Binary Analysis Next Generation (BANG)

A framework for Binary Analysis

20229531 - Farhan Toshi Hermawan
20319034 - Kalila Ayesha Ismail
21229516 - Bisma Divyananda Erlangga
21229529 - Fariha Qorinatuz Zahra
21229539 - Bestha Hemanthini Hrushitha

Bang! bang! bang!
Ppangya, ppangya, ppangya
Bang! bang! Bang!
BIGBANG - 뱅뱅뱅

What is it?

BANG or Binary Analysis Next Generation is a framework tool that's used to process binary files. It can be used to check whether or not a file has any changes done to it or any malicious coding embedded in it.

Why BANG?

BANG is one of the few open source tools that focus on firmware reverse engineering and security.

History

This tool is relatively new, being only available from end of 2018. It is an update from the former Binary Analysis Tool (BAT) developed by the same author.

Purpose

Unpack files to later classify them, this includes data types, and functionality. It can be used for further security analysis; any malicious codes, vulnerability, or suspicious structure.

Framework

BANG is divided into three parts.

1. An unpacking program that unpacks files.
2. Scanning and identification of the unpacked files.
3. Tools to create data sources that are used by analysis programs.

The unpacking program has two things. First is a scan queue where threads pick and choose their tasks. And the second one is a set of parsers from multiple file formats that can examine formats and take out content.

How it works

Scanning:

1. The scanner is a set of metadata which includes a reference to a file.
2. This file is then parsed by one or more parsers.

- Which parses are run is all based on features known as signatures (headers that file formats tend to have) or known extensions (in case there isn't a viable signature).

- If any files are extracted during the process, a new scanning task is created for each unpacked file and put into a queue to be scanned.

- The metadata that came from each scanning process is separated and then stored in a Python pickle. The metadata here usually includes names, hashes, graphics metadata, etc.

Unpacking:

When BANG analyzes a file, they will try to see:

1. Whether or not it could be processed
2. Whether or not it's a padding file
3. Whether it could checked by the extension
4. Whether it could recognize specific signatures

The check function is used to create meta directories where they store information about the files. When a file is an archive, the "unpack" method on the "UnpackParser" will yield all the unpacked files which are then queued up to be scanned.

When BANG **scans a file for processing**, this is what it does:

1. Checks to see if it's a regular file or a special file (only regular files are scanned)
2. Analyze the file to see what kind it is and if data can be extracted from it
3. Compute checksums (MD5, SHA1, SHA256, optionally TLSH for certain files and telfhash for ELF files)

Meta Directories:

As stated before, metadata is data that was extracted from scanned files. All of this data is then stored in a meta directory.

The top meta directory is the “root”. The meta directory does not have the actual file inside though. Instead, it will have the pathname so it could be referred to the file. The meta directory’s “info.pkl” file contains data structure that maps and connects extracted and unpacked paths to other meta directories.

Who Uses It...

- **Security Analysts:** It is used by security analysts to decrypt firmware and determine if there are any security vulnerabilities. Using the unpacker process, they can find any hidden malicious code or backdoors with firmware.
- **Reverse engineers:** By breaking down complex firmwares into smaller, more manageable components, BANG helps to understand how different parts interact and what purpose they serve. It can be used for debugging, compatibility analysis, or even developing workarounds.
- **Security Researchers:** Researchers use this tool in malware analysis to understand the functioning of malwars and potential resolutions.

Pros & Cons

Pros

- **Recursive Unpacking** ability to monitor even the most nested files
- **Versatility** supports analysis of a wide array of file types
- **Detailed Output** useful for further analysis or automated processing in a JSON file
- **Open-Source**

Cons

- **Complexity in Setup and Use** has its own command-line interface or the dependencies required
- **Limited GUI** not approachable for beginners or less technical users
- **No Recent Updates** lack of ongoing development or support compared

How to use BANG to analyze samples

```
ftoshi@kali: ~/binaryanalysis-ng
File  Actions  Edit  View  Help

┌──(ftoshi㉿kali)-[~]
└─$ cd binaryanalysis-ng

┌──(ftoshi㉿kali)-[~/binaryanalysis-ng]
└─$ nix-shell

[nix-shell:~/binaryanalysis-ng]$ cd src
```

1. Go to directory of BANG

2. Unpack file

```
[nix-shell:~/binaryanalysis-ng/src]$ python3 -m bang.cli scan -u /home/ftos
hi/bang_unpackers/output/jpg/test /home/ftoshi/bang_unpackers/jpg/test.jpg
```

File to scan

(save into written directory)

```
[nix-shell:~/binaryanalysis-ng/src]$ python3 -m bang.cli show /home/ftoshi/
bang_unpackers/output/jpg/test/root
```

3. Show result

Both files are .jpg

ftoshi@kali: ~/binaryanalysis-ng

File Actions Edit View Help

```
[nix-shell:~/binaryanalysis-ng/src]$ python3 -m bang.cli show /home/ftoshi/
bang_unpackers/output/jpg/test/root
```

Parser data

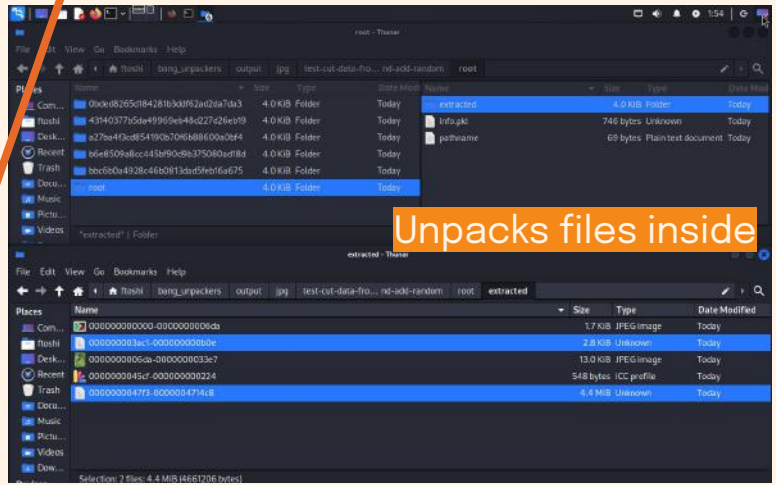
Meta directory	root
Original file	/home/ftoshi/bang_unpackers/jpg/test.jpg
Parser	jpeg
Labels	jpeg, graphics
Parsed size	4676767
sha256	3fe22c38d9a1756c498880188231b8dbd275d05bbe7e38ebcbb71…
md5	c769ae41373e5257300d1c8440670dfa
sha1	6eb3d08b433def52c0b6ad6e14172c0670ad7cd1

```
[nix-shell:~/binaryanalysis-ng/src]$ python3 -m bang.cli show /home/ftoshi/
bang_unpackers/output/jpg/test-add-random-data/root
```

Parser data

[illegible]	root
Original file	/home/ftoshi/bang_unpackers/jpg/test-add-random-data.…
Parser	None
Labels	
sha256	4bb463f8c018e63b3b0e5f10c232d7a65920a18f99e597ed7806f…
md5	cf3b3bb62013f53ccd8af70956651773
sha1	06e07c95c1bf818f9d4af1ed669b00175bb9dd92
tlsh	T13E26339F70EB4403B780CAF138421B8D9FA64731F4DFCE04595…

Labels and Parser does not show .jpg characteristics, malicious?



Unpacks files inside

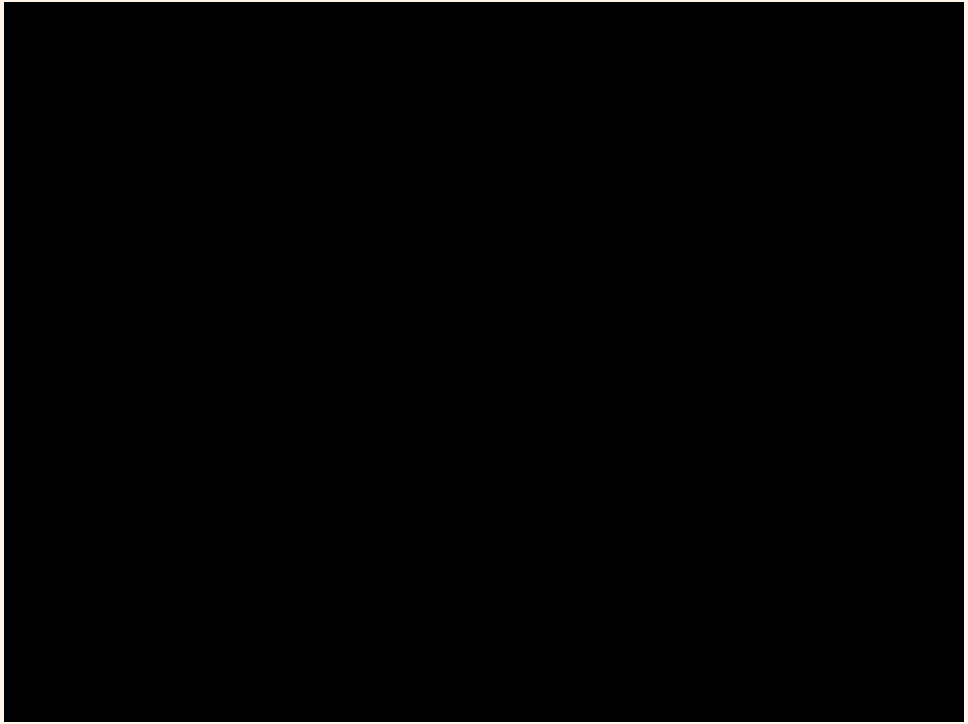
```
[nix-shell:~/binaryanalysis-ng/src]$ exit
exit

┌──(ftoshi㉿kali)-[~/binaryanalysis-ng]
└─$ cd ~
```

4. Exit shell

LIVE DEMO

[Optional] Video Demonstration



References

- Armijnhemel. (n.d.). *Armijnhemel/binaryanalysis-NG: Binary Analysis Next Generation (BANG)*. GitHub. https://github.com/armijnhemel/binaryanalysis-ng
- *Binary analysis next generation review (framework for binary analysis)*. Linux Security Expert. (n.d.-b) https://linuxsecurity.expert/tools/binary-analysis-next-generation/

Thank you!

Any questions?