MetaNet

A botnet with Metasploit integration

By :

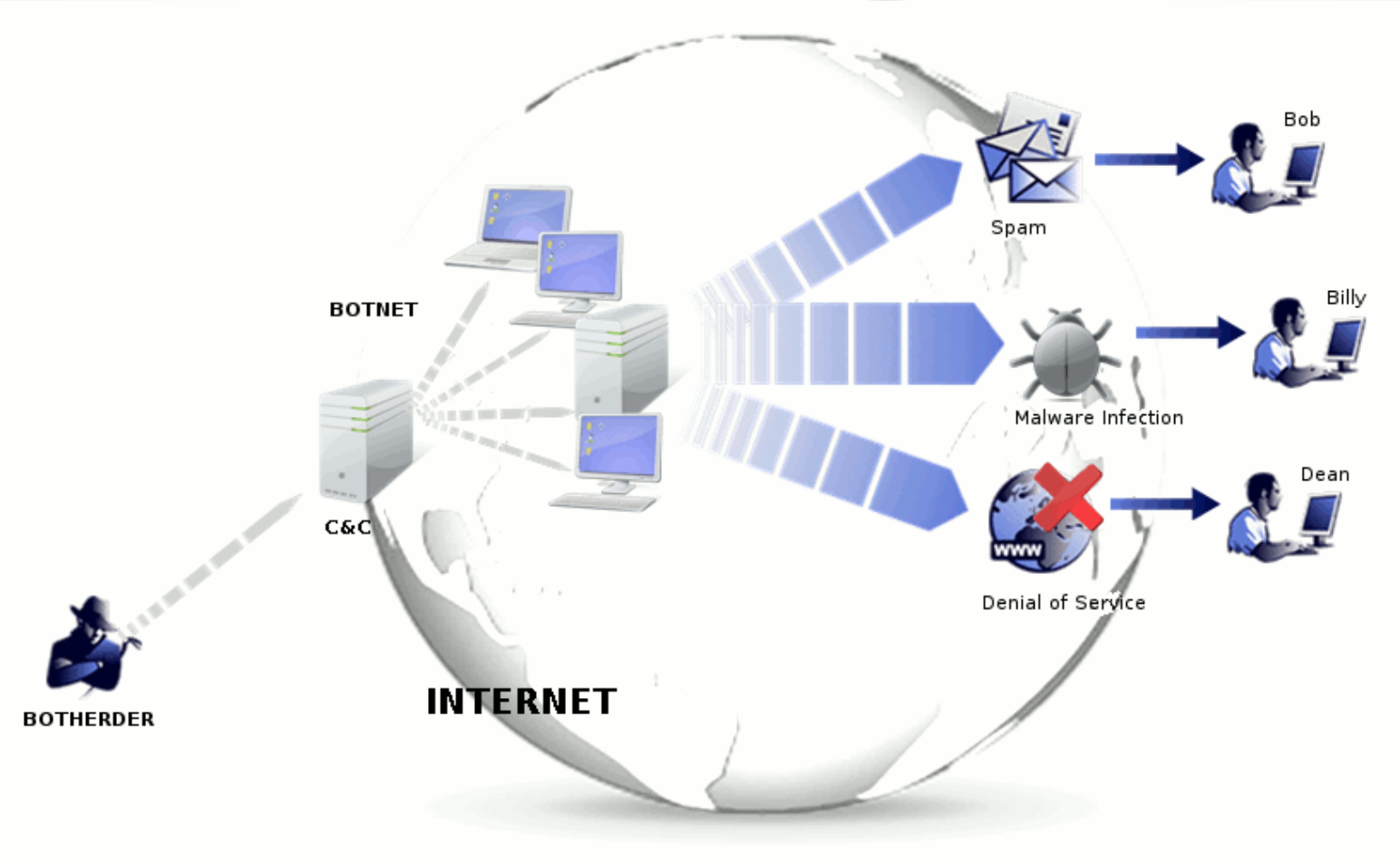
Matan Ramrazker, Guy Gelber

What is a Botnet

- A Botnet is a software that is designed to perform simple automated and usually cyclical operations.
- Botnet management is performed remotely by botnet master that is able to send the bots tasks to perform.

- Botnet try's to distribute itself through the network.

Botnet Diagram



What is an Exploit

An exploit is a piece of software, a chunk of data, or a sequence of commands that takes advantage of a bug or vulnerability in order to cause unintended or unanticipated behavior to occur on computer software or hardware.

Exploits can be run:

1. Locally – Privilege escalation.
2. Remotely – Buffer overflow, backdoor, etc…

What is Metasploit

- Metaspoit Framework is a open source library for penetration and use for developing and executing exploit code against endpoints.
- Metasploit can be used to test the Vulnerability of computer systems that use a software that is vulnerable.
- Metasploit framework has the world's largest database of public, tested exploits.

Our project - MetaNet

- Metanet project integrates those last concepts into one software, Metasploit, Exploit, Botnet.

- Metanet include three major parts, A bot software that is running on compromised machine that includes Metasploit, Server side application saves the bot data and negotiates between the bots and the client side application, Client side application used to control the bots remotely.

The Bot

- The Bot is a multithreaded program that is installed on a compromised computer.
- Our bot coded in C++ language with boost framework, and works on a Linux machine.
- The bot sends every 30 second a "Sign of life" message to the server to inform its online and to get a list of tasks from the server to be executed.
- The bot uses a variation of a concept from networking called "Slow start" that will be describe in the next slide.
- The bot starts a port scan every week in order to find vulnerable machines to compromise.
- Three design patterns are included in our code: Iterator, Factory, Singleton.

Our slow start variation

- Our variation of slow start try to help with server redundancy and provide high availability.
- The bot uses several server domain names in order to provide high availability in case a server is down.
- The bot try's to find an online server, if the server is down, the time to wait to connect to the next server is increased (until predefined limit) in order to achieve quieter network and make the bot more stealth.
- 2,4,8,16…LIMIT seconds.

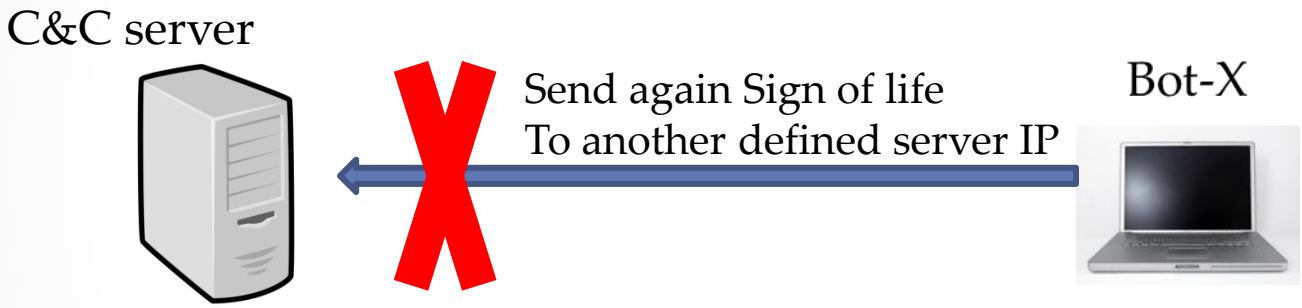
Slow-Start Flow

As Bot starting

If the bot gets a connection error it wait 2 seconds.

Slow-Start Flow

After 2 seconds…



If this server also isn't responding the bot will wait 2*2 sec and so on until reaches its defined limit.

Our port scanner

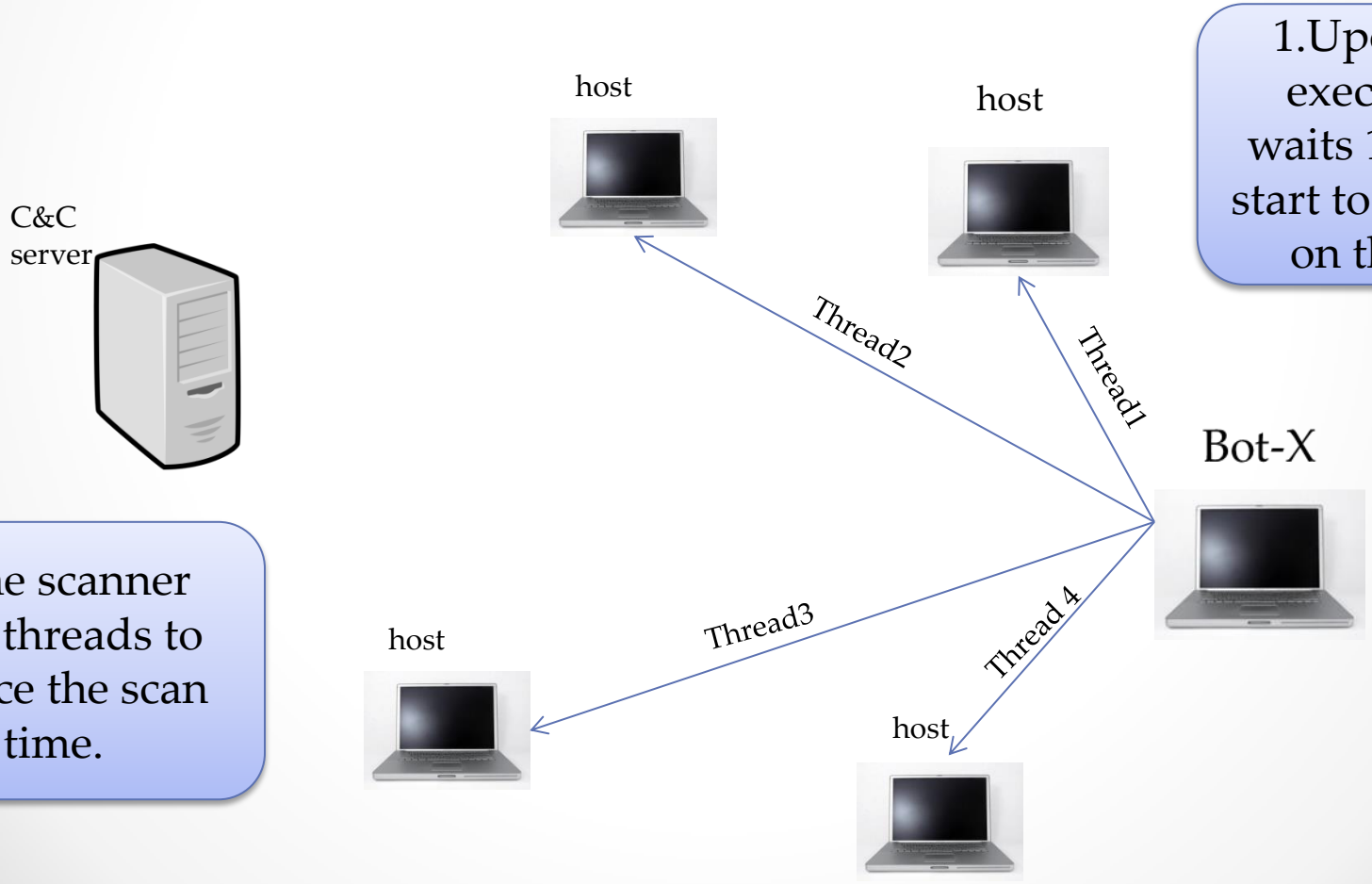
- The purpose of a botnet is to distribute itself by infecting other machines on the network.
- In order to achieve this, the bot scans the local area network for open ports that can be exploited.
- The port scanner uses TCP protocol to scan the ports and creating a full handshake to indicate if a port is open or not on a scanned host.

Our port scanner

- When a bot finds an open port it run will run Metasploit in order to execute an exploit that uses relevant port on the machine.

- If the exploit succeed , it will execute a command that downloads the bot package from the server , install it and run it.

Scanner execution flow



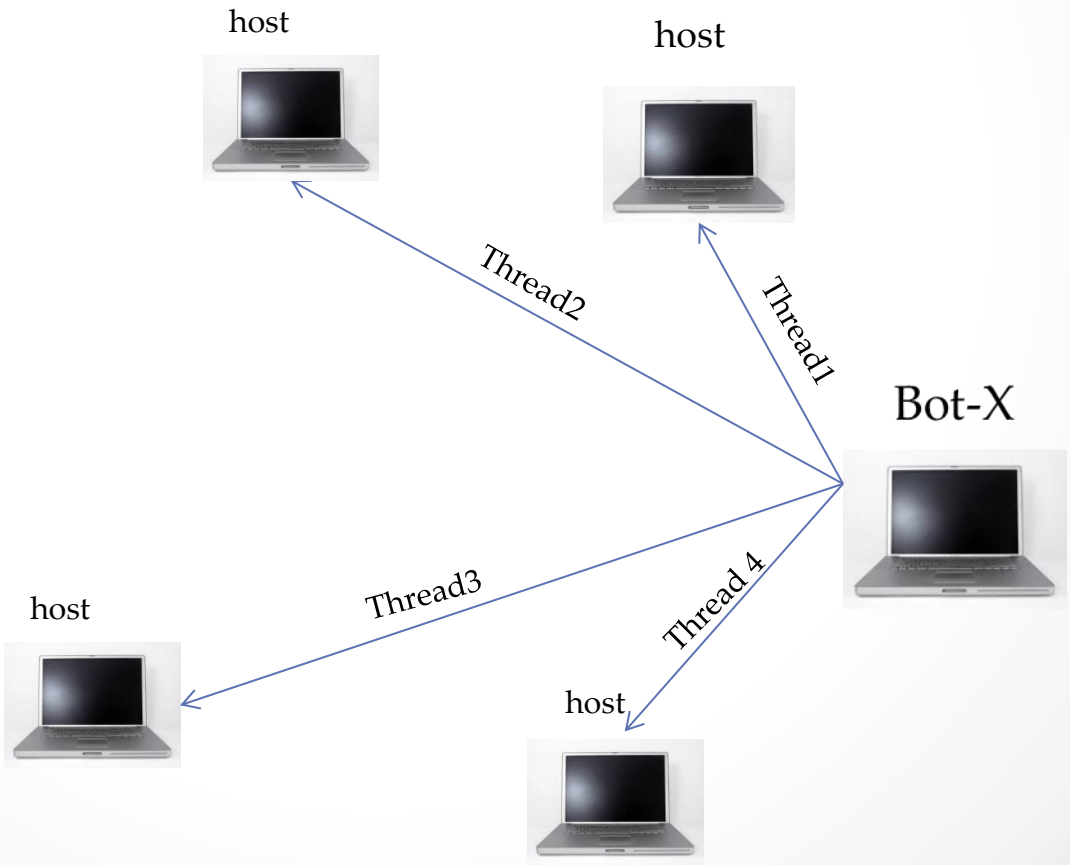
1.Upon Bot-X execution, it waits 10 sec and start to scan hosts on the LAN

2.The scanner uses threads to reduce the scan time.

Scanner execution flow

Every thread takes an IP address from the hosts available on the local network and scans port on this host

C&C server

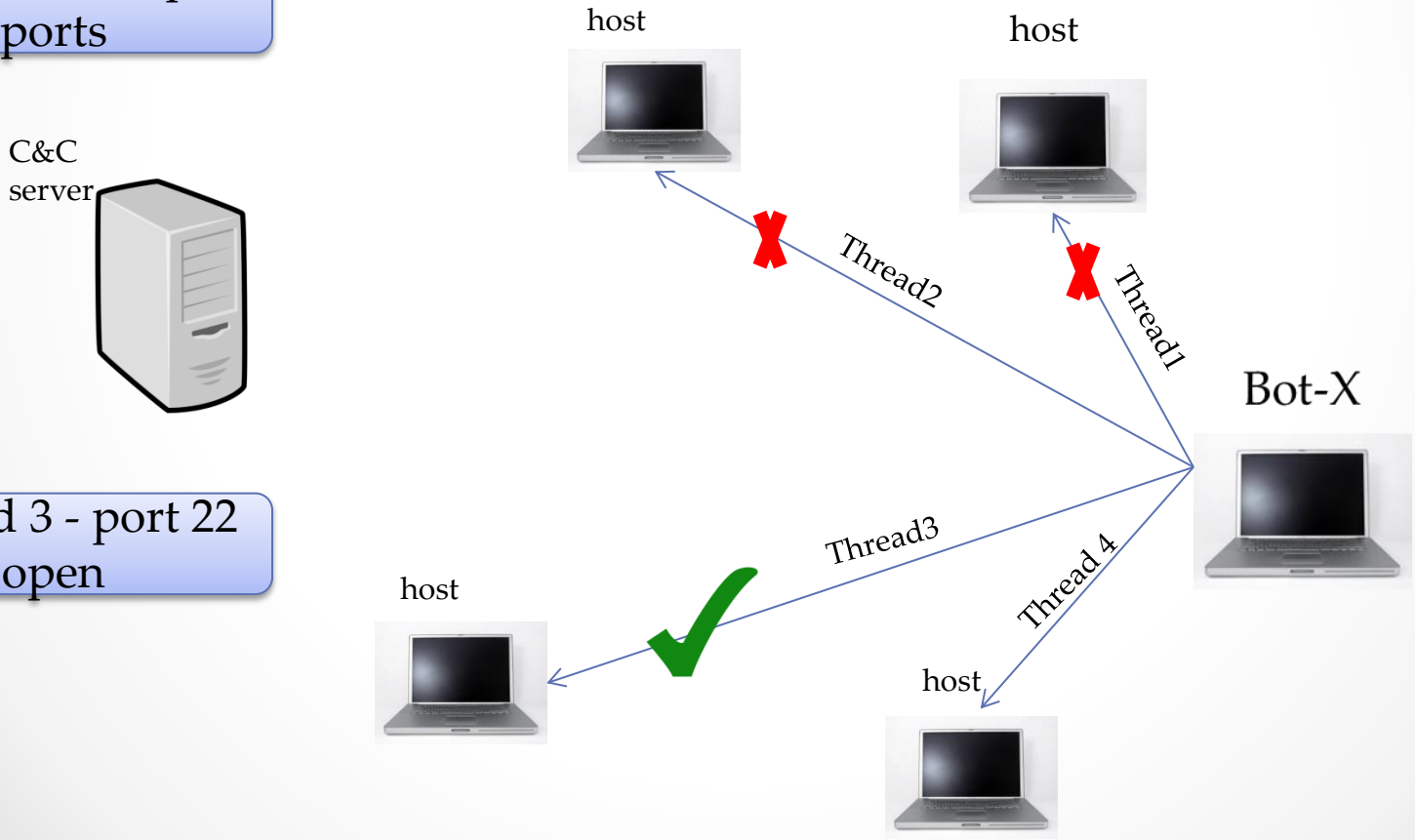


Scanner execution flow

Therad 1- no open ports

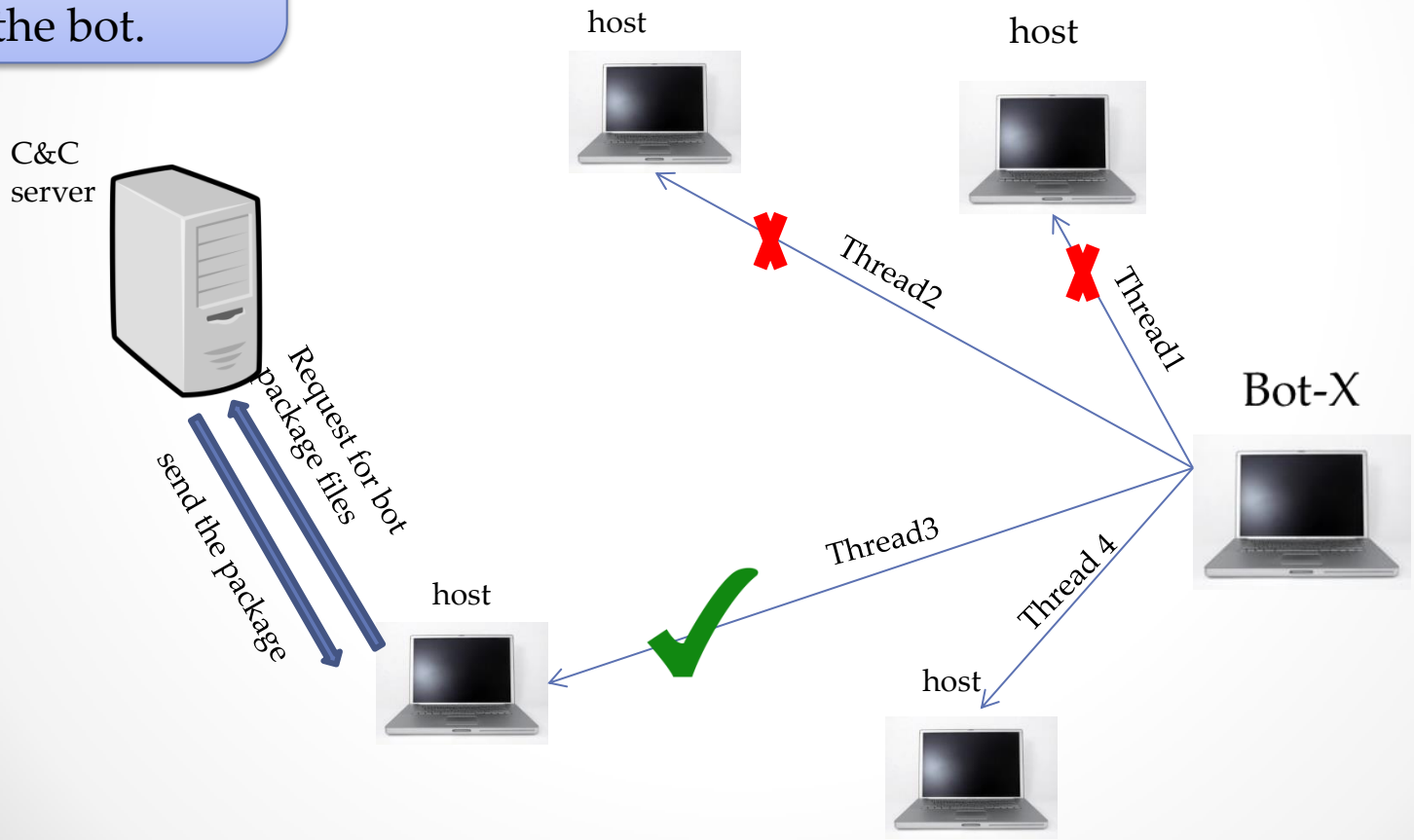
Therad 2- no open ports

Therad 3 - port 22 open



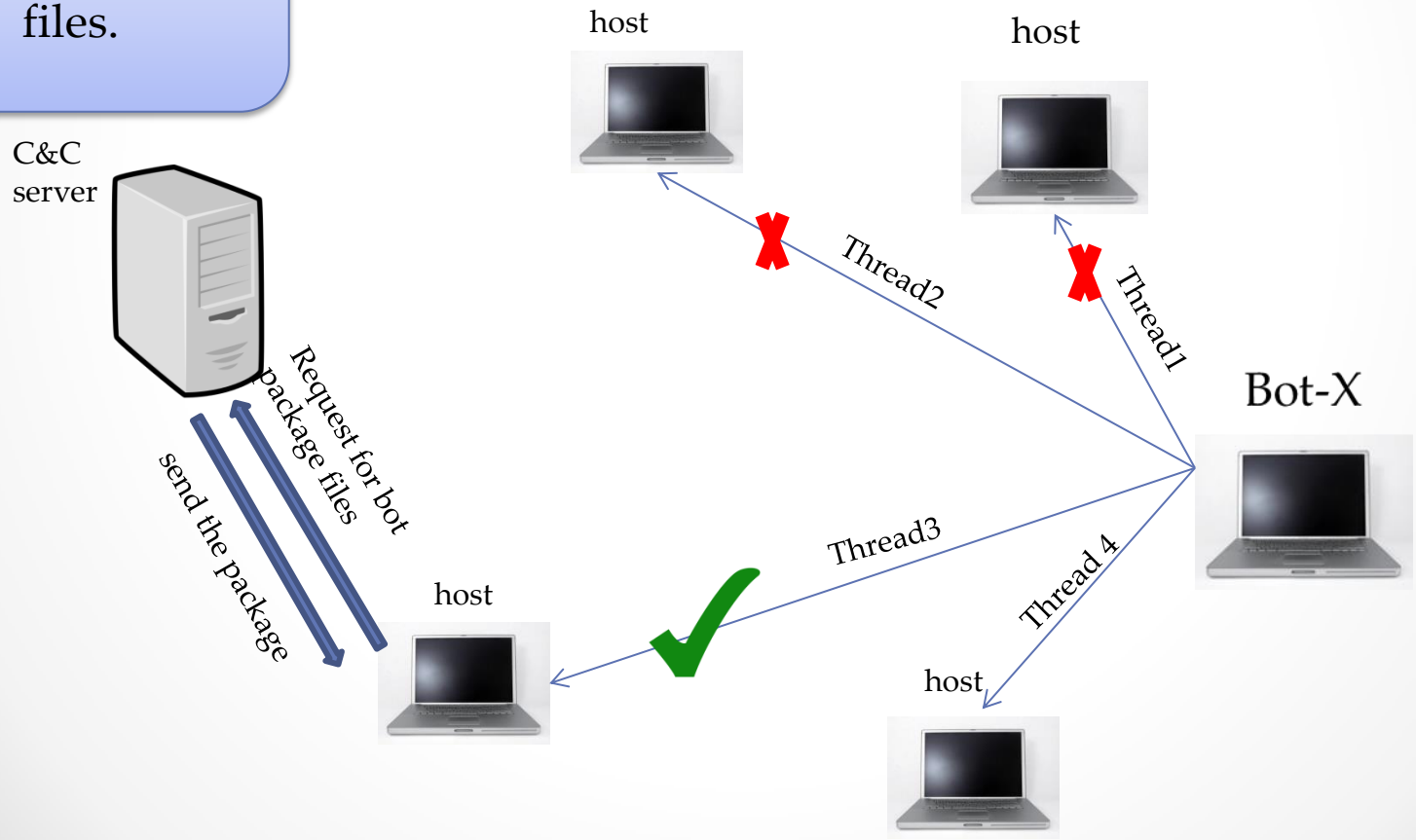
Scanner execution flow

The victim ask for bot package files in order to install the bot.



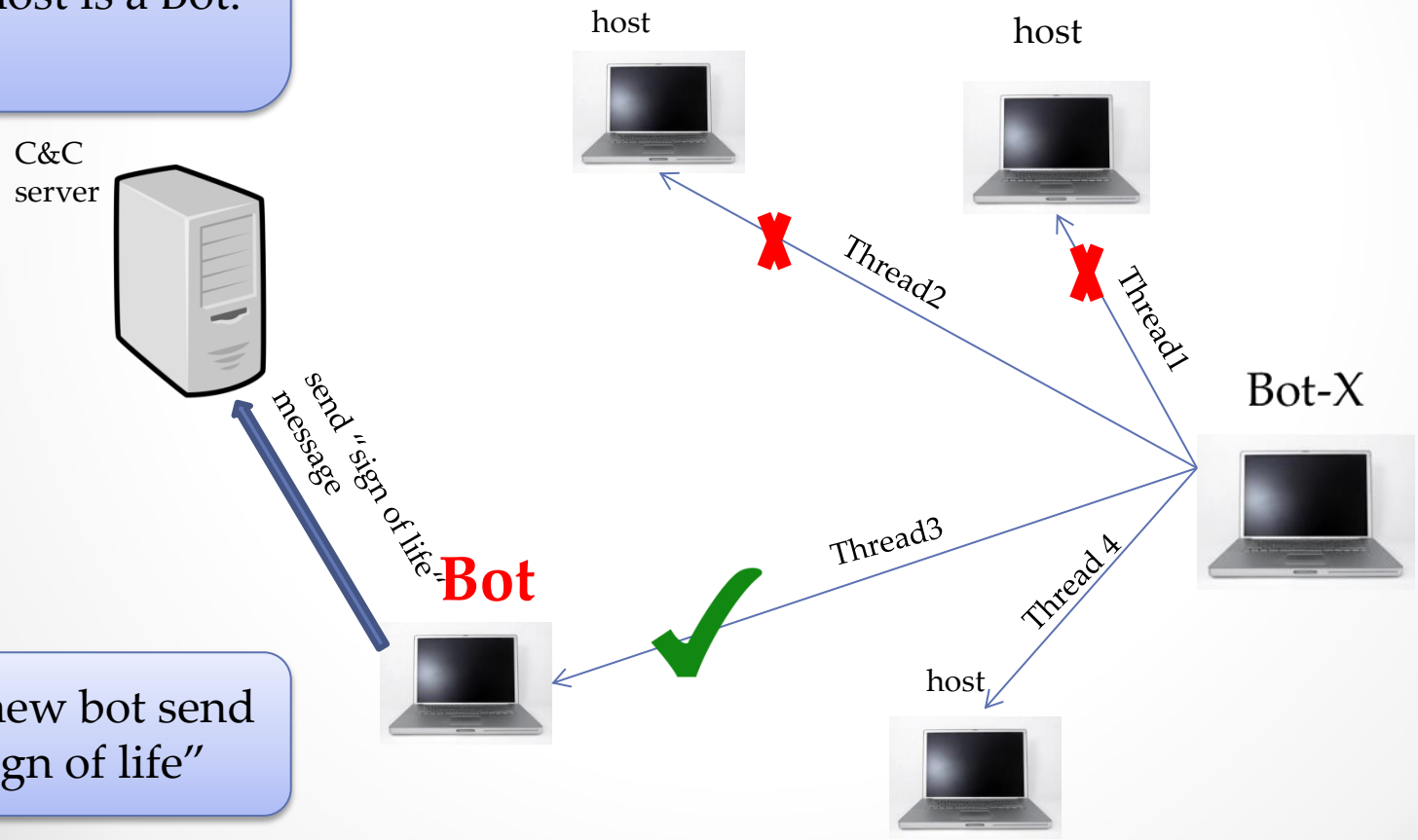
Scanner execution flow

The victim host install the Bot files.



Scanner execution flow

After installing the host is a Bot.



The new bot send "Sign of life"

The server

- The server is written in Flask web framework(Python)
- The server receives http requests (Sign of life) from the bots and sends back a list of tasks to be executed.
- The server stores the bot information in the database and inform the client side application that a bot sent sign of life using WebSocket.

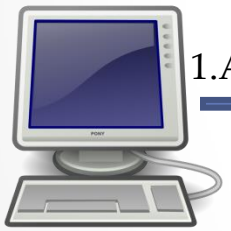
The client side application

- The client side application written in AngularJS with SocketIO.
- Its purpose is creating easy to use graphical user interface for the botnet maintainer to control the bot network.
- The bot master can see the list of the currently online bots, a list of offline bots, and a list of task results.
- Bot master can send tasks to the bots that needs to be executed, for example run a shell command on a bot.
- The client side application uses WebSocket for communication with the server in order to provide real time messaging.

Metanet Flow

2.When Attacker send a Task to bot, the server save the request and wait to bot-X to send him "Sign of life" message.

C&C Interface



1.Attacker send Task to bot X

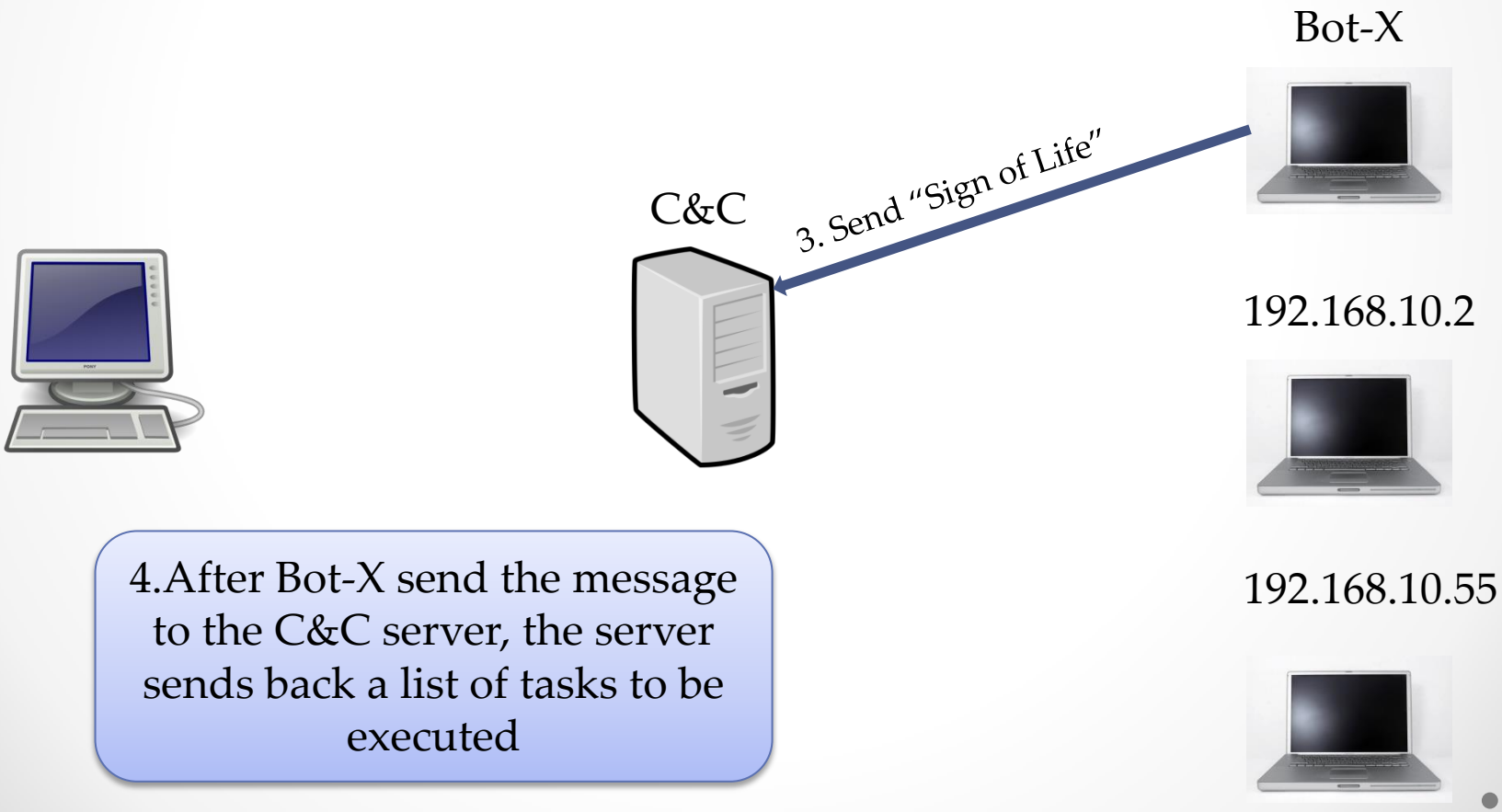
C&C server

Bot-X

192.168.10.2

192.168.10.55

Metanet Flow



Metanet Flow

C&C

5.C&C responds with Task to needs to be executed

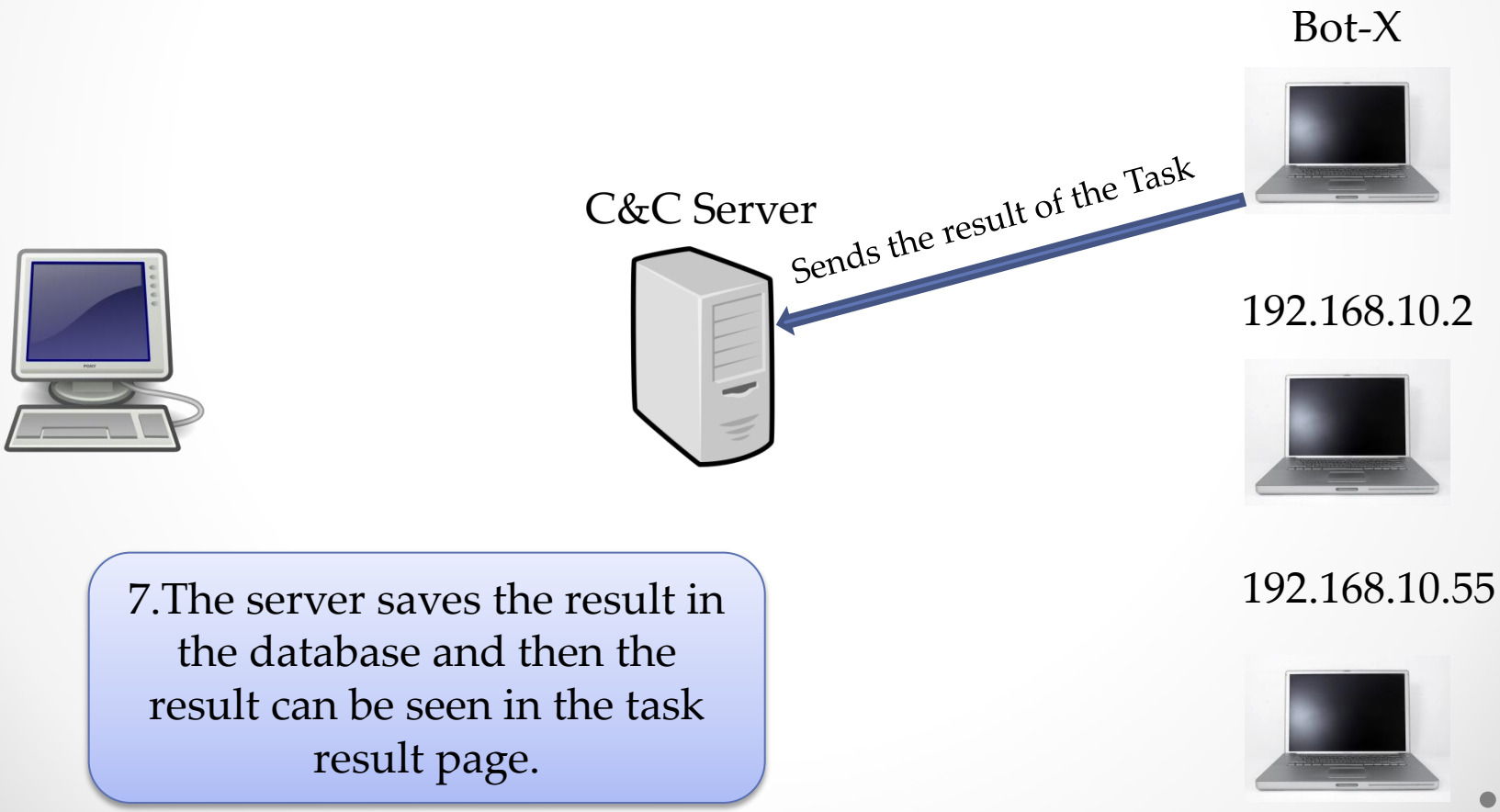
Bot-X

192.168.10.2

192.168.10.55

6.The bot execute the task and sends the result of the task back to the server

Metanet Flow



7.The server saves the result in the database and then the result can be seen in the task result page.