**BACKDOOR-THE SPYMAN**

### A Project Work Report

*Submitted in the partial fulfillment for the award of the degree of*

# BACHELOR OF ENGINEERING

### IN

### INFORMATION SECURITY

### Submitted by:

**Pranav Gupta**

**Akul Sharma**

**Syed Mohd. Arsh**

### University Roll Number

**20BCS3703**

**20BCS3675**

**20BCS3713**

### Under the Supervision of:

**Dr. Monica Luthra**



### CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,

**PUNJAB**

**2020-2024**

***Annexure-4* (A typical specimen of table of contents)**

# Table of Contents

**TITLE** ………………………………………………………………………….…………… 1

**ABSTRACT**………………………………………….…………………………………… 3

**LIST OF TABLES**-………………………………………….………………………… 4

**LIST OF FIGURES**………………………………………….………………………… 5

**INTRODUCTION**----------------------------------------------------------------------------------- 7

**LITERATURE REVIEW**----------------------------------------------------------------------- 8

**LITERATURE REVIEW SUMMARY** --------------------------------------------------9

**PROBLEM FORMULATION** -------------------------------------------------------------- 10

**OBJECTIVES** ------------------------------------------------------------------------------------------ 11

**METHDOLOGY** ------------------------------------------------------------------------------------- 12

**REFRENCES** ------------------------------------------------------------------------------------------- 13

**ABSTRACT**

**In this project, we try to implement a backdoor using python. Advantages and drawbacks of the discussed systems are explained finally**. **An attack using backdoor has the intent to embed hidden back- doors into deep neural networks (DNNs) The attacked models perform well on certain samples, whereas their predictions are changed maliciously if the hidden backdoor is activated by attacker-specified triggers. This threat could happen when the training process is not fully controlled, such as training on third-party datasets or adopting third-party models, which poses a new and realistic threat. Although backdoor learning is an emerging and rapidly growing research area, there is still no comprehensive and timely review of it. In this article, we present the first comprehensive survey of this realm. We summarize and categorize existing backdoor attacks and defenses based on their characteristics, and provide a unified framework for analyzing poisoning-based backdoor attacks. Besides, we also analyze the relation between backdoor attacks and relevant fields (i.e., adversarial attacks and data poisoning), and summarize widely adopted benchmark datasets. Finally, we briefly outline certain future research directions relying upon reviewed works. Backdoor** **programs are applications that allow cybercriminals or attackers to access computers remotely. Backdoors can be installed in both software and hardware components. Many backdoor programs make use of the IRC backbone, receiving commands from common IRC chat clients.**

***List of Tables***

##### Table Title page

### Abstract 3

### Introduction 4

### Literature review 8

### Literature review summary 10

### Project formulation 11

### Objectives 12

### Methodology 15

### 

## List of Figures

##### Figure Title page

**Images 17**

**Refrences 20**

# INTRODUCTION

**While performing a burglary, a thief takes advantage of the vulnerabilities of the house and gets entry into it, bypassing the security. Backdoor attack on the computer system is quite similar.**

**In cybersecurity terms, a Backdoor attack is a malicious way to bypass security and infiltrate computer devices. Like the actual theft, a Backdoor attack allows the cyberattacks to go in and out of the system without being discovered by the security system of the device.**

**After the backdoor entry into the system, cybercriminals can get high-level access to the system and control it. Once control is taken, the attackers can freely perform the intended malicious tasks like gaining remote access, introducing additional malware, hacking the system, stealing personal and financial data, and many more.**

**However, the same Backdoor might also be beneficial. We live in a technologically advanced age where we store all of our private information on our gadgets, just like our adversaries, be they terrorists or a rival group, do. Therefore, we have decided to create a backdoor to aid our government or cyber army in quickly obtaining all the data and their plans.**

# LITERATURE REVIEW

**Imagine you are a cyber spy. Your day job is to tap cryptographically protected communications systems. But how? Straightforward cryptanalysis has long become impractical: the task of breaking modern algorithms, if implemented correctly, far exceeds all computational power available to humanity. That leaves *sabotage.***

**You can target many Achilles heels of a crypto system: random-bit generators, side channels, binary builds, certification authorities, and weak default configurations. You infiltrate the teams that design, implement, and standardize commercial security systems and plant there hidden weaknesses, known as *backdoors*, that later allow you to bypass the cryptography.**

**Backdoors play a crucial**[**role in targeted attacks**](https://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-backdoor-use-in-targeted-attacks.pdf)**because they can be used to take control of affected systems, allowing attackers to steal credentials and establish connections without being found.  With backdoors, attackers can perform the following techniques: port binding, connect-back, connection availability abuse, legitimate platform abuse, common service protocol abuse, protocol/port listening, custom DNS lookup use, and port reuse.**

**The working of backdoor attacks depends on the way they enter the system. As observed, the most common ways, using which a backdoor can enter into a system, are using malware or using backdoor-specific software/hardware. A detailed explanation of these two is as quoted below.**

1. **Backdoor malware**

**An imposter piece of technology, this**[**malware**](https://www.wallarm.com/what/malware-types-and-detection)**pretends to be something else so that actions like data theft, malware installation, and creating a backdoor into the systems can be performed seamlessly.**

**It is also Called backdoor Trojan for its behavioral similarity with Trojans that permit an attacker to reach the core infrastructure of an application/software/network. To understand it better, you must know how Trojan operates.**

**A Trojan is a file with malicious content and can be to use and can be delivered in the form of an email attachment, downloadable file, cyber threats like malware, and so on. To make things worse, Trojans have worm-like abilities that make them competent to replicate and expand. Without demanding any further efforts, Trojan can spread to other systems as well.**

**Regardless of guises, each sort of Trojan is harmful and has the potential to cause serious damage to the target.**

1. **Built-in or proprietary backdoors**

**Think of it as a backdoor to be used by property owners in the case of an emergency. Such types of backdoors are deployed by software or hardware professionals and do not always have ill intentions. They exist as a component of the software and permits owners/developers to gain instant access to the application/software.**

**This immediate access helps them to test a code, fix a software bug, and even detect any hidden vulnerability without being involved in the real/authenticated account creation process.**

**Mostly, they aren’t removed before the final product launch or delivery. At times, they are made secure in order to give instant access to a few users only. But, there are incidents where built-in backdoors are delivered with the original software by fault or negligence**.

### 2.1)Literature Review Summary

### Table 2.1: Literature review summary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year and citation** | **Article Title** | **Purpose of the study** | **Tools/ Software used** | **Comparison of technique done** | **Source (Journal/ Conference)** | **Findings** |
| 2022 | Backdoor the spyman | To create and implement a Backdoor | Python IDE  Python code editor | Python libraries are used | Journal | Intrusion using a backdoor was successful. |

# PROBLEM FORMULATION

**When it comes to backdoor attacks we have various types. But mainly we have two types i.e. Poisoning based and Non poisioning attacks. The examples of poisoned samples generated by different types of backdoor attacks:**

**In the visible attack, the backdoor trigger is a white square stamped on the bottom right corner of the poisoned image, which is visible.**

**In the invisible attack, the trigger is a noise with a small magnitude, which is invisible. Moreover, the target label of the poisoned image is different from the ground-truth label of its benign version in the poison-label attack, whereas these labels are the same in the clean-label attack.**

**In the optimized attack, the trigger is optimized through the targeted universal adversarial attack associated with the target class instead of a simple handcraft pattern.**

**The poisoned image is exactly the same as its benign version in the semantic attack. In this case, the trigger is the combination of two semantic objects (i.e., “bird” and “human”). Images containing these objects simultaneously will be classified by the infected models as the “car.”**

**In the sample-specific attack, the trigger patterns are sample-specific instead of sample-agnostic.**

**In the physical attack, the (digital) poisoned image is captured by the camera from the physical space.**

**Different from all-to-one attacks where all poisoned samples have the same target label, different poisoned samples may have different target labels in the all-to-all attack.**

# 

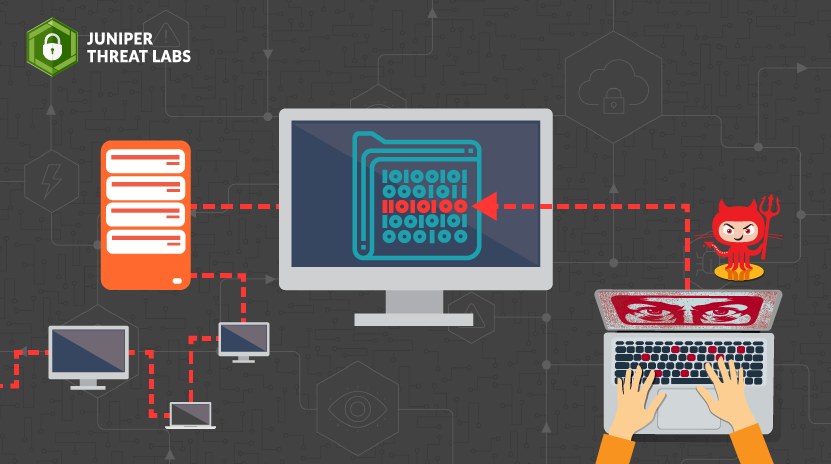
# OBJECTIVES

**In this project implementation a successful connection establishment is to be made with the remote host. A computer that resides in some distant location from which data are retrieved. It typically refers to a server in a private network or the public Internet. However, it can also refer to a user's PC in another location that is accessed over the Internet for file transfer or remote control operation. Up until the 1990s, a remote host was almost always a single, centralized computer system that was accessed using terminals directly connected or over private lines or via a dial-up modem**

**Interaction with the remote host is possible, then remote shell can be deployed. A remote shell is a tool for executing commands on a device through a command-line shell (a program enabling computer control through commands) on another. A remote shell can be used for remote configuration of devices, for monitoring, detecting and fixing bugs, for working on a remote server, etc.; in addition, remote shell-type tools are used by cybercriminals to gain access to targeted computers.  Remote shell session can be initiated either by a local device (which sends commands) or a remote one (on which commands are executed). The former is referred to as a bind shell, the latter as a reverse shell. Key logger can also be started. A keylogger is a tool that can record and report on a computer user's activity as they interact with a computer. The basic functionality of a keylogger is that it records what you type and, in one way or another, reports that information back to whoever installed it on your computer. (We'll go into the details in a moment.) Since much of your interactions with your computer—and with the people you communicate with via your computer—are mediated through your keyboard, the range of potential information the snooper can acquire by this method is truly vast, from passwords and banking information to private correspondence.**

**Some keyloggers go beyond just logging keystrokes and recording text and snoop in a number of other ways as well**

**Various other things will be applied i.e. Logging of keys , taking of screenshot, viewing the remote host system information etc. can be done.**



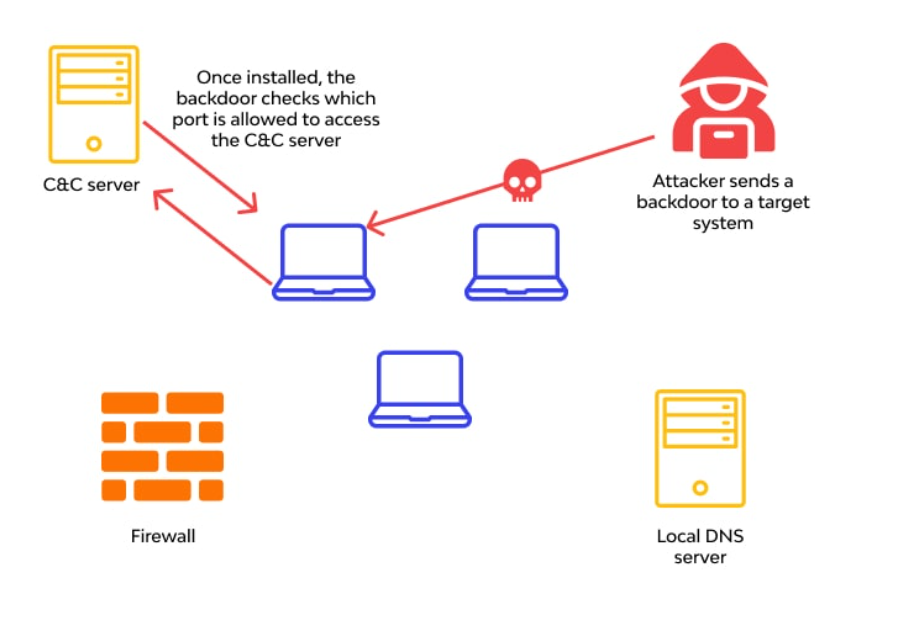
# METHODOLOGY

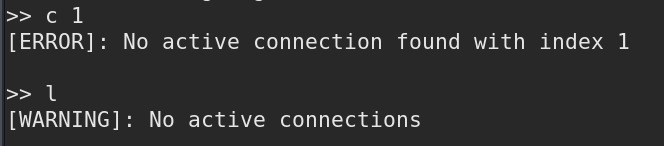
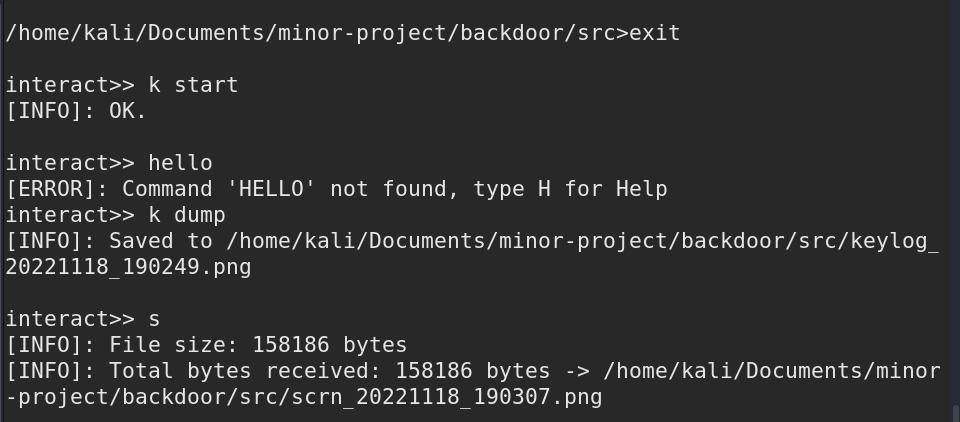
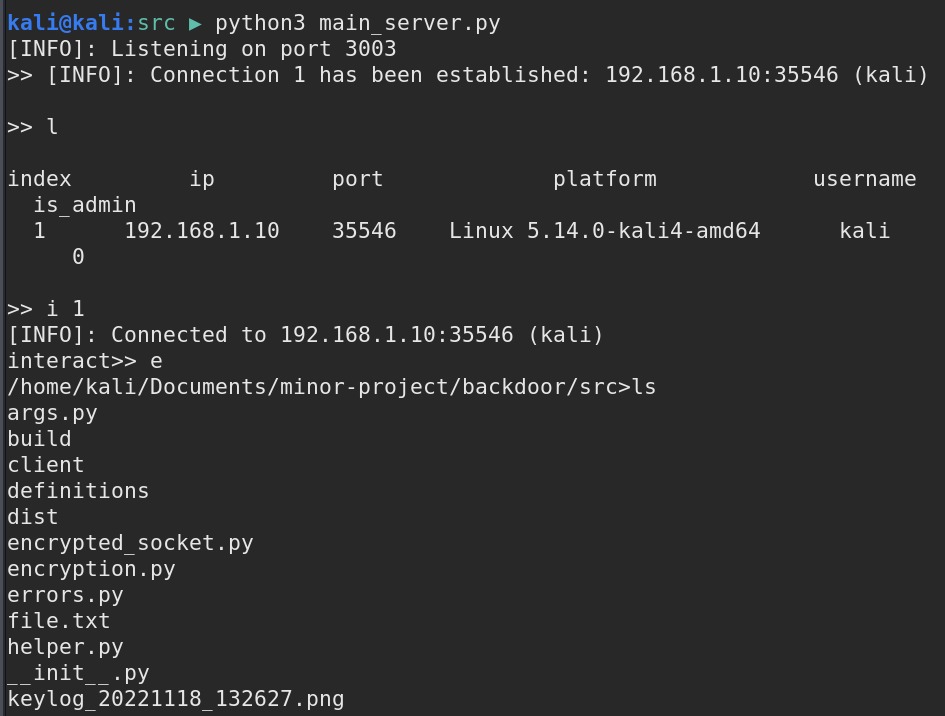
**The following methodology will be followed to achieve the objectives defined for proposed research work:**

* **Detailed study of backdoor poisoning will be done**
* **Installation and hand on experience on existing approaches of Backdoor using python will be done. Relative pros and cons will be identified.**
* **Various parameters will be identified to evaluate the proposed system.**
* **Comparison of new implemented approach with exiting approaches will be done.**

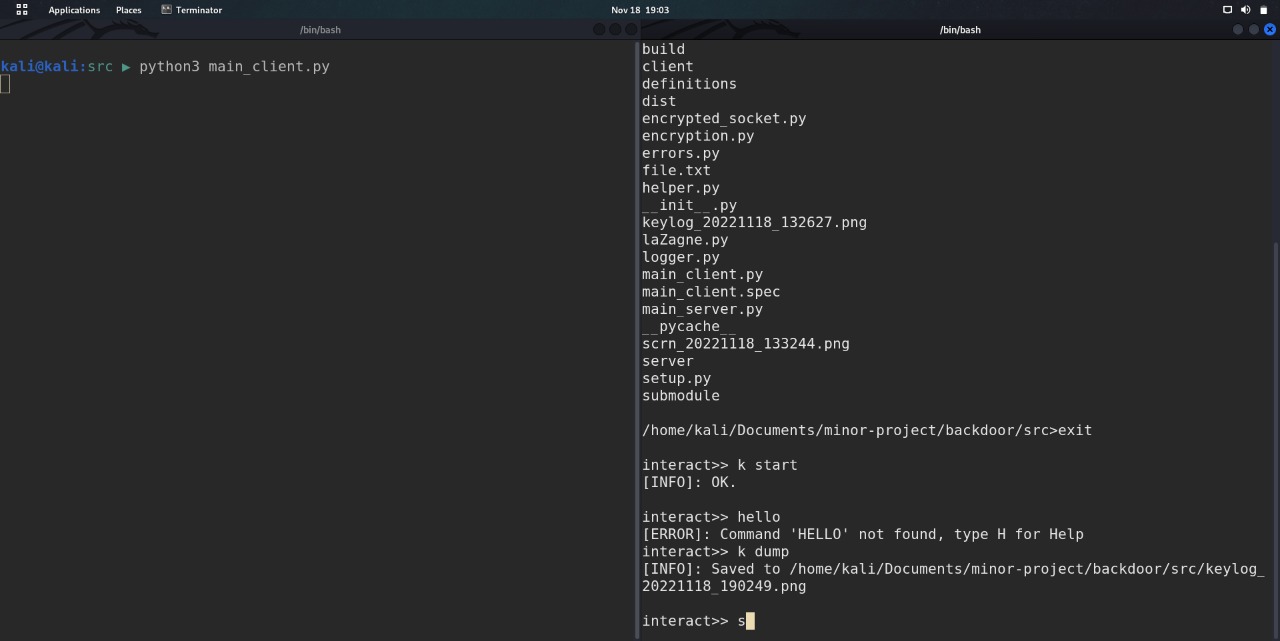
**In this proposed system, the python IDLE will be used to establish the connection between the client and the server; in this case, the client will be the remote host trying to connect to the server hosted by the adversary. After the successful establishment of the connection, the adversary will be able to remotely control the victim's system by spawning up a remote shell for listing out the files in the system and also will be able to see the configuration of the compromised system. Not only this adversary will also be able to launch a keylogger attack and take screenshots.**

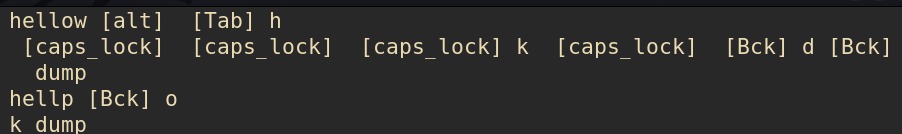
**Working of the Backdoor the Spyman**

****









1. **TENTATIVE CHAPTER PLAN FOR THE PROPOSED WORK**

#### CHAPTER 1: INTRODUCTION

**This chapter will cover the overview of the proposed project.**

#### CHAPTER 2: LITERATURE REVIEW

**This chapter include the literature available for Gesture Detection System and the findings of the researchers will be highlighted which will become basis of current implementation.**

#### CHAPTER 2: BACKGROUND OF PROPOSED METHOD

**This chapter will provide introduction to the concepts which are necessary to understand the proposed system.**

#### CHAPTER 4: METHODOLOGY

**This chapter will cover the technical details of the proposed approach.**

#### CHAPTER 5: EXPERIMENTAL SETUP

**This chapter will provide information about the subject system and tools used for evaluation of proposed method.**

#### CHAPTER 6: RESULTS AND DISCUSSION

**The result of proposed technique will be discussed in this chapter.**

#### CHAPTER 7: CONCLUSION AND FUTURE SCOPE

**The major finding of the work will be presented in this chapter. Also directions for extending the current study will be discussed.**

**6.) REFERENCES**

[1] http://www.robots.ox.ac.uk/˜vgg/

software/vgg\_face/, 2015. VGG Face

Descriptor.

[2] Wieland Brendel, Jonas Rauber, and Matthias Bethge.

Decision-based adversarial attacks: Reliable attacks

against black-box machine learning models. In Proc.

of ICLR, 2018.

[3] Tom B Brown, Dandelion Mane, Aurko Roy, Mart ´ ´ın

Abadi, and Justin Gilmer. Adversarial patch. In Proc.

of NeurIPS Workshop, 2017.

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In Proc. of

IEEE S&P, 2017.

[5] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo,

Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian

Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering.

arXiv preprint arXiv:1811.03728, 2018.

[6] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi,

and Cho-Jui Hsieh. Zoo: Zeroth order optimization

based black-box attacks to deep neural networks without training substitute models. In Proc. of AISec,

2017.

[7] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and

Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint

arXiv:1712.05526, 2017.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai

Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In Proc. of CVPR, 2009.

[9] Yansong Gao, Chang Xu, Derui Wang, Shiping Chen,

Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks.

[10] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks

on deep neural networks. IEEE Access, 7:47230–

47244, 2019.

[11] Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and

Dawn Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in AI systems.

arXiv preprint arXiv:1908.01763, 2019.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian

Sun. Deep residual learning for image recognition. In

Proc. of CVPR, 2016.

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten,

and Killian Q Weinberger. Densely connected convolutional networks. In Proc. of CVPR, 2017.

[14] Diederik P Kingma and Jimmy Ba. Adam: A

method for stochastic optimization. arXiv preprint

arXiv:1412.6980, 2014.

[15] Ram Shankar Siva Kumar, Magnus Nystrom, John

Lambert, Andrew Marshall, Mario Goertzel, Andi

Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning–industry perspectives. arXiv

preprint arXiv:2002.05646, 2020.

[16] Alexey Kurakin, Ian Goodfellow, and Samy Bengio.

Adversarial machine learning at scale. In Proc. of

ICLR, 2017.

[17] Shaofeng Li, Benjamin Zi Hao Zhao, Jiahao Yu, Minhui Xue, Dali Kaafar, and Haojin Zhu. Invisible

backdoor attacks against deep neural networks. arXiv

preprint arXiv:1909.02742, 2019.

[18] Yiming Li, Tongqing Zhai, Baoyuan Wu, Yong Jiang,

Zhifeng Li, and Shutao Xia. Rethinking the trigger

of backdoor attack. arXiv preprint arXiv:2004.04692,

2020.

[19] Cong Liao, Haoti Zhong, Anna Squicciarini, Sencun

Zhu, and David Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. arXiv preprint arXiv:1808.10307, 2018.

[20] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg.

Fine-pruning: Defending against backdooring attacks

on deep neural networks. In Proc. of RAID, 2018.

[21] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn

Song. Delving into transferable adversarial examples

and black-box attacks. In Proc. of ICLR, 2016.

[22] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing

Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain

stimulation. In Proc. of CCS, 2019.

[23] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan

Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang.

Trojaning attack on neural networks. In Proc. of

NDSS, 2018.

[24] Giulio Lovisotto, Henry Turner, Ivo Sluganovic, Martin Strohmeier, and Ivan Martinovic. Slap: Improving

physical adversarial examples with short-lived adversarial perturbations. arXiv preprint arXiv:2007.04137,

2020.

[25] Shiqing Ma, Yingqi Liu, Guanhong Tao, Wen-Chuan

Lee, and Xiangyu Zhang. Nic: Detecting adversarial

samples with neural network invariant checking. In

Proc. of NDSS, 2019.

[26] Agnieszka Mikolajczyk and Michal Grochowski. Data

augmentation for improving deep learning in image

classification problem. In Proc. of IIPhDW, 2018.

[27] Sinno Jialin Pan and Qiang Yang. A survey on transfer

learning. IEEE Transactions on knowledge and data

engineering, 2009.

[28] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow,

Somesh Jha, Z. Berkay Celik, and Ananthram Swami.

Practical black-box attacks against machine learning.

In Proc. of Asia CCS, 2017.

[29] Nicolas Papernot, Patrick McDaniel, Somesh Jha,

Matt Fredrikson, Z. Berkay Celik, and Ananthram

Swami. The limitations of deep learning in adversarial

settings. In Proc. of Euro S&P, 2016.

[30] Sylvain Paris. A gentle introduction to bilateral filtering and its applications. In Proc. of SIGGRAPH.

2007.