

Data and Network Security

(Master Degree in Computer Science and Cybersecurity)

Lecture 3

Outline for today

- How malware is spread?
- Emerging trends

How malware is spread?

- Email attachments
- Internet downloads
- File sharing networks
- Removable media
- ...



How malware is spread?



- **Phishing emails:** Attackers send fraudulent emails with malicious links or attachments, and the users, by clicking on links (or opening malicious attachments), unknowingly download/execute malware.

How malware is spread?



- USB and other removable media: malware is already prepared and stored in the removable media (usually autorun scripts)

How malware is spread?



- **Drive-by-downloads:** When you visit a compromised (or malicious) site, malware is downloaded automatically in your system. A large majority of such malware relies on browser vulnerabilities.

How malware is spread?



- **Malvertising:** Malware is embedded in online ads and clicking on those ads the users are redirected to the malicious site and we end up to the drive-by-download scenario

How malware is spread?



- **File sharing networks:** Malware is embedded in pirated software, movies, or cracked applications shared through P2P networks.

How malware is spread?



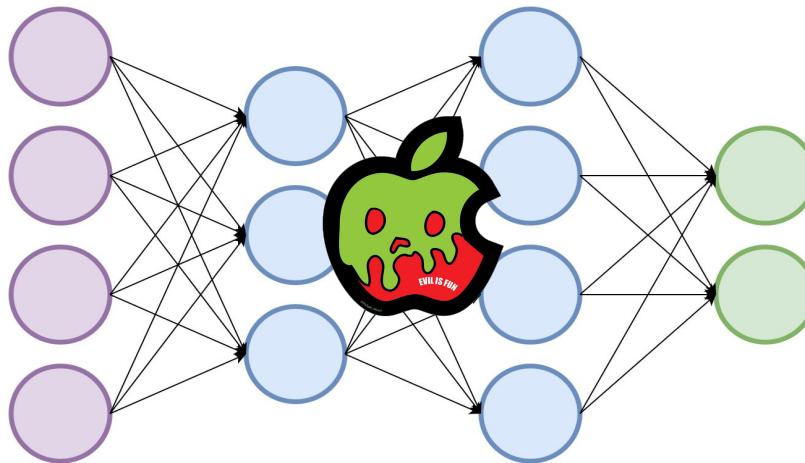
- **Software vulnerabilities:** Malware leverages unpatched software flaws to gain access

How malware is spread again?

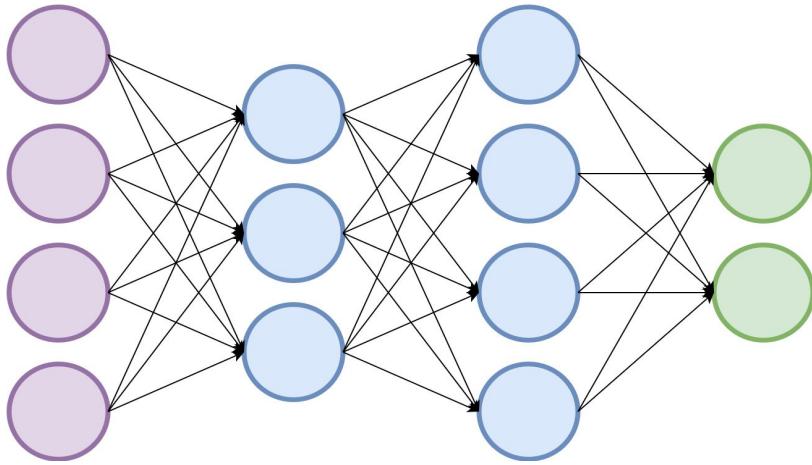


"Look beyond what you see"

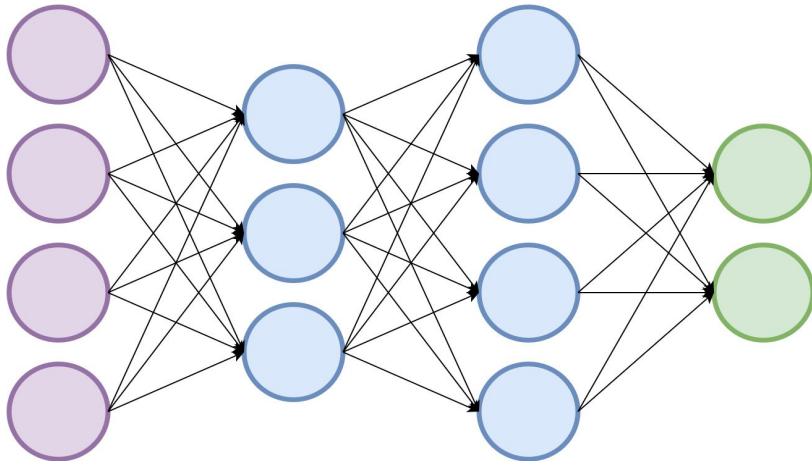
GIST - Malware can also hide inside a deep neural network



Preliminaries - Neural Network



A neural network is a computational model inspired by the structure and functioning of the human brain. It consists of interconnected nodes, or "neurons," organized in layers.



Information flows through the network, undergoing transformations at each layer, to perform tasks such as pattern recognition, classification, and prediction.

AI is rapidly becoming instrumental in an ever-growing number of areas

MIT
Technology
Review

SIGN IN

SUBSCRIBE



BIOTECHNOLOGY AND HEALTH

DeepMind is using AI to pinpoint the causes of genetic disease

Fresh from solving the protein structure challenge, Google's deep-learning outfit is moving on to the human genome.

By Antonio Regalado

September 19, 2023



GOOGLE DEEPMIND

<https://www.technologyreview.com/2023/09/19/1079871/deepmind-alphamissense-ai-pinpoint-causes-genetic-disease/>

<https://cloud.google.com/blog/products/gaming/generative-ai-in-the-games-industry>

<https://www.mckinsey.com/capabilities/operations/our-insights/generative-ai-in-finance-finding-the-way-to-faster-deeper-insights#/>

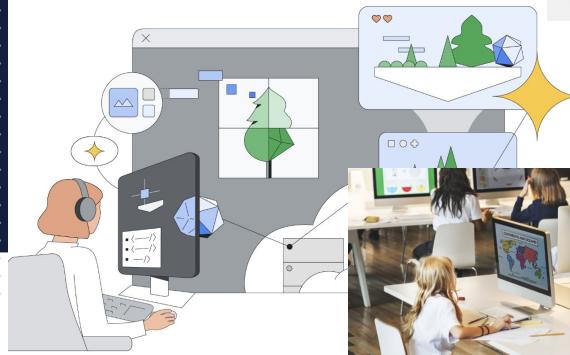
<https://www.morganstanley.com/ideas/generative-ai-education-outlook>

McKinsey
& Company
Operations



The generative AI revolution in the games industry: A path to boundless creativity

October 6, 2023



RESEARCH

Generative AI Is Set to Shake Up Education

Dec 22, 2023

While educators debate the risks and opportunities of generative AI as a learning tool, some education technology companies are using it to increase revenue and lower costs.

5 2 5 0 0 0 5 8
 4 9 8 7 5 1 5 3
 4 0 0 2 3 4 3 4
 3 6 2 2 9 2 3 7
 1 2 2 4 5 3 8 6
 2 1 4 1 6 4 0 9
 5 8 8 0 5 6 6 3
 5 4 5 7 9 9 7 1

MNIST images



CIFAR-10 images



faces



album covers

In less than a decade, we went from this...

images from:

- <https://blog.openai.com/generative-models/>

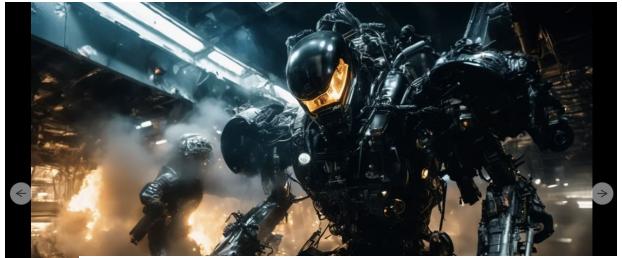
- Goodfellow et al. Generative Adversarial Networks

- Radford et al. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks



bedrooms

To this...



2:23 🔍 GPT-3.5 🔍 GPT-4

An image generated by GPT-3.5 and GPT-4 showing a large, metallic, multi-limbed robot standing in a dark, industrial environment with fire in the background.

Who was Robert Oppenheimer | ↗

LANCE WHITNEY Who was Robert Oppenheimer

CHATGPT Robert Oppenheimer was an American physicist and one of the key figures in the development of the atomic bomb during World War II. He led the Los Alamos Laboratory, which was responsible for the Manhattan Project, a top-secret government research project aimed at creating the first nuclear weapons. Oppenheimer's leadership and scientific contributions were instrumental in the successful creation of the atomic bomb. However, after witnessing the devastating power of the bomb during the tests in New Mexico in 1945, he expressed mixed feelings about its use. He later became an advocate for the peaceful use of atomic energy and played a significant role in shaping science and education policies in the United States.

q w e r t y u i o p
a s d f g h j k l
z x c v b n m ↵
123 ↵

Message ↴ ↵



Prompt: Photorealistic closeup video of two pirate ships battling each other as they sail inside a cup of coffee.

0:01 / 0:15 ↴ ↵

<https://openai.com/sora>

<https://arstechnica.com/information-technology/2023/07/stable-diffusion-xl-puts-ai-generated-visual-worlds-at-your-gpus-command/>

<https://www.zdnet.com/article/official-chatgpt-app-for-android-finally-launches/>



The cost of Building a high-performance DNN



Huge computational power

Large quantities of data





The cost of Building a high-performance DNN

Large quantities of data



Huge computational power

Users prefer to consume the Pre-Trained Model



Hugging Face

kaggle

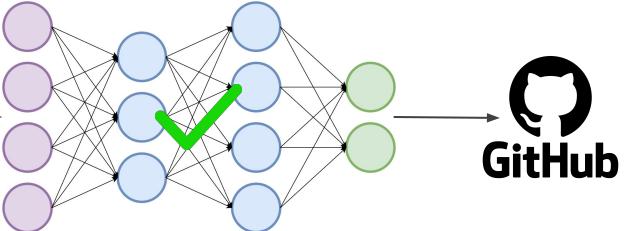
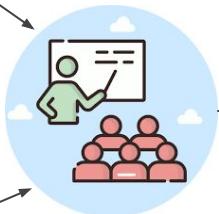
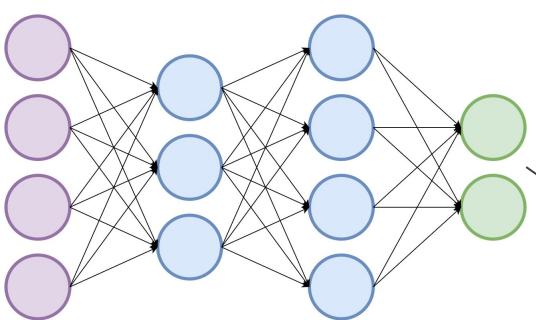


Model Zoo

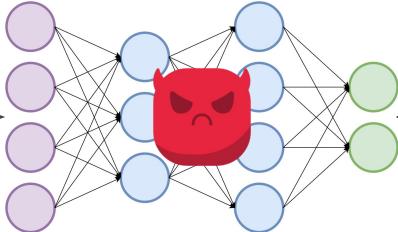
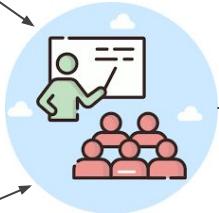
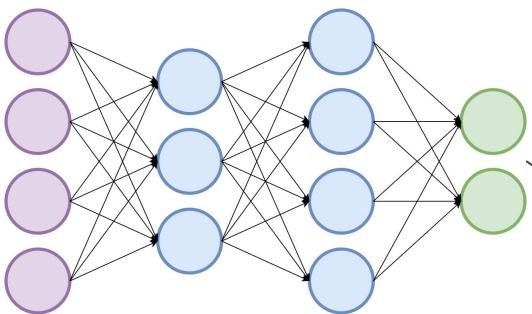
GitHub

The GitHub logo features a black octocat icon inside a circle, followed by the word "GitHub" in a bold, black, sans-serif font.

Do you trust your model?



Do you trust your model?



To run these DNNs...

- You need specific software/libraries
 - TensorFlow
 - Pytorch
 - ...

To run these DNNs...

- You need specific software/libraries
 - TensorFlow
 - Pytorch
 - ...

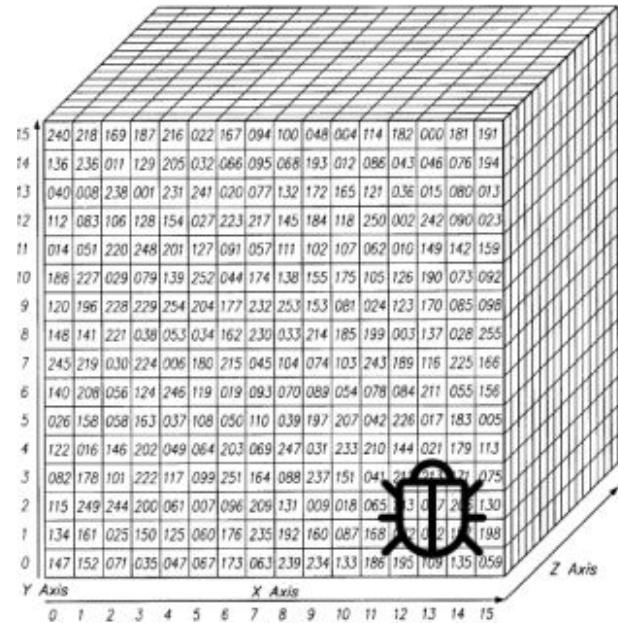
TensorFlow saves variables in binary *checkpoint files* that map variable names to tensor values.



Caution: TensorFlow model files are code. Be careful with untrusted code. See [Using TensorFlow Securely](#) for details.

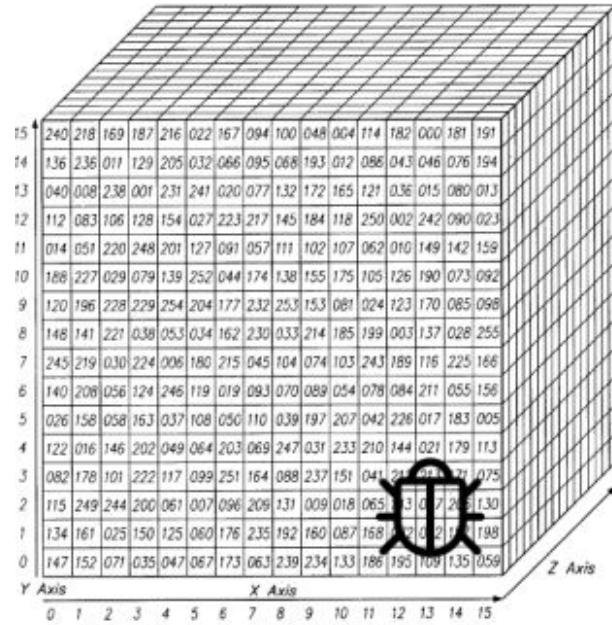
How are these DNNs structured again?

- In the end of days, DNN weights are essentially a matrix of floating point numbers.

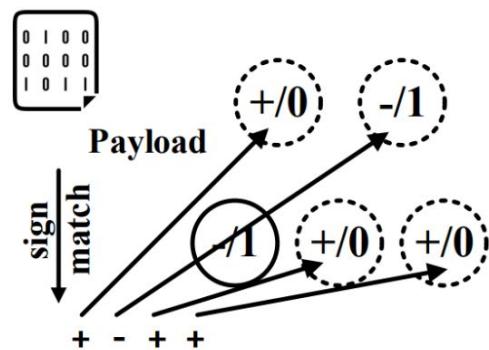
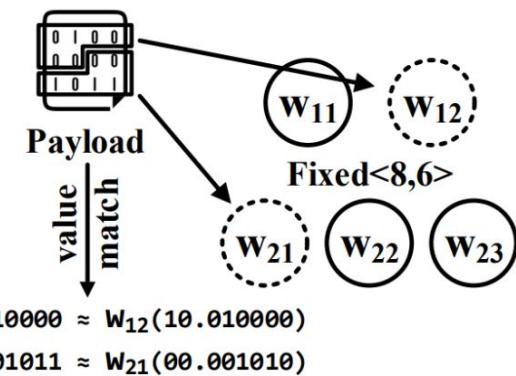
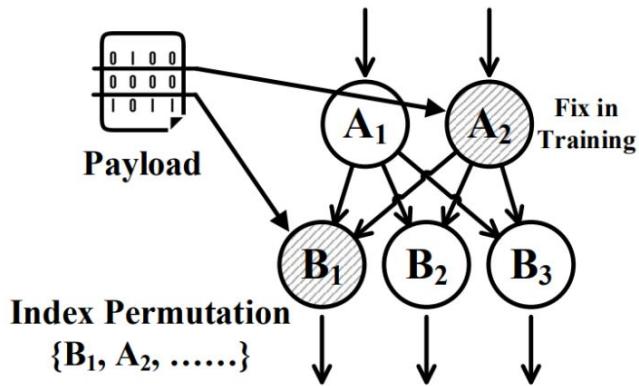
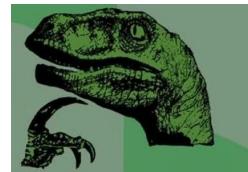


How are these DNNs structured again?

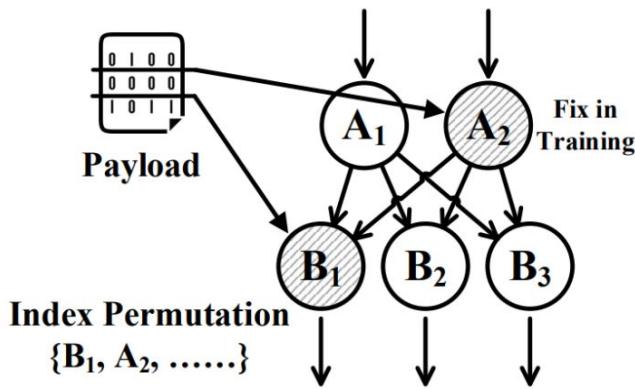
- In the end of days, DNN weights are essentially a matrix of floating point numbers.
- These values can be altered by also making sure the performance of the DNN on its intended task remains the same.



How to embed a malicious payload into a DNN?



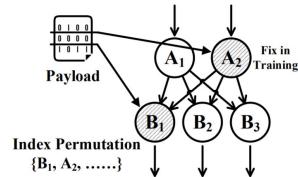
Resilience Training



Resilience Training: Fix some neurons (parameter) and train the network to perform good on the intended legitimate task without altering those nodes.

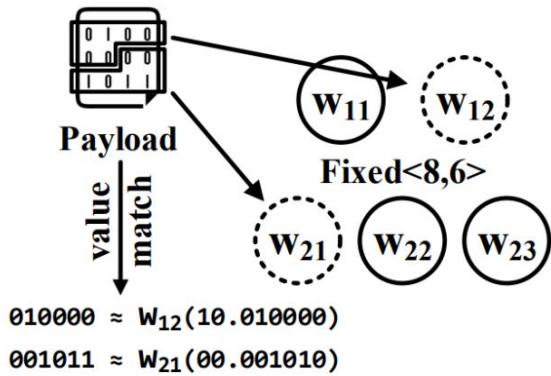
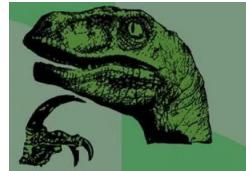
- Malware bits are spread in the fixed parameters
- Need to collect in the appropriate order to have the correct malware binary
- Detectable?
- Robust?

Resilience Training (Model impact)



	Original	EquationDrug	ZeusVM-decrypted	Cerber	Ardamax	NSIS	Kelihos	Mamba	WannaCry	
Medium DNNs	Resnet50	75.2%	75.2%	75.4%	74.8%	74.7%	75.1%	75.3%	74.6%	74.8%
	Inceptionv3	78%	78.3%	78.4%	78.2%	77.9%	78.4%	78.1%	77.6%	78.4%
	Densnet201	77%	77.2%	76.7%	77.1%	76.9%	77.3%	76.5%	77.2%	77.3%
	Resnet18	70.7%	71.1%	71.2%	70.9%	71%	70.4%	70.3%	70.9%	71.3%
Small DNNs	Googlenet	69.8%	70.3%	69.2%	69.6%	71%	70.5%	69.3%	70.4%	70.2%
	Comp.VGG-16	70.5%	68.3%	69.1%	71.2%	69.1%	68.4%	63.4%	56.1%	22.6%
	Comp.Alexnet	57%	55.4%	56.7%	57.2%	54.1%	38.2%	34.3%	16.7%	3.9%
	Squeezeenet	57.5%	56.8%	54.3%	53.2%	48.3%	35.4%	29.6%	15.1%	4.1%
	Mobilenet	70.9%	71.2%	68.5%	66.7%	54.4%	32.5%	29.1%	6.1%	0.7%

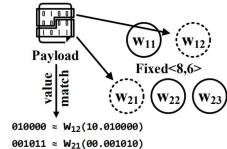
Value Mapping



Value mapping: Train the network to perform good on the intended legitimate task without altering anything:

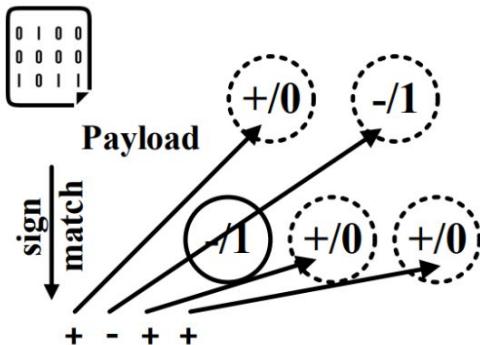
- Malware bits are mapped in some parameters of the DNN
- Need to collect in the appropriate order to have the correct malware binary.
- Detectable?
- Robust?

Value Mapping (Model impact)



	Original	EquationDrug	ZeusVM-decrypted	Cerber	Ardamax	NSIS	Kelihos	Mamba	WannaCry
Medium DNNs	Resnet50	75.2%	74.8%	74.7%	75.1%	75.3%	74.6%	74.8%	74.6%
	Inceptionv3	78%	78.3%	78.4%	78.2%	78%	77.2%	78.3%	78.4%
	Densnet201	77%	77.1%	76.9%	77.3%	76.5%	77.2%	76.5%	77.3%
	Resnet18	70.7%	71.1%	70.2%	70.5%	71.6%	72.1%	70.9%	71%
Small DNNs	Googlenet	69.8%	70.1%	68.3%	70.2%	68.1%	70.3%	69.8%	68.4%
	Comp.VGG-16	70.5%	71.3%	71%	68.3%	69.1%	71.2%	69.1%	69.2%
	Comp.Alexnet	57%	56.9%	56.7%	57.2%	54.1%	-	-	-
	Squeezezenet	57.5%	57.3%	56.9%	55.7%	56.8%	39.7%	43.2%	21.8%
	Mobilenet	70.9%	69.2%	71%	70.5%	70.3%	54.7%	48.6%	49.3%

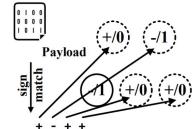
Sign Mapping



Sign mapping: Train the network to perform good on the intended legitimate task without altering anything:

- Malware bits are mapped in some parameters of the DNN (following the sign of the parameter)
- Need to collect in the appropriate order to have the correct malware binary.
- Detectable?
- Robust?

Sign Mapping (Model Impact)



	Original	EquationDrug	ZeusVM-decrypted	Cerber	Ardamax	NSIS	Kelihos	Mamba	WannaCry
Large DNNs	Vgg19 (17.16MB)	71.1%	71.3%	71.2%	70.7%	71.2%	70.9%	71.1%	71.2%
	Vgg16 (16.45MB)	70.5%	71%	70.6%	69.9%	70.2%	71.1%	70.8%	70.2%
	Alexnet (7.27MB)	57%	57.2%	57.1%	56.7%	57.3%	57.2%	56.8%	56.9%
	Resnet101 (5.32MB)	77.1%	77.2%	77.4%	76.7%	76.3%	77%	77.3%	77.5%
Medium DNNs	Resnet50 (3.05MB)	75.2%	74.8%	75.3%	75.1%	75.5%	75.4%	75.3%	74.9%
	Inceptionv3 (2.84MB)	78%	77.4%	78.2%	78.1%	77.8%	78%	78.1%	-
	Densnet201 (2.38MB)	77%	77.3%	77.2%	76.4%	76.7%	77.1%	76.4%	-
	Resnet18 (1.39MB)	70.7%	71.1%	70.8%	71%	71.2%	69.5%	-	-
Small DNNs	Googlenet (0.83MB)	69.8%	68.3%	70.2%	70.1%	-	-	-	-
	Comp.VGG-16 (1.34MB)	70.5%	71.3%	71%	70.4%	69.6%	-	-	-
	Comp.Alexnet (0.83MB)	57%	55.4%	56.7%	56.3%	-	-	-	-
	Squeezezenet (0.15MB)	57.5%	-	-	-	-	-	-	-
	Mobilenet (0.5MB)	70.9%	68.3%	-	-	-	-	-	-

Detectable/Robust?

Baselines	Selected Malware Samples					
	Asprox	Bladabindi	Destover	Kovter	Stuxnet	ZeusVM
Vanilla-malware	72.97%	75.68%	83.78%	62.16%	89.19%	91.89%
Stegomalware	8.11%	10.81%	13.51%	5.41%	0.00%	8.11%
*LSB substitution	0.00%	2.70%	2.70%	0.00%	0.00%	0.00%
*Sign-mapping	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Detectable/Robust?

Baselines	Selected Malware Samples					
	Asprox	Bladabindi	Destover	Kovter	Stuxnet	ZeusVM
Vanilla-malware	72.97%	75.68%	83.78%	62.16%	89.19%	91.89%
Stegomalware	8.11%	10.81%	13.51%	5.41%	0.00%	8.11%
*LSB substitution	0.00%	2.70%	2.70%	0.00%	0.00%	0.00%
*Sign-mapping	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%



How malware is spread again?



"Look beyond what you see"

Preliminaries - Digital (stealthy) communication

Digital Communication

- Consists of the transfer of data or information using digital signals over a point-to-point channel.
- Data or information are digitally encoded as discrete signals, and are transferred through a communication channel.



What is CDMA?



- Channel coding technique by which a narrowband signal is spread into a wider bandwidth.
- Codes the data at a higher frequency by using pseudorandomly generated codes that the receiver knows.
- developed in the 1950s for military communications:
 - resist the enemy efforts to jam the communication channel.
 - hide the fact that the communication is taking place.



Communication channel

CDMA 101



Binary sequence: [0, 1, 1]

Assuming $\omega = 0$:



PSK

[-1, 1, 1]

Using Phase-Shift Keying:

- 0 → $-\cos(\omega t)$

- 1 → $\cos(\omega t)$

- ω is the transmitting frequency

- $0 \leq t < T$

CDMA 101



Binary sequence: [0, 1, 1]



PSK [-1, 1, 1]

Spreading code: [-1, 1, -1, -1, 1]

CDMA 101



Binary sequence: [0, 1, 1]



PSK [-1, 1, 1]

Spreading code: [-1, 1, -1, -1, 1]

Chip sequence: [1, -1, 1, 1, -1, -1, 1, -1, -1, 1, -1, 1, -1, -1, -1, 1]

CDMA 101



Binary sequence: [0, 1, 1]



PSK [-1, 1, 1]

Spreading code: [-1, 1, -1, -1, 1]

Chip sequence: [1, -1, 1, 1, -1, -1, 1, -1, -1, 1, -1, 1, -1, -1, -1, 1]

-5

+5

+5

CDMA example communication - 2 users

Binary sequence: [0, 1]

PSK

[-1, 1]

Spreading code: [-1, -1, 1, 1]

Chip sequence: [1, 1, -1, -1, -1, -1, -1, 1, 1]

Binary sequence: [0, 0]

[-1, -1]

[1, -1, 1, -1]

[-1, 1, -1, 1, -1, 1, -1, 1]

CDMA example in communication

User #1 [1, 1, -1, -1, -1, -1, 1, 1]

User #2 [-1, 1, -1, 1, -1, 1, -1, 1]

Combined signal: [0, 2, -2, 0, -2, 0, 0, 2] - two users worth of data



CDMA example - Decode user #1

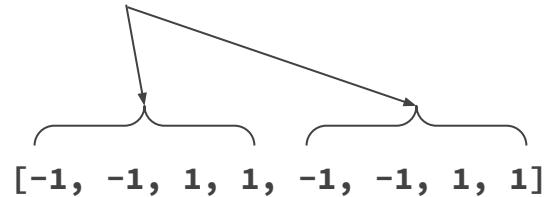
Combined signal: [0, 2, -2, 0, -2, 0, 0, 2] - two users worth of data

Spreading code: [-1, -1, 1, 1]

CDMA example - Decode user #1

Combined signal: [0, 2, -2, 0, -2, 0, 0, 2] - two users worth of data

Spreading code: [-1, -1, 1, 1]



CDMA example - Decode user #1

Combined signal: [0, 2, -2, 0, -2, 0, 0, 2] - two users worth of data



[-1, -1, 1, 1, -1, -1, 1, 1]

[0, -2, -2, 0, 2, 0, 0, 2]



Add up

Add up

CDMA example - Decode user #1

Combined signal: [0, 2, -2, 0, -2, 0, 0, 0, 2] - two users worth of data



[-1, -1, 1, 1, -1, -1, 1, 1]

[0, -2, -2, 0, 2, 0, 0, 2]

-4

+4

CDMA example - Decode user #1

Combined signal: [0, 2, -2, 0, -2, 0, 0, 0, 2] - two users worth of data



[-1, -1, 1, 1, -1, -1, 1, 1]

[0, -2, -2, 0, 2, 0, 0, 2]

-4

+4

What was the length of the spreading code?

CDMA example - Decode user #1

Combined signal: [0, 2, -2, 0, -2, 0, 0, 0, 2] - two users worth of data



[-1, -1, 1, 1, -1, -1, 1, 1]

[0, -2, -2, 0, 2, 0, 0, 2]

————— —————

We have to divide by 4

-4

+4

-1 +1

CDMA example - Decode user #1

Combined signal: [0, 2, -2, 0, -2, 0, 0, 0, 2] - two users worth of data



[-1, -1, 1, 1, -1, -1, 1, 1]

[0, -2, -2, 0, 2, 0, 0, 2]

————— —————

-4

+4

Remember PSK mapping

-1 was 0

and

+1 was 1

-1 +1

CDMA example - Decode user #1

Combined signal: [0, 2, -2, 0, -2, 0, 0, 2] - two users worth of data



[-1, -1, 1, 1, -1, -1, 1, 1]

[0, -2, -2, 0, 2, 0, 0, 2]

User #1 message was **01**

-4

+4

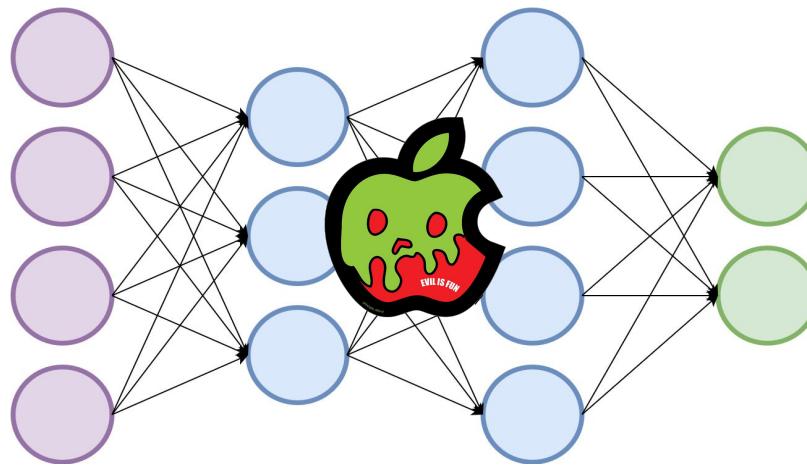
0

1

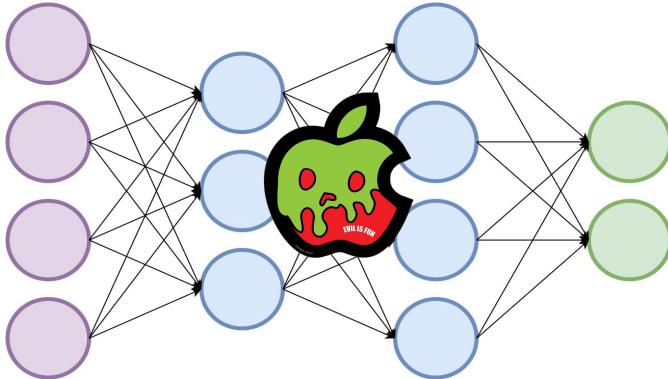
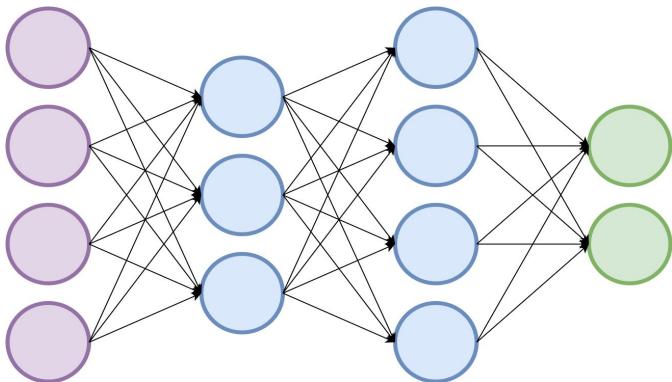
Why CDMA is stealthy?



- Spreading codes are pseudorandomly generated and are in the orders of **tens of thousands**



Malware embedding into DNNs: HowTo



Malware embedding into DNNs: HowTo

Malicious Payload \rightarrow P bits $b = [b_0, \dots, b_{P-1}]$

[1, 0, 1, 0]

Malware embedding into DNNs: HowTo (cont.)

Malicious Payload \rightarrow P bits $\mathbf{b} = [b_0, \dots, b_{P-1}]$

- Each bit is encoded as ± 1 .

$$\begin{matrix} [1, 0, 1, 0] \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ [+1, -1, +1, -1] \end{matrix}$$



Malware embedding into DNNs: HowTo (cont.)

Malicious Payload \rightarrow P bits $\mathbf{b} = [\mathbf{b}_0, \dots, \mathbf{b}_{P-1}]$

- The spreading code (c_i) of each bit is a vector of ± 1 that is of the same length of vector W , namely R .

$$[+1, -1, +1, -1]$$



$$[+1, +1, -1, +1, \dots, -1] \quad c_i$$

R elements

Malware embedding into DNNs: HowTo (cont.)

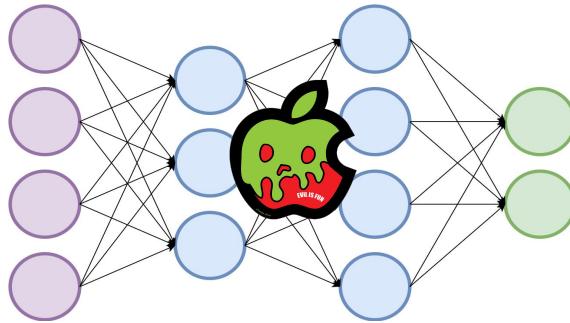
Malicious Payload \rightarrow P bits $\mathbf{b} = [b_0, \dots, b_{P-1}]$

- \mathbf{C} is an R by P matrix that collects all the codes.

$$\begin{matrix} & +1 & +1 & -1 & \\ & -1 & -1 & -1 & \\ & +1 & -1 & +1 & \\ \mathbf{R} & \cdot & \cdot & \cdot & \\ & \cdot & \cdot & \cdot & \cdots \cdots \cdot \\ & \cdot & \cdot & \cdot & \\ & -1 & -1 & -1 & +1 \end{matrix}$$

\mathbf{P}

Malware embedding into DNNs: HowTo (cont.)



$$w^{\text{Malicious}} = w + \gamma c_b$$

γ – gain factor to control
the power of the signal

Payload Extraction

Each bit b_i of the malicious payload $\mathbf{b} = [b_0, \dots, b_{p-1}]$ can be recovered by performing:

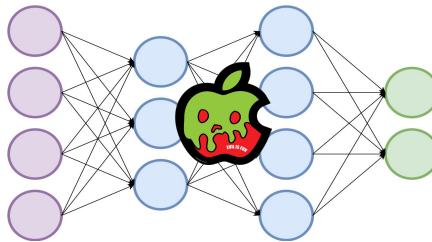
$$\hat{b}_i = c_i^T W^{\text{Malicious}}$$



$$b_i = \text{sign}(\hat{b}_i)$$

How does it perform?

Stealthiness



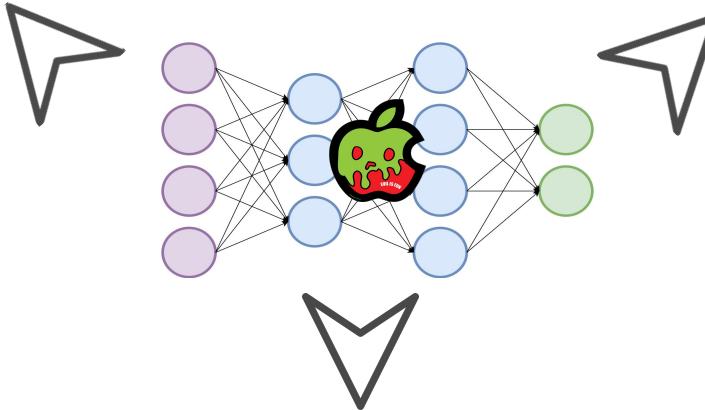
**Impact on
model
performance**



Robustness

Stealthiness

Stealthiness



**Impact on
model
performance**

Robustness

Stealthiness



1. Evaluation against anti-malware software.
2. Statistical analysis



Stealthiness



1. **Evaluation against anti-malware software.**
2. Statistical analysis

	Malware samples				
	Stuxnet	Destover	Bladabindi	Zeus	Kovter
Plain malware	89.19%	83.78%	72.97%	91.89%	62.16%
Stegomalware	0.00%	13.51%	8.11%	10.81%	5.41%
Malicious-DNN	0.00%	0.00%	0.00%	0.00%	0.00%

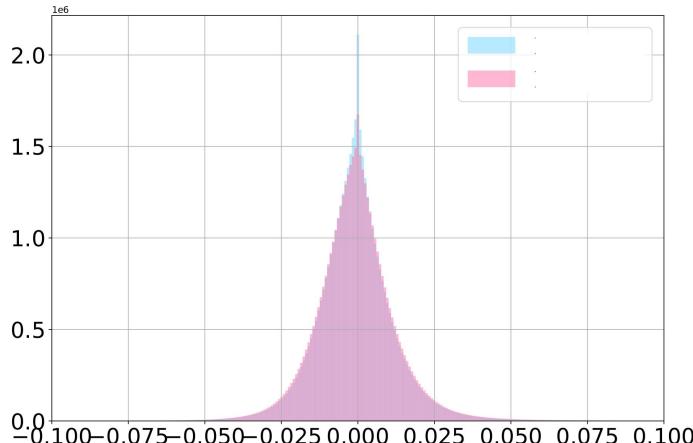
Detection rate on **Metadefender** metascan feature

Stealthiness



1. Evaluation against anti-malware software.

2. Statistical analysis

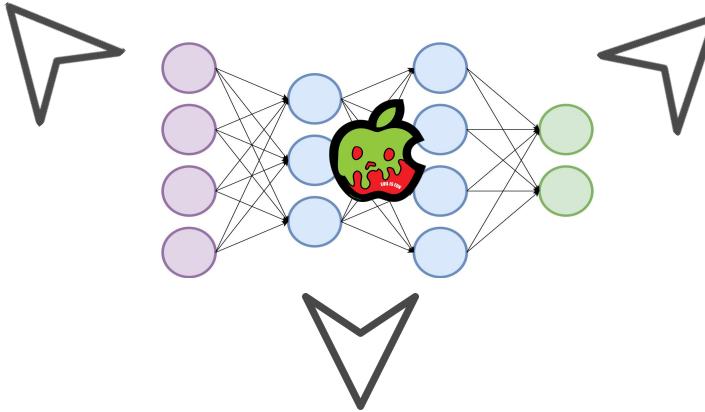


ResNet-101 architecture
Cerber malware

Comparison between the distribution of the weight parameters of the baseline regular model and the **one containing the malware**.

Stealthiness

Stealthiness

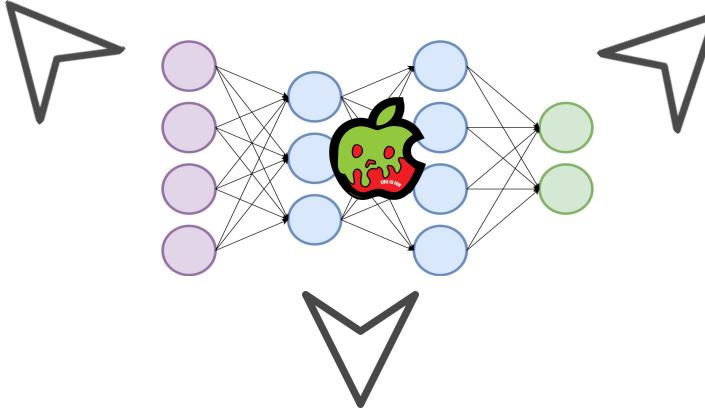


**Impact on
model
performance**

Robustness

Impact on model performance

Stealthiness



**Impact on
model
performance**

Robustness

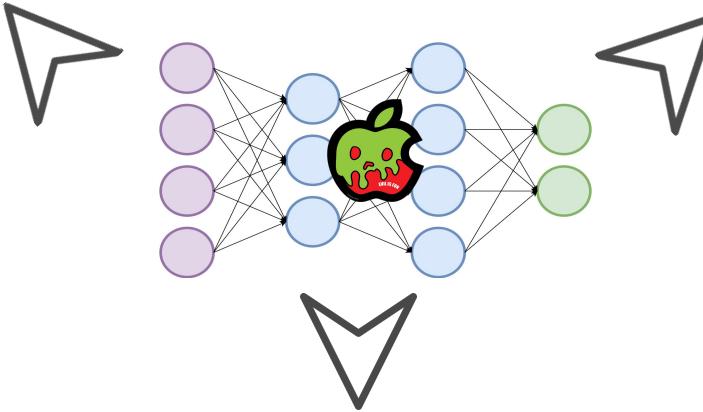
Impact on model performance

	DenseNet		ResNet50		ResNet101		VGG11		VGG16	
Malware	Bas.	Mal.	Bas.	Mal.	Bas.	Mal.	Bas.	Mal.	Bas.	Mal.
Stuxnet	62.13	61.22	75.69	75.34	76.96	76.87	70.13	70.09	73.37	73.34
Destoyer	62.13	52.36	75.69	74.89	76.96	76.79	70.13	70.05	73.37	73.28

Classification accuracy (%) on the ImageNet task before and after embedding different sized malware into different sized neural networks.

Impact on model performance

Stealthiness

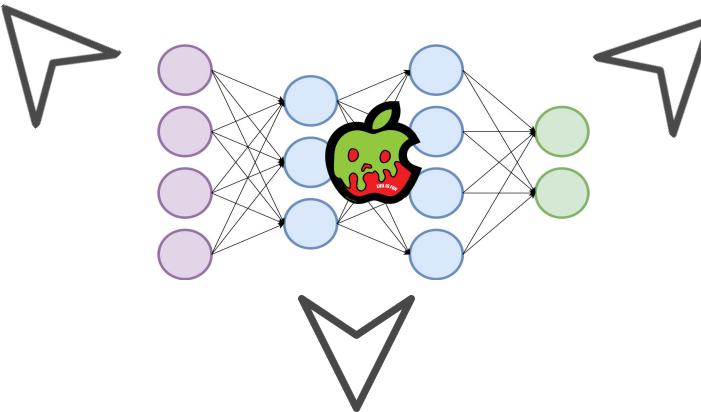


**Impact on
model
performance**

Robustness

Robustness

Stealthiness

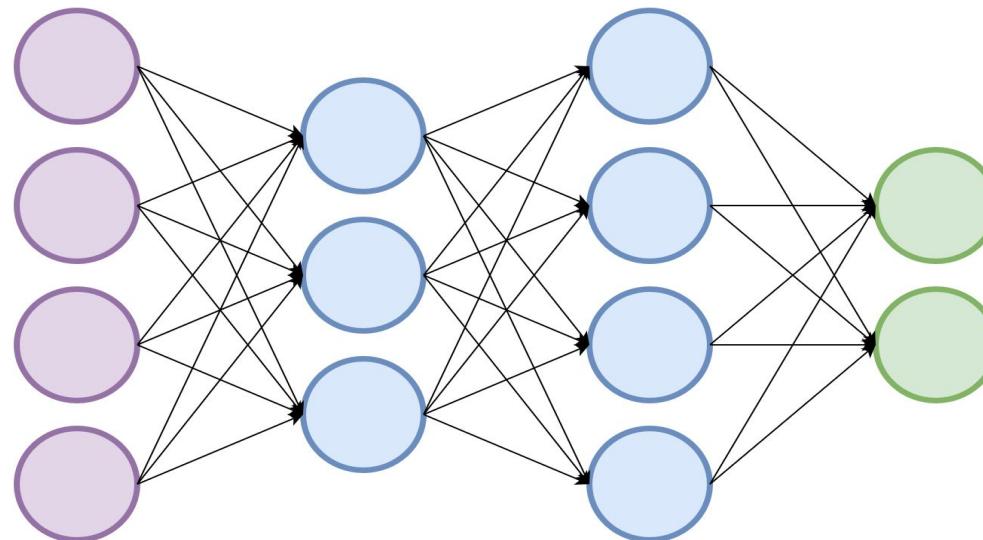


**Impact on
model
performance**

Robustness

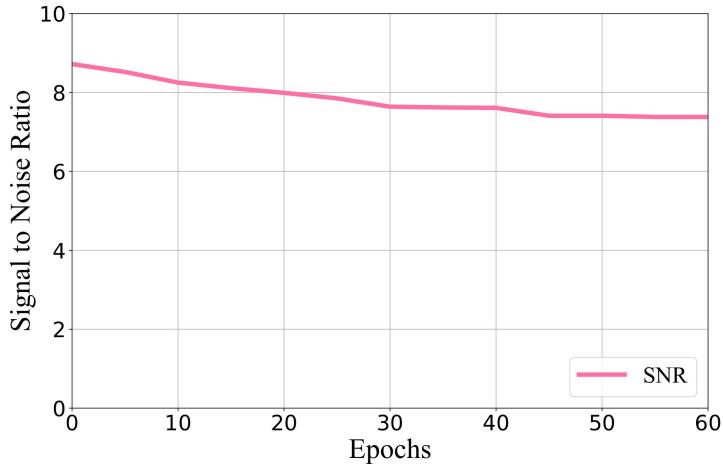
Robustness

1. Robustness towards fine-tuning
2. Robustness towards parameter pruning



Robustness

1. Robustness towards fine-tuning
2. Robustness towards parameter pruning



The change in signal-to-noise ratio of the malicious payload when the VGG11 architecture is trained on CIFAR10 and repurposed to solve the CIFAR100 task.

Robustness

1. Robustness towards fine-tuning
2. **Robustness towards parameter pruning**

Pruning Ratio	Does the payload survive?
25.00%	YES
50.00%	YES
75.00%	YES
90.00%	NO
99.00%	NO



Why so robust?

Binary sequence: [0, 1, 1]



PSK [−1, 1, 1]

Spreading code: [−1, 1, −1, −1, 1]

Chip sequence: [1, −1, 1, 1, −1, −1, 1, −1, −1, 1, −1, 1, −1, −1, −1, 1]

−5

+5

+5



Reading Material

1. Introduction to [Deep Neural Networks](#). (book chapter)
2. Digital communication and CDMA: [Link-1](#), [Link-2](#) (research papers).
3. Hiding malware into Deep Neural Networks [Link-1](#) [Link-2](#) (research papers).