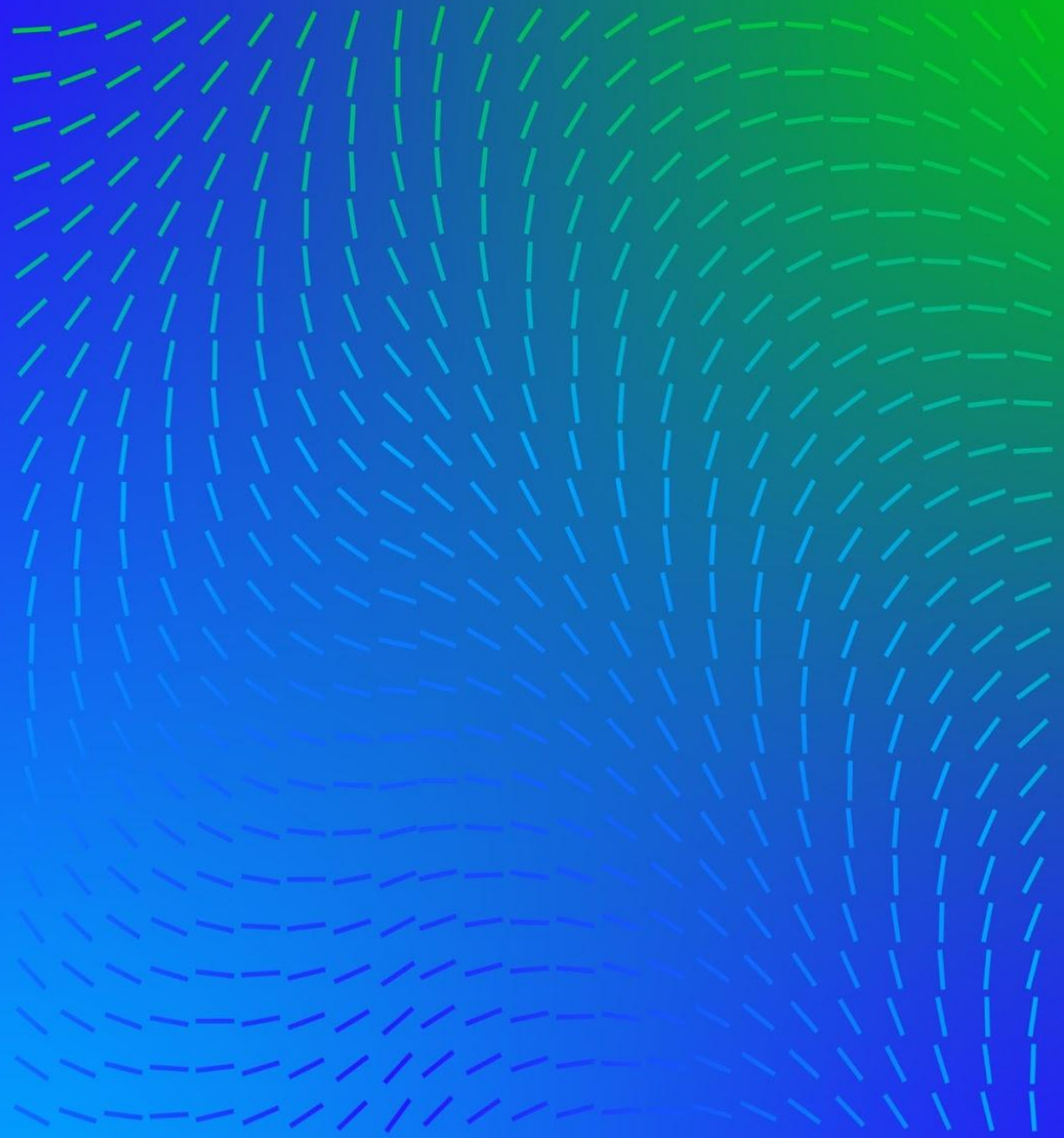
Trellix

Ghidra Analysis & Automation Masterclass

Botconf Angers

Max Kersten
Senior Malware Analyst



Download files

oSamples: https://drop.pm/5
- Password: ghidra

oGhidra script template: https://drop.pm/ad (rename to AmadeyWorkshopTemplate.java)

oBSim database:
https://github.com/advanced-threat-research/BSim/tree/main/golang/windows/database
- Download bsim.golang-runtimes.windows.386-amd64.h2.medium-nosize.mv.7z.001 through 006

oFIDB database:
https://github.com/advanced-threat-research/FIDBs/blob/main/golang/windows/amd64/Trellix.ARC's.Library_golang-1.2.2-through-1.21.6-os-windows_x86.LE.64.default.fidb

Trellix

Table of contents

Trellix

About me

- **Max ‘Libra’ Kersten**
  - /in/Libranalysis on LinkedIn
  - @maxkersten.nl on BlueSky
- **Senior malware analyst at Trellix’ Advanced Research Center**
- **I write blogs about reverse engineering**

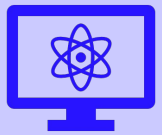
Trellix

Who are you

A brief introduction round

Trellix

About the workshop



Aims to teach concepts

Re-usable concepts in other tools

Ghidra

Freely available and easy-to-use

Focus on the analyst's mindset

Avoid rabbit holes

Trellix

Virtual safety

Virtual machines

Snapshots

Old, but not defunct, samples

Trellix

The dragon

Modular framework

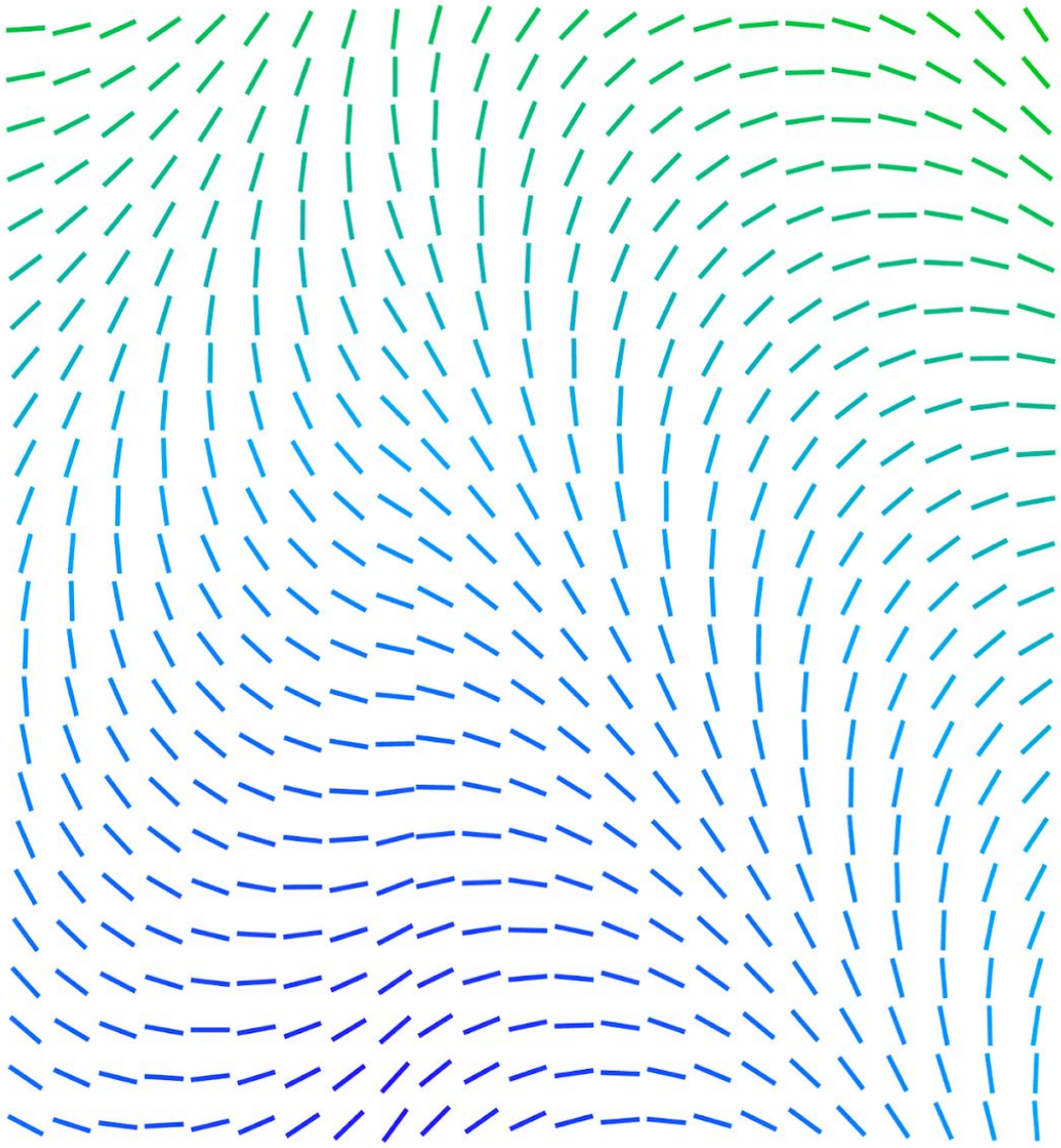
Extension can be written in Java & Python

Project based

Collaboration is made easy

Universal language: p-code

Trellix



Demo

Let’s dive in!

Trellix

A dragon's habits

Views

Data types

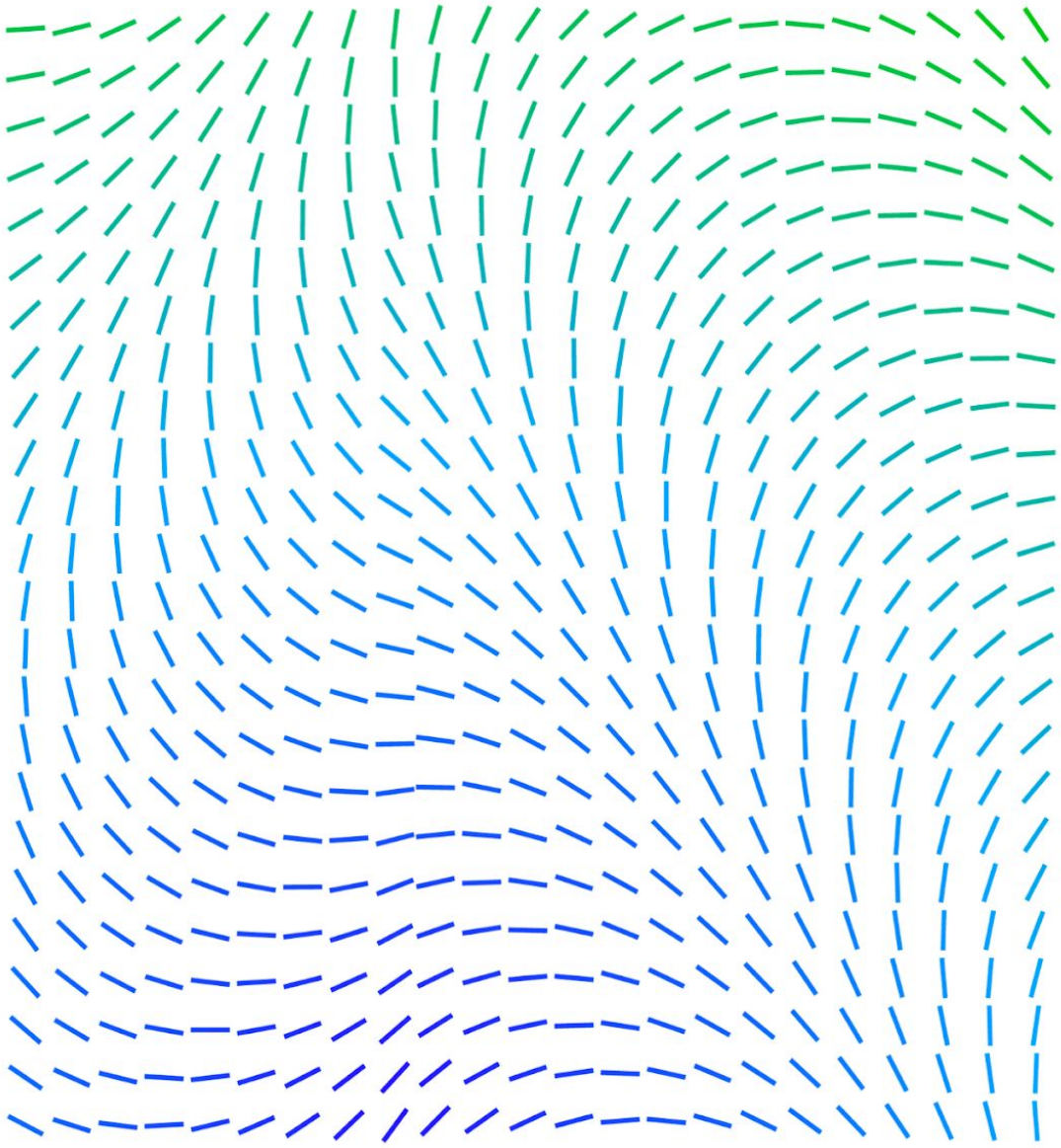
Functions

Hotkeys

Scripting

Headless execution

Trellix



Demo

Let’s dive in!

Trellix

Scripting

The FlatAPI

Ghidra Script State

All scripts, when run, will be handed the current state in the form of class instance variable. These variables are:

1. `currentProgram`: the active program
2. `currentAddress`: the address of the current cursor location in the tool
3. `currentLocation`: the program location of the current cursor location in the tool, or null if no program location exists
4. `currentSelection`: the current selection in the tool, or null if no selection exists
5. `currentHighlight`: the current highlight in the tool, or null if no highlight exists

Headless execution

```
analyzeHeadless <project_location> <project_name>[/<folder_path>] | ghidra://<server>[:<port>]/<repository_name>[/<folder_path>]
    [[-import [<directory>|<file>]+] | [-process [<project_file>]]]
    [-preScript <ScriptName> [<arg>]*]
    [-postScript <ScriptName> [<arg>]*]
    [-scriptPath "<path1>[;<path2>...]"]
    [-propertiesPath "<path1>[;<path2>...]"]
    [-scriptlog <path to script log file>]
    [-log <path to log file>]
    [-overwrite]
    [-recursive]
    [-readOnly]
    [-deleteProject]
    [-noanalysis]
    [-processor <languageID>]
    [-cspec <compilerSpecID>]
    [-analysisTimeoutPerFile <timeout in seconds>]
    [-keystore <KeystorePath>]
    [-connect [<userID>]]
    [-p]
    [-commit ["<comment>"]]
    [-okToDelete]
    [-max-cpu <max cpu cores to use>]
    [-loader <desired loader name>]
```

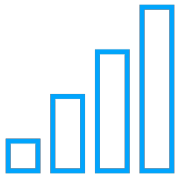
Trellix

The samples

An overview

Amadey

Understanding Ghidra



XorDDoS

String decryption

qBit

Function recovery

Trellix

XorDDoS bot

String decryption

Find the strings

Where are the encrypted strings loaded?

Decrypt a string

Recreate the string decryption routine in a language of choice

Decrypt all strings

What is the structure in which some strings are found?

How would you decrypt them all?

Trellix

```
        MOV     dword ptr [EBP + local_3c],0x0
        JMP     LAB_0804d12e
LAB_0804d108
        MOV     EDX,dword ptr [EBP + local_3c]
        MOV     EAX,EDX
        SHL     EAX,0x2
        ADD     EAX,EDX
        SHL     EAX,0x2
        ADD     EAX,daemonname
        MOV     dword ptr [ESP + local_3dec],0x14
        MOV     dword ptr [ESP]=>local_3df0,EAX
        CALL    encrypt_code
        ADD     dword ptr [EBP + local_3c],0x1
LAB_0804d12e
        CMP     dword ptr [EBP + local_3c],0x16
        JBE     LAB_0804d108
```

```
        MOV     dword ptr [EBP + i],0x0                 ;sets the counter to 0
        JMP     loop_compare                            ;jumps to the loop comparison
loop_body
        MOV     EDX,dword ptr [EBP + i]                 ;stores the counter in EDX
        MOV     EAX,EDX                                 ;stores the counter in EAX
        SHL     EAX,0x2                                 ;shift left by two, equals times 4 (2^2)
        ADD     EAX,EDX                                 ;adds the multiplied value to the copy of the counter
        SHL     EAX,0x2                                 ;shift left by two again
        ADD     EAX,daemonname                          ;add the start address of the array
        MOV     dword ptr [ESP + local_3dec],0x14       ;push 0x14 on the stack
        MOV     dword ptr [ESP]=>local_3df0,EAX         ;push the array (plus offset) to the stack
        CALL    encrypt_code                            ;call the decryption function
        ADD     dword ptr [EBP + i],0x1                 ;increment the counter by one
loop_compare
        CMP     dword ptr [EBP + i],0x16                ;compares the counter to 0x16
        JBE     loop_body                               ;jump if below or equal
```

Trellix

Multiplication is repeated addition

```
for (int i = 0; i <= 0x16; i++)
{
        int result = i;
        result = result * 4;
        result += i;
        result = result * 4;
}
```

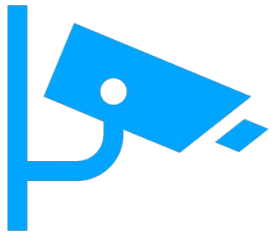
```
for (int i = 0; i <= 0x16; i++)
{
        int result = ((i * 4) + i) * 4;
        System.out.println(result);
}
```

```
((i * 4) + i) * 4;
(i * 5) * 4
i * 20
```

Trellix

Amadey

Understanding Ghidra

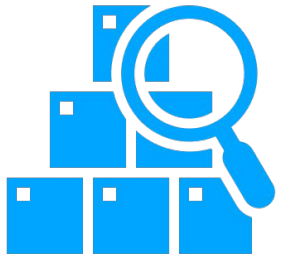


Behaviour

How does it escalate its privilege?

How does it persist?

(Bonus) How does it handle C2 communication?



Strings

How are the strings decrypted?

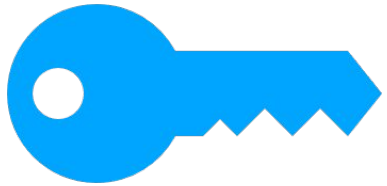
Which AVs are checked for?

Work with the template script to automate string decryption

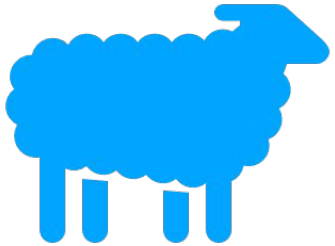
Trellix

Hashes

Different types for different purposes



Cryptographic



Fuzzy

Trellix

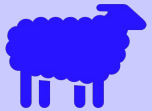
Function recovery

Default naming scheme

Address based

FunctionID

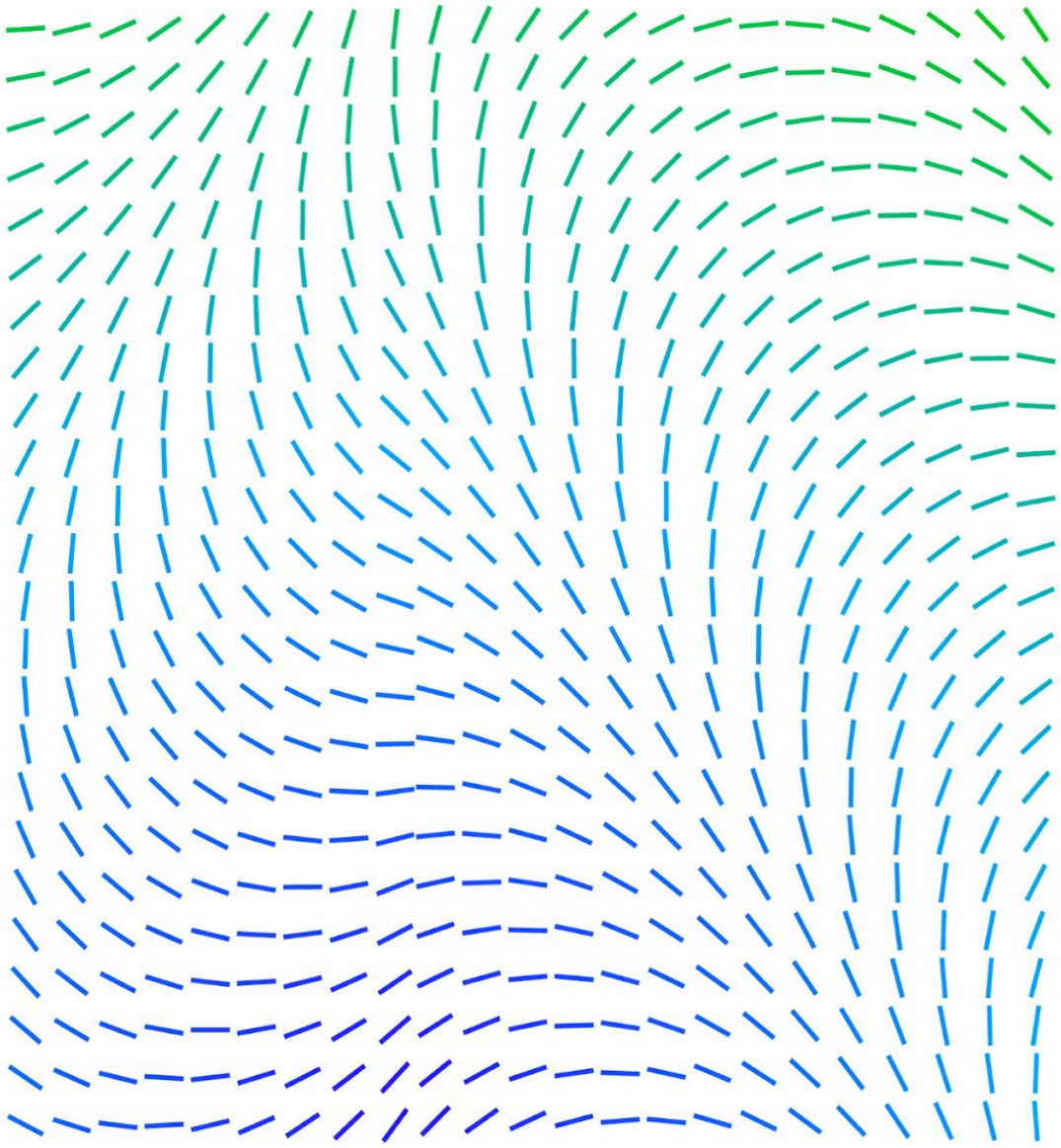
Exact matches



BSim

Fuzzy matches

Trellix



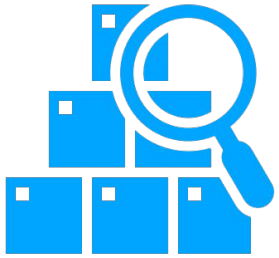
Demo

Let’s dive in!

Trellix

qBit

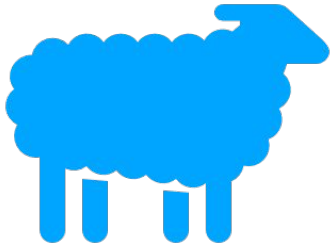
Applied function name recovery



FunctionID

Configure Ghidra to use the additional FIDB

Apply the signatures to the qBit sample

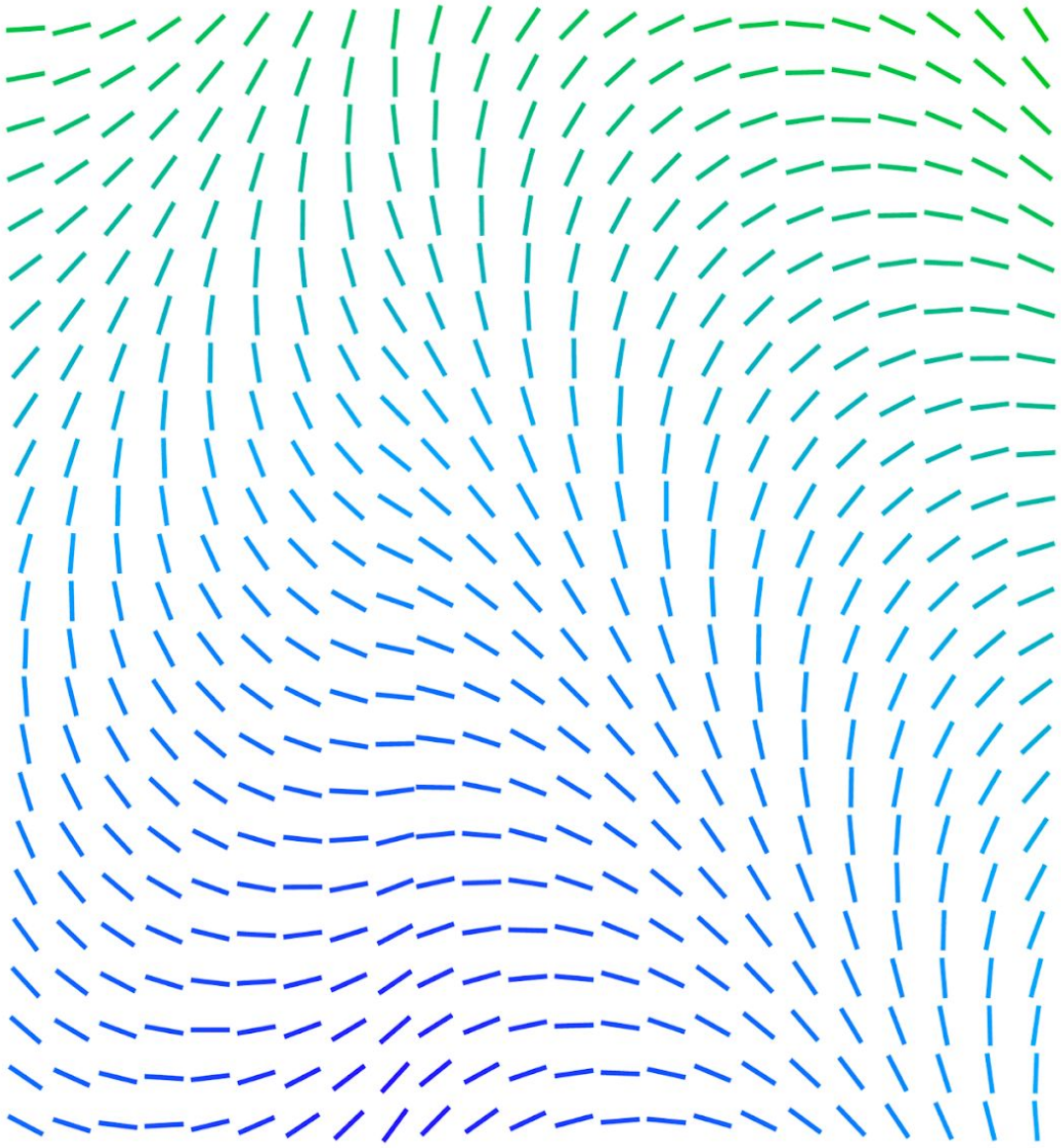


BSim

Run the rename script

Run the rename script headlessly

Trellix



Q&A

For questions, you can also reach out to me via /in/Libranalysis aka Max Kersten on LinkedIn

Trellix