BACKDOOR – THE SPYMAN

A Project Work Synopsis

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

BACHELOR OF ENGINEERING

INFORMATION SECURITY

Submitted by:

PRANAV GUPTA
20BCS3703
SYED MOHD ARSH MEHDI RIZVI
20BCS3713
AKUL SHARMA
20BCS3675

Under the Supervision of:

MONICA LUTHRA

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413, PUNJAB

NOVEMBER, 2022

Table of Contents

1 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.

1.1 Problem Definition

A security backdoor is a way for governments, law enforcement, or others to gain access to encrypted or locked information. If you want to use some kind of encryption key to protect your data from others, it's like locking your front door. Having a backdoor is like another way to get my information without my knowledge or help.

One similar problem was that the government wanted Apple's help to unlock the iPhone of one of the San Bernardino terrorists, but Apple refused to help. Apple says that if it helps the government break into this iPhone, it will help the US government and others around the world break into countless devices, which could lead to a violation of public privacy. people. The government had a pretty strong argument – who could oppose the government's war on terror?

In this case, a backdoor would have been very beneficial for the government. A backdoor is a malware type that negates normal authentication procedures to access a system. As a result, remote access is granted to resources within an application, such as databases and file servers, giving perpetrators the ability to remotely issue system commands and update malware.

Once installed, backdoors are difficult to remove. Traditionally, detection involved using software scanners to look for signatures of known malware in the host file system. However, this process is error-prone. Backdoor shell files are almost always obfuscated through the use of alias names and, more importantly, encryption (sometimes even multiple layers of encryption).

Detection is even more complicated because many applications are built on external frameworks using third-party plugins; these are sometimes loaded with built-in vulnerabilities or backdoors. Empirical analyzers and signature-based rules may not be able to detect code hidden in such frameworks.

1.2 Project Overview

In this project, we are designing a backdoor that can act as a spy man for the military or the government. Now the government can use this backdoor to get intelligence from terrorists or any other adversary. Along with impressive advancements affecting all aspects of our society, AI technology based on deep neural networks (DNNs) raises security concerns. While active attacks at the time of testing attracted the researchers' initial attention, backdoor attacks, which exploit the ability to corrupt DNN models by interfering with training, represent another serious threat to the reliability of AI techniques. In backdoor attacks, the attacker corrupts the training data to cause misbehavior during testing. However, the trial time error is only fired when there is a trigger event. This way, the damaged network continues to work as expected for normal entries, and malicious behavior occurs only when an attacker decides to enable a hidden backdoor in the network. Recently, backdoor attacks have been the subject of an intense field of research focused on both developing new attack classes and recommending possible countermeasures. The purpose of this review is to review the work published to date, classifying the different attack and defense types proposed to date. The classification of the analysis instruction is based on the extent of the attacker's control over the training process and the defender's ability to verify the integrity of the data used for training and monitoring activities. The behavior of DNNs during training and testing. time. Therefore, the proposed analysis is suitable to highlight the strengths and weaknesses of attacks and defenses in the application scenarios in which they operate.
We all have a vested interest in a safe and secure Internet. Even US intelligence agencies have been victims of poor security, most notably the Shadow Brokers data breach. Perhaps this is why, despite arguments against the dangers of ubiquitous encryption, the US government remains a major sponsor of Tor, largely because it believes in the value of national security. involved in enabling secure communications. Modern technology is fluid and fast. To keep up, law enforcement must adapt. But finding ways to weaken encryption only looks back to the past instead of focusing on the direction the technology is going. We should have conversations about new investigative techniques, not trying to maintain the access we had before encryption was so pervasive, especially when so much is at stake.

1.2 Software Specification

- Python IDE
- Python code editor

The following Python libraries have been used in this project

- sys
- socket
- pynput
- re
- argparse

2 LITERATURE REVIEW

Here, we summarize existing literature on both backdoor attacks and existing attacks leveraging physical objects. Backdoor Attacks and Defenses. An attacker launches backdoor attacks against a DNN model in two steps. During model training, the attacker poisons the training dataset by adding samples associating inputs containing a chosen pattern (the trigger δ) with a target label yt. This produces a backdoored model that correctly classifies benign inputs but "misclassifies" any input containing the trigger δ to the target label yt. At inference time, the attacker activates the backdoor by adding the trigger δ to any input, forcing the model to classify the input as yt. First proposed in [10, 23], backdoor attacks have advanced over the years to employ human imperceptible triggers [19, 17] and more effective embedding techniques [34, 18], and can even survive transfer learning [45]. Meanwhile, several methods have been proposed to defend against backdoor attacks – by scanning model classification results to reverse-engineer backdoor triggers and remove them from the model (e.g. Neural Cleanse [42]), pruning redundant neurons to remove backdoor triggers (e.g. STRIP [9]), or detecting the presence of poisoning data in the training dataset (e.g. Activation Clustering [5], Spectral Signatures [38]). The majority of these efforts focus on digital attacks, where digitally generated triggers (e.g. a random pixel pattern) are digitally appended to an image.

Clean-label poisoning attacks [35, 36] can exhibit similar, unexpected behavior on specific inputs, but misclassify a specific set of benign inputs usually from a single label, and do not generalize based on a trigger.

Physical Backdoor Attacks. Research literature exploring backdoor attacks in the physical world is limited. One work [10] showcased an example where a DNN model trained to recognize a yellow square digital trigger misclassifies an image of a stop sign with a yellow post-it note. Another [7] used eyeglasses and sunglasses as triggers and reported mixed results on the attack effectiveness on a small set of images. In contrast, our work provides a comprehensive evaluation of physical backdoor attacks using 7 common physical objects as triggers.

3 PROBLEM FORMULATION

1. An encryption backdoor would aid law enforcement and intelligence agencies in their efforts to combat and prevent crime. This would also expedite investigations because agencies would be able to intercept communications and search suspects' electronic devices to gather data. Officials claim that a backdoor would greatly benefit investigations of terrorism and hate crimes.

2. It can be used to restore user access when there is no other option. It can also be utilized for troubleshooting purposes.

3. It can help uncover child sexual abuse material (CSAM) hidden in encrypted messaging applications.

4 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 the 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. A 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. A 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 several other ways as well
Various other things will be applied i.e. Logging of keys, taking screenshots, viewing the remote host system information, etc. can be done.

The Main Objectives of this project are as follows:

- To obtain important credentials of an adversary.
- To know the intentions of the adversary.
- Prediction of future attacks.
- To gain intelligence against the adversary

5 METHODOLOGY

When building a backdoor, there are two components needed:

1. Client: These are the components that will be installed on the victim’s computer, initiates a connection to the attacker’s network, accepts commands, and sends data to and fro.
2. Server: This is the component that will be installed on the attacker system acting as the entry point listening to the client connection, accepting the connection if it’s from the victim, sending commands, and receiving data.

To make this work, we’ll be using the Socket module that comes with built-in Python. The socket module is used to send data/messages to and fro over a network. In this case, the server will be sending commands (messages), the client receives a message (commands), sends a reply (data), and vice versa.

So we are going to be building two components: client.py and server.py.
client.py

1. We will import the modules we’ll be using – socket module (initiating our network connection) and subprocess (for running commands in shell)
2. Declare the attackers (our) remote REMOTE_HOST and REMOTE_PORT. Update the REMOTE_HOST with your IP or local host.
3. Create the socket connection for the client and connect it to our REMOTE server
4. Then we will add a while loop, that keeps listening and waiting for messages or commands
5. We will extract the message from .recv(1024), decode it to a string, and pass it to the subprocess program responsible for running the command.
6. After running the command, we will check for both output and error, then send both along over the network.

server.py

1. We will import the socket module (for listening and accepting network connection)
2. Declare our HOST and PORT. Update the HOST with your IP or local host.
3. Create the socket connection, listen to the incoming connection, and accept, if any (when the user runs the program).
4. We will create a while loop to maintain the connection between the client and server components.
5. From here we can then ask the attacker to enter a command, send the command, and get a response sent by the client component.

To test this, you will need to run the two components simultaneously and connect to the same HOST and PORT.

CONCLUSION AND FUTURE SCOPE

We have explained what a backdoor is, how innocent-looking and dangerous it could be and how to build one. These kinds of programs are a big threat because it’s hard to detect them as they’re hidden in a simple program and can look like normal software. Most backdoor programs are bundled as .dll in Windows, or binary or packaged in Python GUI frameworks like Kivy.

Don’t use this to gain unwanted access to any computer without permission because it is illegal. Do not use this program for any illegal reasons even though it’s in its simplest form.

- It can be used to gather information about potential terrorist or cyber-attacks.
- It can support police investigations and crime-solving efforts.
- The army can utilize it to track down their adversaries and acquire intelligence so they can be ready and take appropriate actions.
- It can be used to find out if someone is leaking sensitive information that could jeopardize national security.

6 TENTATIVE CHAPTER PLAN FOR THE PROPOSED WORK

CHAPTER 1: INTRODUCTION

This chapter will cover the overview of ....

CHAPTER 2: LITERATURE REVIEW

This chapter includes the literature available for..............................The findings of the researchers will be highlighted which will become the basis of the current implementation.

CHAPTER 2: BACKGROUND OF PROPOSED METHOD

This chapter will provide an introduction to the concepts which are necessary to understand theproposed 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 the evaluationof the proposed method.

CHAPTER 6: RESULTS AND DISCUSSION

The result of the 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 extendingthe current study will be discussed.

7 REFERENCES

[1] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Killian Q Weinberger. Densely connected convolutional networks. In Proc. of CVPR, 2017.

[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 defense against trojan attacks on deep neural networks. In Proc. of ACSAC, 2019.

[10] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badness: 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.