Improving Trace Synthesis

by Utilizing Computer Vision for User Action Emulation

Lukas Schmidt, University of Münster

Introduction: Why Are Datasets Needed in Digital Forensics ?

Digital Forensics entails the analysis of extensive volumes of unstructured data, originating from diverse sources.

Therefore training datasets are needed to:

- Teach investigators
- Validate forensic tools
- Advance algorithms & machine learning models
- Pursue research

Digital Forensics Is In Demand of **Realistic** Datasets

- Better transferability of research results, applicability in practical settings

- Advent of Machine Learning & Artificial Intelligence amplifies demand: **reliability of pre-trained models depends on the quality of the training datasets,** deciding over the usefulness in real world scenarios

- Use of realistic training data correlates with quality of research outcomes

Obtaining Realistic Datasets is a Problem

- Can’t use real evidence, mainly due to ethical and legal reasons

- Sharing datasets is hampered by **demands on privacy protection** or the **threat of possible copyright infringements**

- Therefore the Forensic community faces a shortcoming of realistic datasets: **the dataset gap problem**

The Dataset Gap Problem

- Negatively impacts research and practice
- Inhibits reproducibility of results
- Researchers waste valuable resources in obtaining custom datasets

Solution: Synthetic Datasets

- Unproblematic replications of realistic evidence

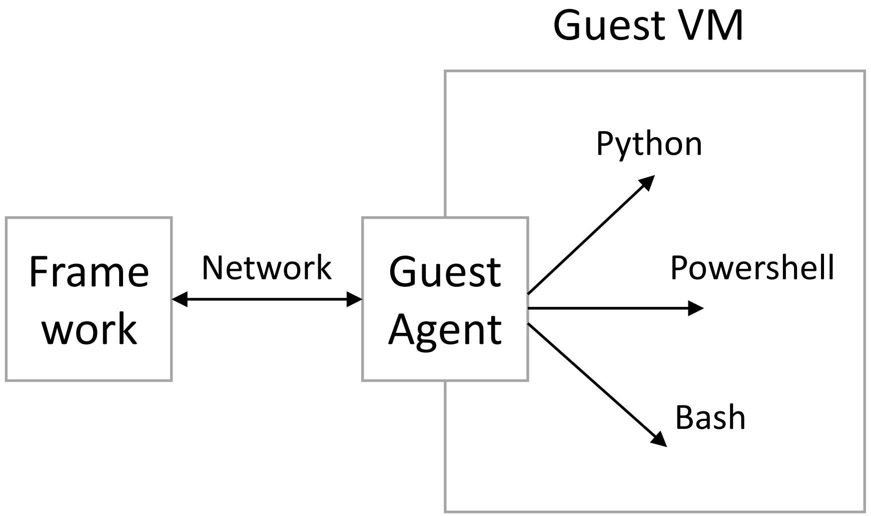
- **How ?** By populating disk images with traces of emulated user behaviour

- Manual Synthesis – time and resource intensive, doesn‘t scale, careful execution and planning needed

- Automated Synthesis - several frameworks were introduced, aiming to ease and scale dataset creation

Automated Trace Synthesis

- Trace creation by replaying user actions in Virtual Machines (guest VMs)
- Control instances (guest agents) run in guest VMs, receiving commands via network
- Guest agents execute emulated user actions in VMs



Drawbacks of Existing Solutions

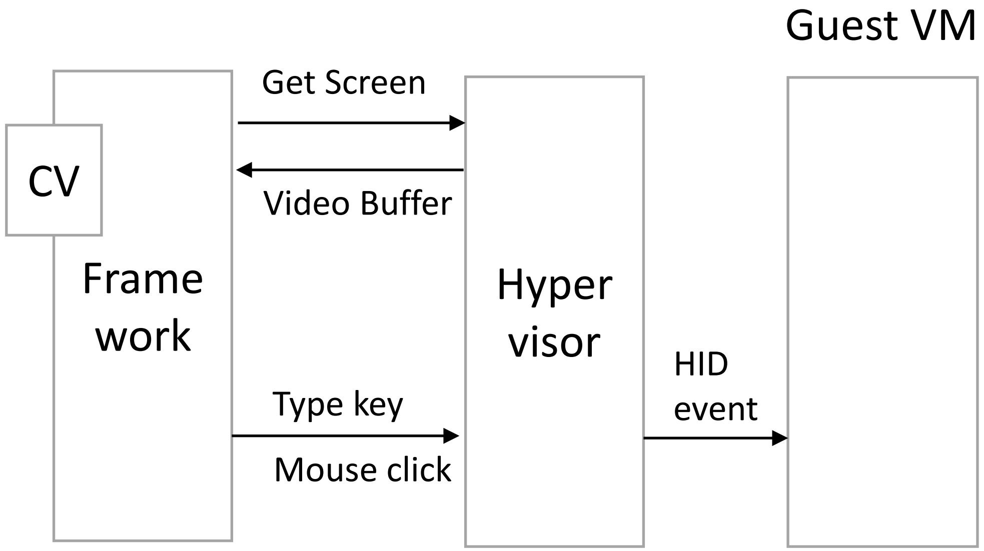
- Trace pollution – emulation techniques cause side effects, leading to unwanted traces:
  - Network connections to guest agents
  - Program / script execution
  - Software artifacts

- Reduced usefulness of synthesized datasets, especially when considering Machine Learning and Artificial Intelligence.

- Missing GUI automation capabilities

A Novel Approach – VM-external GUI Automation

- **Aim:** More realistic user action emulation leading to more realistic traces
- **How ?** Combine a hypervisor and computer vision algorithms
- Computer Vision identifies GUI-elements in guest VM desktop (x,y coordinates)
- Hypervisor creates USB-HID events, not distinguishable from human-generated input for guest VM

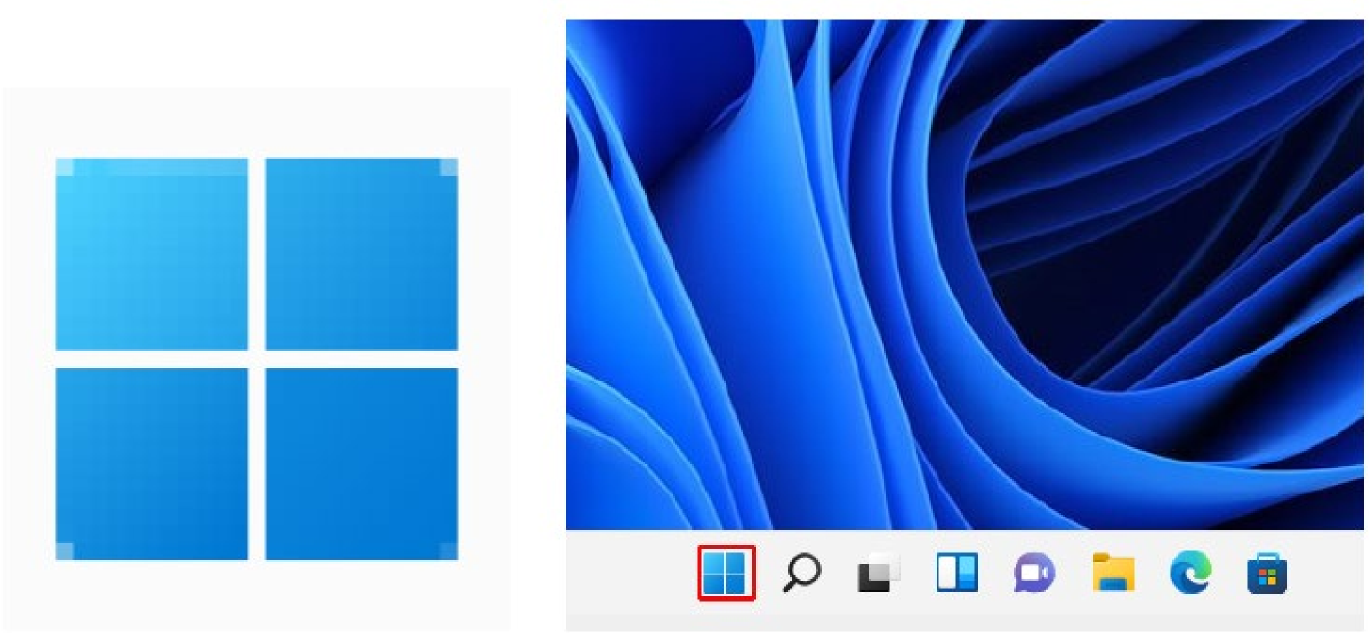


Step 1: GUI-element identification

- Provide a template of the GUI-element

- Template matching: use Computer Vision to match a template in the current desktop

- This way obtaining x,y coordinates to work with



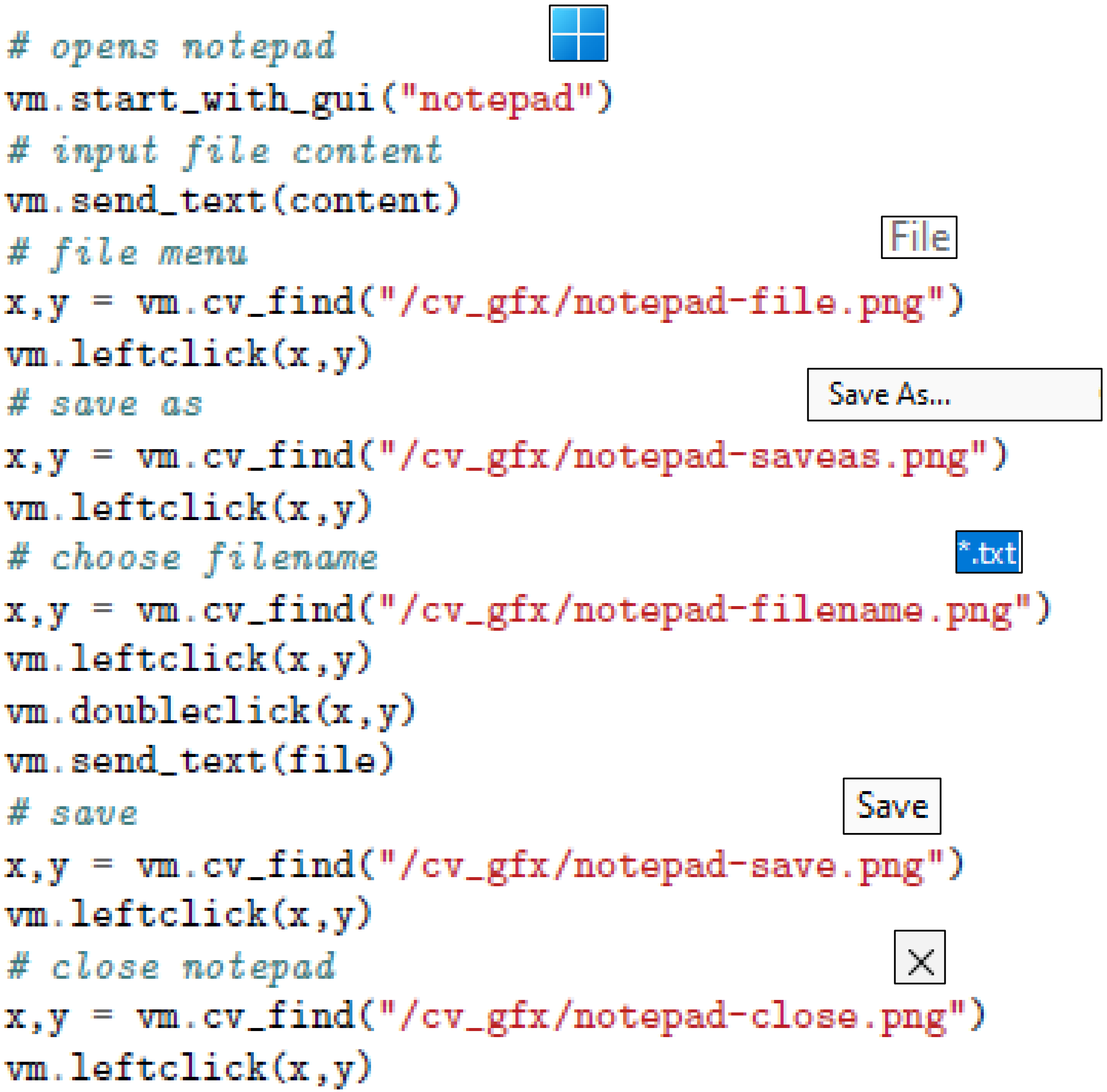
(a) Home Button template.

(b) Matched template in screenshot.

Step 2: React with USB-HID events

Example: Create a file with Notepad

- Start Notepad using the Windows home button
- Insert some text into Notepad
- Save file using the save dialog
- We provide an open source framework ☺



```
# opens notepad
vm.start_with_gui("notepad")
# input file content
vm.send_text(content)
# file menu
x,y = vm.cv_find("/cv_gfx/notepad-file.png")
vm.leftclick(x,y)
# save as
x,y = vm.cv_find("/cv_gfx/notepad-saveas.png")
vm.leftclick(x,y)
# choose filename
x,y = vm.cv_find("/cv_gfx/notepad-filename.png")
vm.leftclick(x,y)
vm.doubleclick(x,y)
vm.send_text(file)
# save
x,y = vm.cv_find("/cv_gfx/notepad-save.png")
vm.leftclick(x,y)
# close notepad
x,y = vm.cv_find("/cv_gfx/notepad-close.png")
vm.leftclick(x,y)
```

How to Evaluate our Approach ?

- Create a simple imaginary scenario
  - User visiting websites, downloading files, executing sqlmap, …

- Emulate this scenario on Windows 11 using the most popular emulation techniques: Python & Powershell, and our Computer Vision approach (3 synthetic disk images)

- Extract & compare unique traces of each approach's disk image with Plaso and Pandas (differential analysis)
  - Registry, Sqlite files, Link files, OLECF files, Event Log, Prefetch files

Performing the Differential Analysis

Why ? Compute the feature delta of traces sets, so we can evaluate the unique traces of each approach.

- Removal of traces without timestamps in every trace set.
- Removal of duplicate entries equal in timestamp and content for each trace set.
- Generate the union of all trace sets, then remove all duplicate entries in the union (keeping the trace origin for identification).

Results: More and More Diverse Traces Using GUI automation

GUI automation leads to more traces, increase of approximately 20% in total.

Reasons:

- Browser operations result in a greater amount of cached pages and saved cookies
- Windows Timeline artifacts in the registry contain traces of executed programs
- Additional traces of file usage can be found in link files, OLECF files (AutomaticDestinations) and registry keys (MostRecentlyUsed, typed_urls)
- And more traces unparsed by Plaso parsers

Results: Omits Trace Pollution

VM-external GUI-automation omits traces created by other solutions, e.g.:

- Software artifacts (executables, libraries, …)
- Traces of services, network connections or remote logins in the event log
- Traces of program execution in event log, registry or prefetch files

Benefits of VM-external GUI Automation

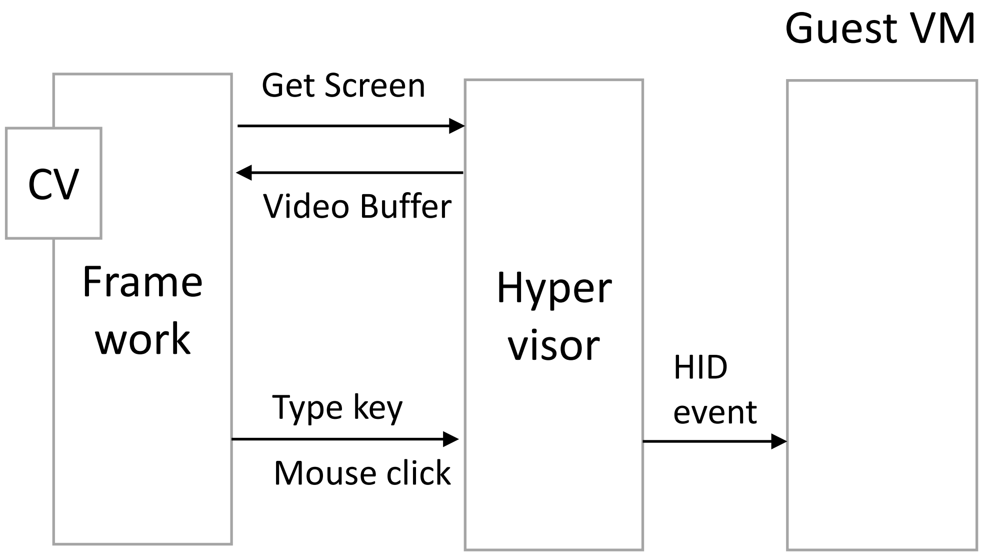
- Independent of guest agents
- OS-agnostic solution, working with every (QEMU-)virtualizable operating system
- No trace pollution (e.g. traces of program execution inside the guest VM, or software artifacts)
- Replay user actions with scriptable user input

Future Work

- Work towards large-scale, automated synthesis of multi-source datasets
- Integration into existing dataset-synthesis frameworks, which can set up infrastructure, plan user actions etc.
- Investigate possible applications in other areas, e.g. network or memory forensics, malware analysis or attack simulations
- More related work on how sociological and criminalistic aspects should mirror in synthetic datasets
  - What does realistic “wear and tear” look like ?
  - What is typical user behavior and device usage ?
  - Can we generate this automatically ?

Thank You ! Questions ?

- Combining Hypervisor & Computer Vision
- VM-external GUI automation
- User Action Emulation


Guest VM
CV
Frame work
Get Screen
Video Buffer
Type key
Mouse click
Hyper visor
HID event