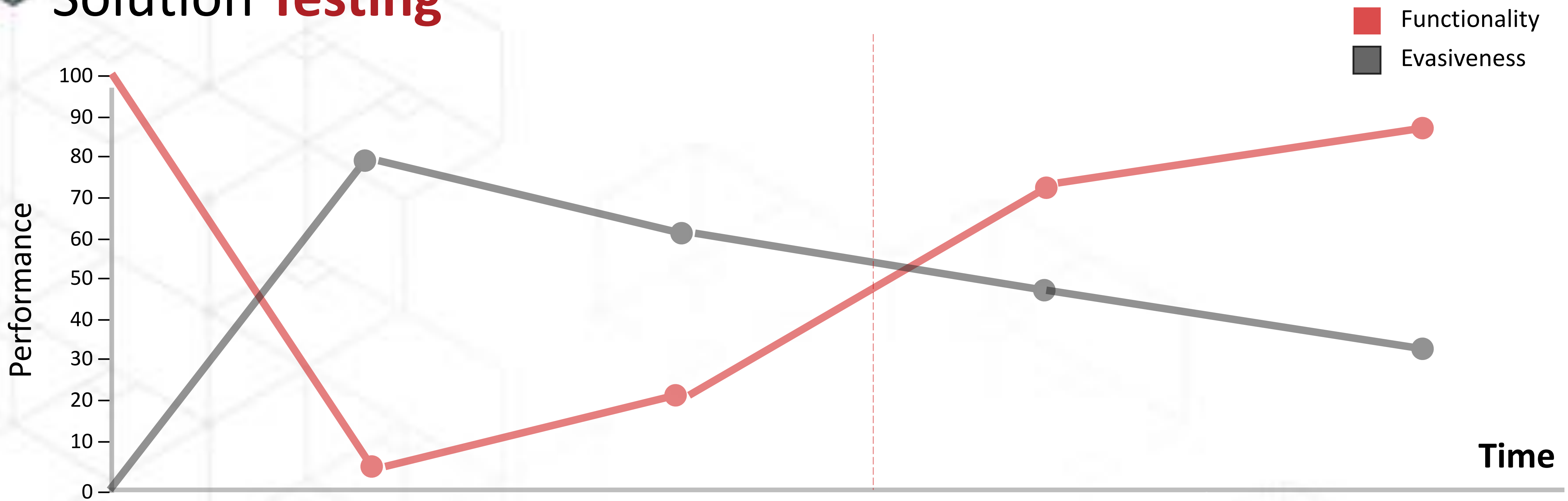
Generative Adversarial Network

3

PESIDIOUS | Malware Mutation Using **Reinforcement Learning** and **Generative Adversarial Networks**



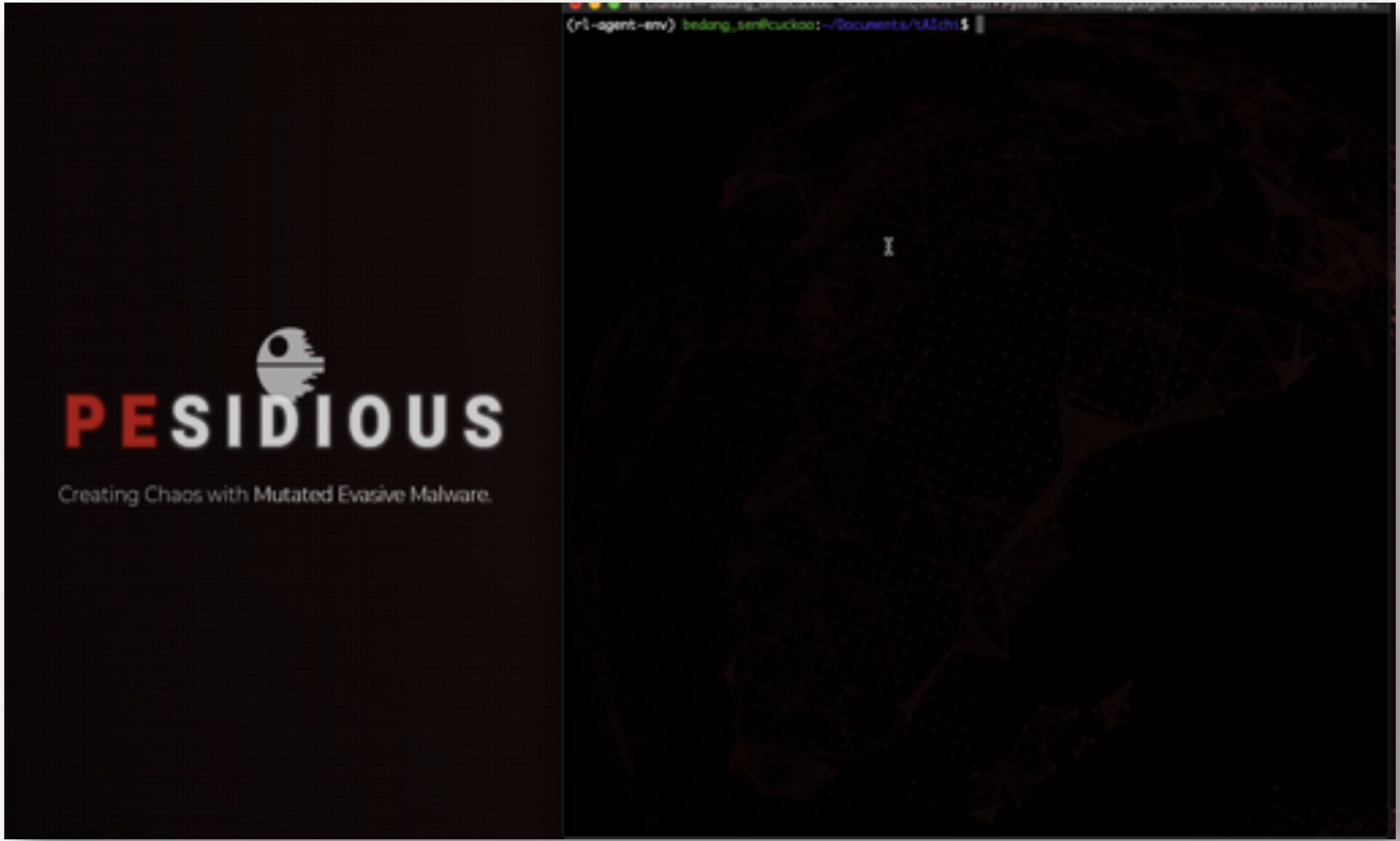
Solution **Testing**



- **Maintaining the functionality:**
    - Filtering out the PE32 files based on 32 bit
    - Filtering out DLLs and sections
    - Using C++ instead of Python for the malware reconstruction

- **Improving performance:**
    - Using a combination of the machine learning models scores
    - Initially we trained it with backdoors; now we are giving it more diverse malwares
    - For testing we made a comparison between AI and human

Project Demo



Project **Demo**

Run Our Mutated Malware In A Cloud Based **Secure Sandboxed Environment.**

Variant.Ransom.Cerber.171:

66 detected

https://bit.ly/2DaxtVz

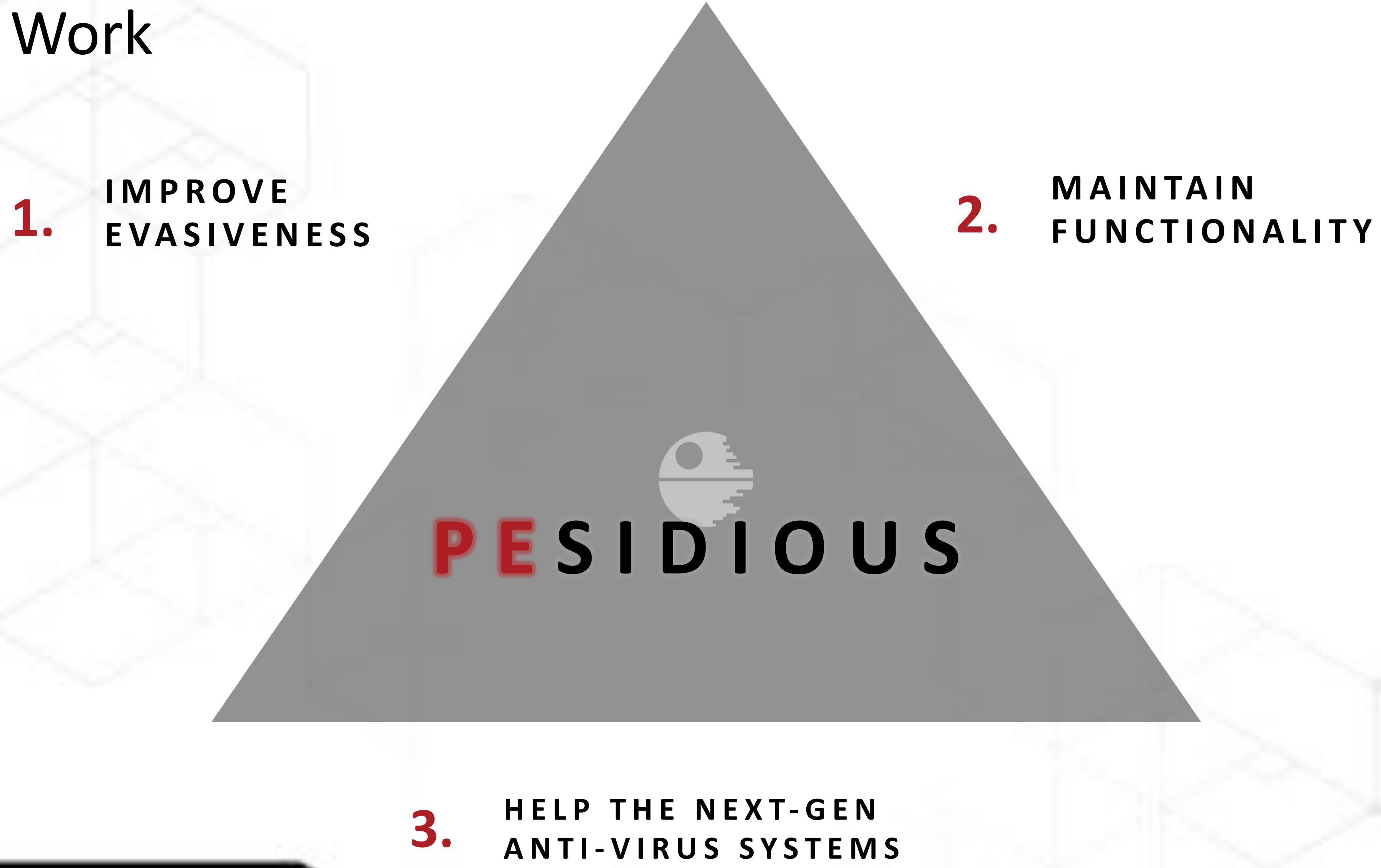
Mutated Variant.Ransom.Cerber.171:

40% more evasive

100% functionality

https://bit.ly/32TFTLU

Future Work


1. IMPROVE EVASIVENESS
2. MAINTAIN FUNCTIONALITY
PESIDIOUS
3. HELP THE NEXT-GEN ANTI-VIRUS SYSTEMS

Thank You!

HITB LOCKDOWN 002 livestream

Bedang Sen, Bedang.Sen@ibm.com & Chandni Vaya, Chandni.Vaya@ibm.com