guac-parser

By wit0k

The Proof of Concept (PoC) script enables the parsing of the Guacamole protocol, which includes a stream of recorded session replay. Consequently, providing the capability to extract screenshots, triggered by progress indicators based on the percentage of completion, from the recording progress. Additionally, it includes functionality for replaying the recording within an integrated viewer.

Remark: Python 3.13.1t (Free-threading) is recommended

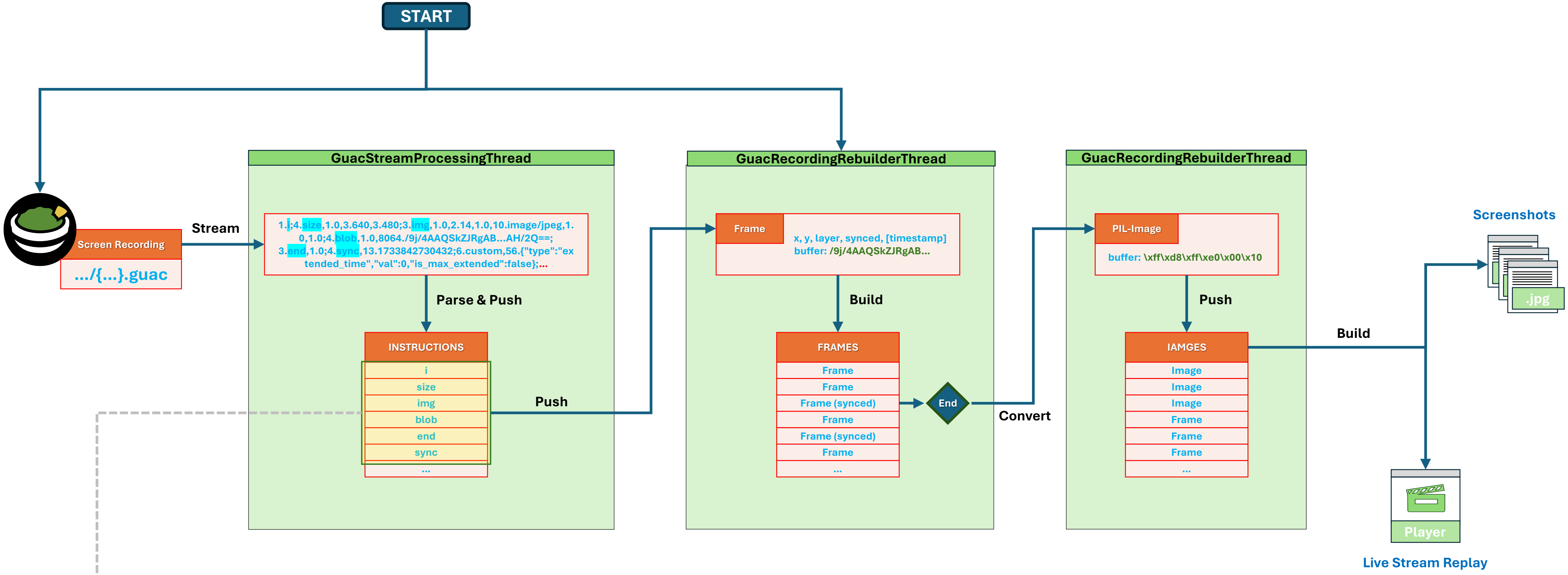
Key Features

- **Speed:** Parses multiple videos in seconds (depends on network stream throughput)
- **Portable**: Extensible design allows for easy addition of new instruction types.
- **Instruction Parsing**: Decodes the Guacamole session recording (guac) instructions stream.
- **Screenshot Dumping**: Extracts screenshots from the session recording stream, capturing the visual behavior at specified progress points.
- **Recording Replay**: Enables the replay of the entire detonation session in a built-in viewer (BETA – currently supports single session only).

To Do

- **Offline Replay**: Add the capability to create MP4 videos from the recording stream for easier sharing, review, or auditing.
- **AI support**: A model to analyze and summarize activities from screen shots

Summary



```
1.i;4.size,1.0,3.640,3.480;3.img,1.0,2.14,1.0,10.image/jpeg,1.
0,1.0;4.blob,1.0,8064./9j/4AAQSkZJRgAB...AH/2Q==;
3.end,1.0;4.sync,13.1733842730432;6.custom,56.{"type":"ex
tended_time","val":0,"is_max_extended":false};...
```

```
x, y, layer, synced, [timestamp]
buffer: /9j/4AAQSkZJRgAB...
```

```
buffer: \xff\xd8\xff\xe0\x00\x10
```

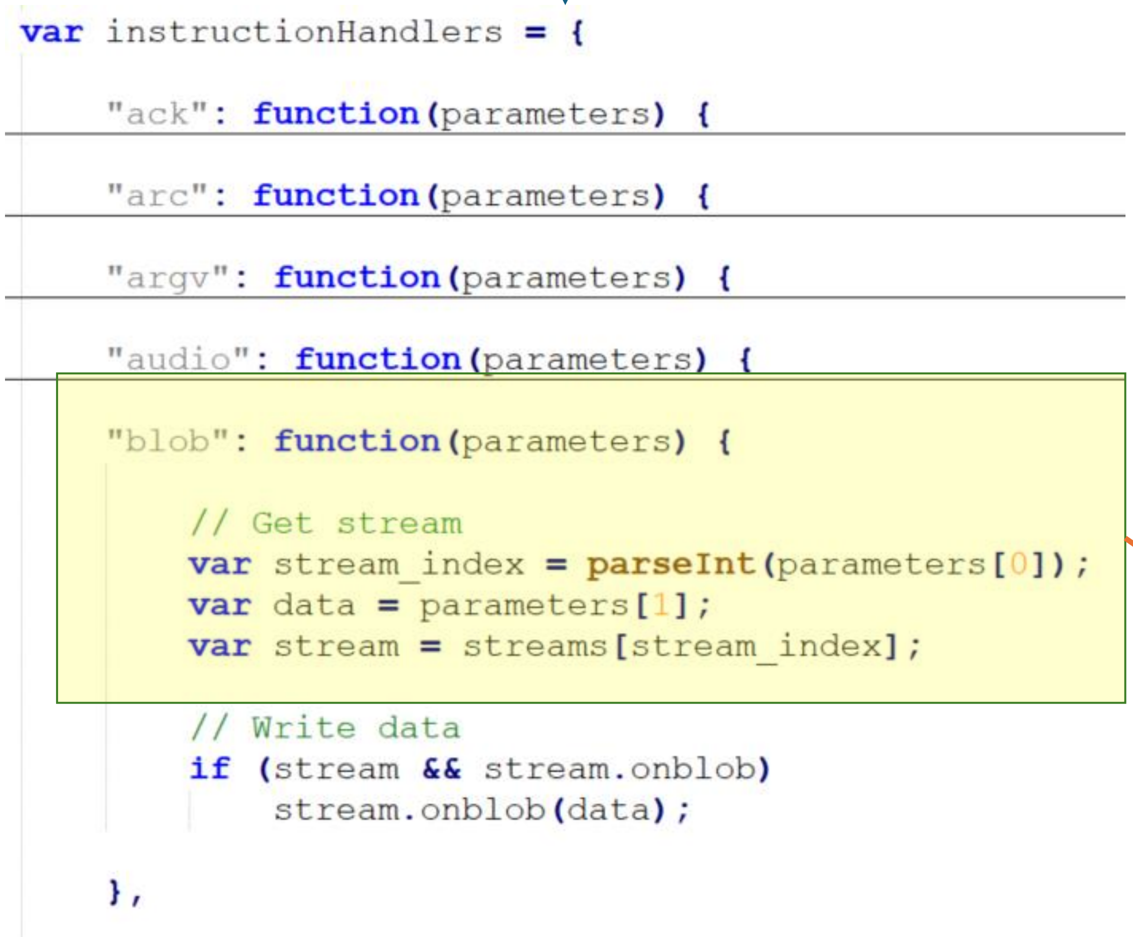
i	size	img	blob	blobN	end	sync	custom
i		img	blob	blobN	end	sync	
		img	blob		end	sync	
		img	blob		end		
						sync	custom

Remarks

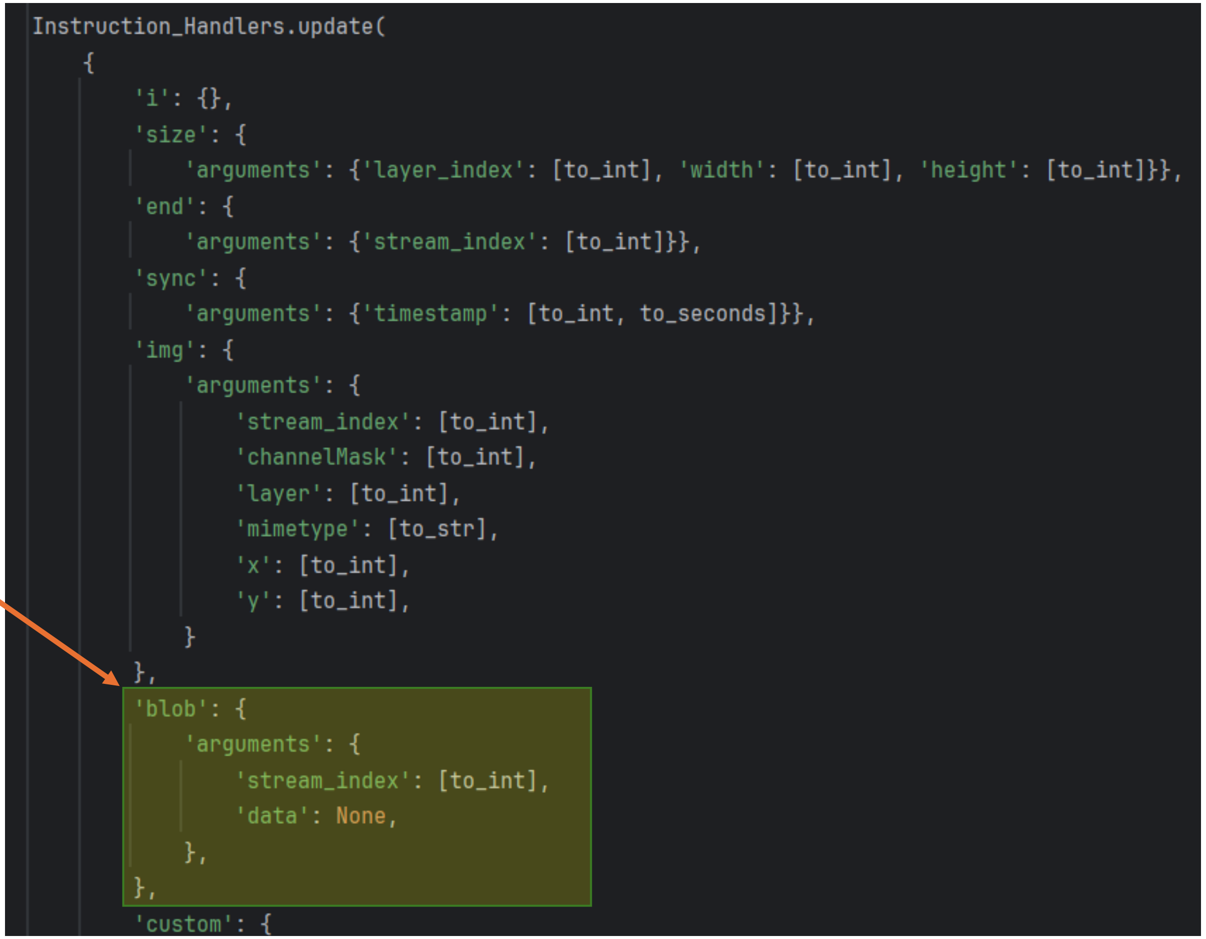
- The instruction blobs within **img** ... **end** must be concatenated
- Refresh the display on frames with **sync** element
- Each instruction **size** forces the resize of **Display**
- **All Layers** are auto-resizing according to **Default Layer (0)**

Flow

guacamole-client-1.5.5.tar\guacamole-client-1.5.5\guacamole-common-js\src\main\webapp\modules\client.js

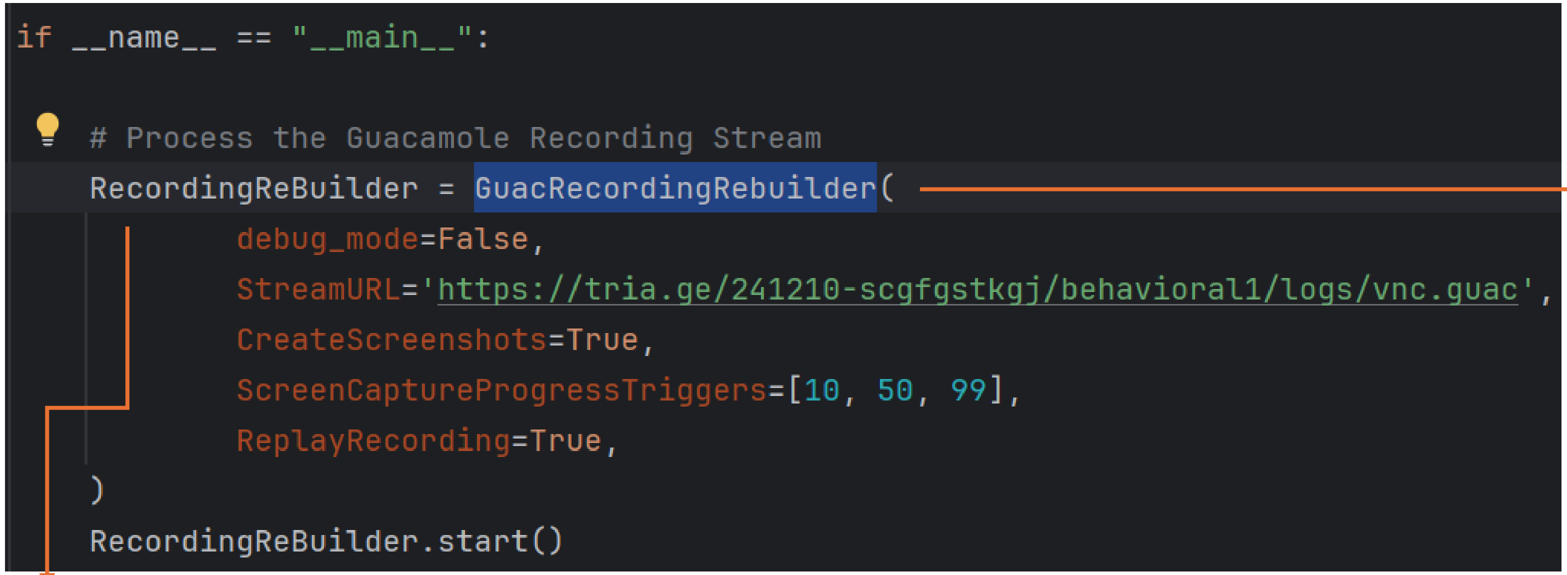


```
var instructionHandlers = {

    "ack": function(parameters) {

    "arc": function(parameters) {

    "argv": function(parameters) {

    "audio": function(parameters) {

    "blob": function(parameters) {

        // Get stream
        var stream_index = parseInt(parameters[0]);
        var data = parameters[1];
        var stream = streams[stream_index];

        // Write data
        if (stream && stream.onblob)
            stream.onblob(data);

    },
```

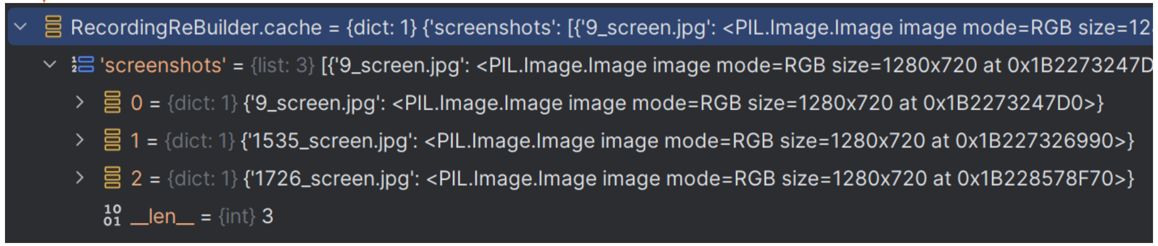


```
Instruction_Handlers.update(
    {
        'i': {},
        'size': {
            'arguments': {'layer_index': [to_int], 'width': [to_int], 'height': [to_int]}},
        'end': {
            'arguments': {'stream_index': [to_int]}},
        'sync': {
            'arguments': {'timestamp': [to_int, to_seconds]}},
        'img': {
            'arguments': {
                'stream_index': [to_int],
                'channelMask': [to_int],
                'layer': [to_int],
                'mimetype': [to_str],
                'x': [to_int],
                'y': [to_int],
            }
        },
        'blob': {
            'arguments': {
                'stream_index': [to_int],
                'data': None,
            },
        },
        'custom': {
```

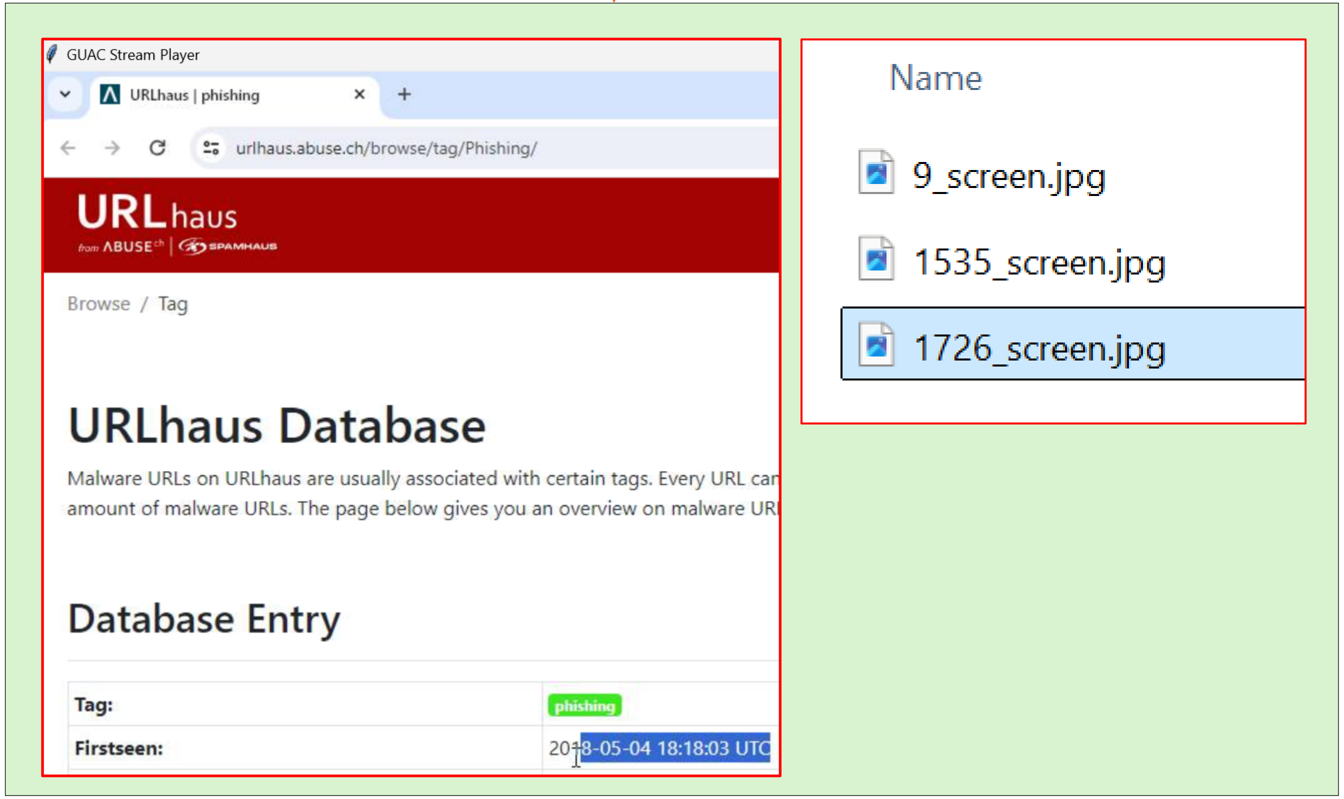
Parser



```python
if __name__ == "__main__":

    # Process the Guacamole Recording Stream
    RecordingReBuilder = GuacRecordingRebuilder(
            debug_mode=False,
            StreamURL='https://tria.ge/241210-scgfgstkgj/behavioral1/logs/vnc.guac',
            CreateScreenshots=True,
            ScreenCaptureProgressTriggers=[10, 50, 99],
            ReplayRecording=True,
    )
    RecordingReBuilder.start()
```



```
RecordingReBuilder.cache = {dict: 1} {'screenshots': [{'9_screen.jpg': <PIL.Image.Image image mode=RGB size=12
  'screenshots' = {list: 3} [{'9_screen.jpg': <PIL.Image.Image image mode=RGB size=1280x720 at 0x1B2273247D
    0 = {dict: 1} {'9_screen.jpg': <PIL.Image.Image image mode=RGB size=1280x720 at 0x1B2273247D0>}
    1 = {dict: 1} {'1535_screen.jpg': <PIL.Image.Image image mode=RGB size=1280x720 at 0x1B227326990>}
    2 = {dict: 1} {'1726_screen.jpg': <PIL.Image.Image image mode=RGB size=1280x720 at 0x1B228578F70>}
    __len__ = {int} 3
```



URLhaus Database

Malware URLs on URLhaus are usually associated with certain tags. Every URL car
amount of malware URLs. The page below gives you an overview on malware URI

Database Entry

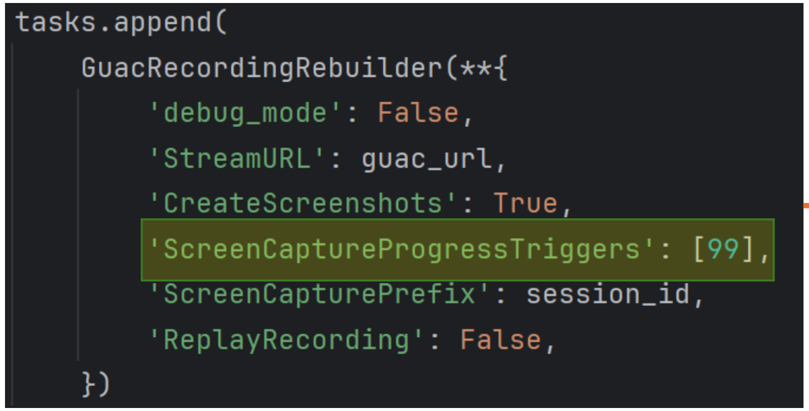
Tag:	phishing
Firstseen:	2018-05-04 18:18:03 UTC

Name

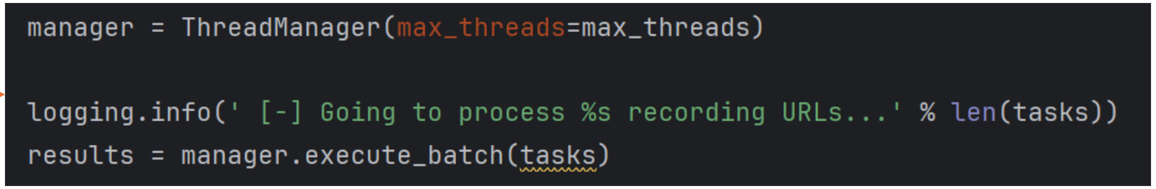
- 9_screen.jpg
- 1535_screen.jpg
- 1726_screen.jpg

Example

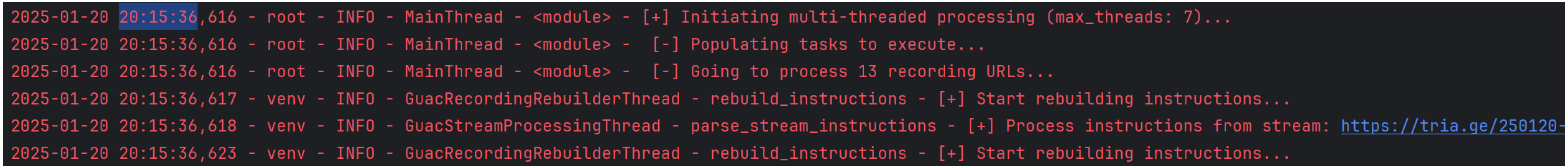
Capturing a screen shot when the replay has reached **99%** for 13 submissions



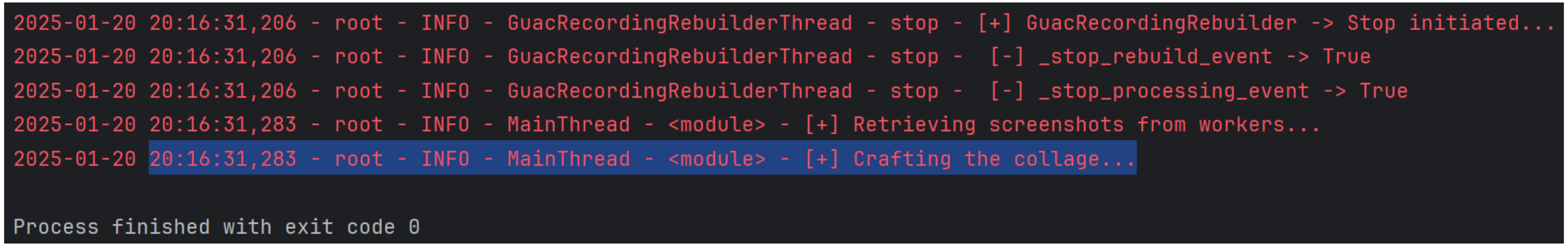
```
tasks.append(
    GuacRecordingRebuilder(**{
        'debug_mode': False,
        'StreamURL': guac_url,
        'CreateScreenshots': True,
        'ScreenCaptureProgressTriggers': [99],
        'ScreenCapturePrefix': session_id,
        'ReplayRecording': False,
    })
```



```
manager = ThreadManager(max_threads=max_threads)

logging.info(' [-] Going to process %s recording URLs...' % len(tasks))
results = manager.execute_batch(tasks)
```



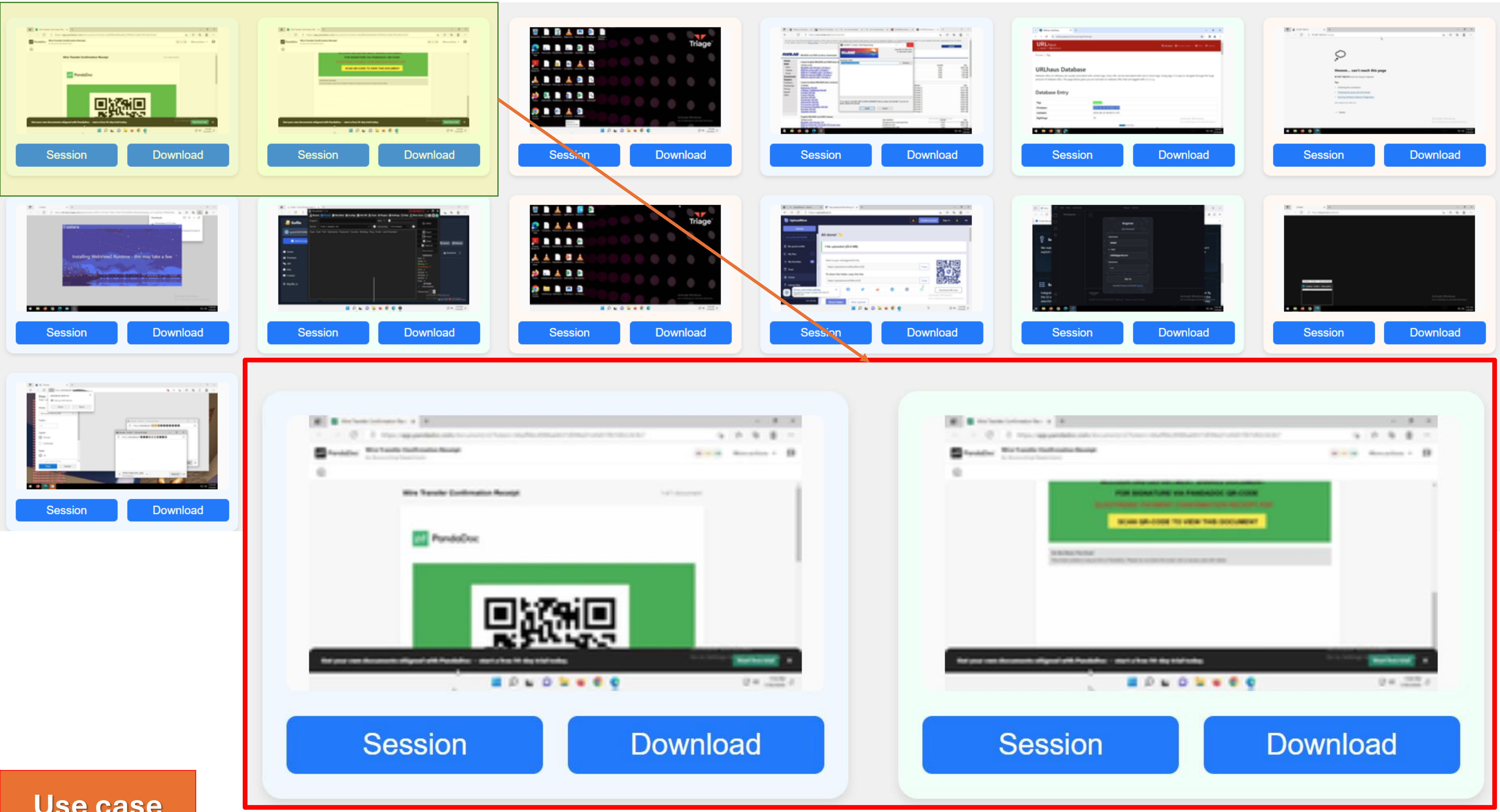
```
2025-01-20 20:15:36,616 - root - INFO - MainThread - <module> - [+] Initiating multi-threaded processing (max_threads: 7)...
2025-01-20 20:15:36,616 - root - INFO - MainThread - <module> -  [-] Populating tasks to execute...
2025-01-20 20:15:36,616 - root - INFO - MainThread - <module> -  [-] Going to process 13 recording URLs...
2025-01-20 20:15:36,617 - venv - INFO - GuacRecordingRebuilderThread - rebuild_instructions - [+] Start rebuilding instructions...
2025-01-20 20:15:36,618 - venv - INFO - GuacStreamProcessingThread - parse_stream_instructions - [+] Process instructions from stream: https://tria.ge/250120-
2025-01-20 20:15:36,623 - venv - INFO - GuacRecordingRebuilderThread - rebuild_instructions - [+] Start rebuilding instructions...
```



```
2025-01-20 20:16:31,206 - root - INFO - GuacRecordingRebuilderThread - stop - [+] GuacRecordingRebuilder -> Stop initiated...
2025-01-20 20:16:31,206 - root - INFO - GuacRecordingRebuilderThread - stop -  [-] _stop_rebuild_event -> True
2025-01-20 20:16:31,206 - root - INFO - GuacRecordingRebuilderThread - stop -  [-] _stop_processing_event -> True
2025-01-20 20:16:31,283 - root - INFO - MainThread - <module> - [+] Retrieving screenshots from workers...
2025-01-20 20:16:31,283 - root - INFO - MainThread - <module> - [+] Crafting the collage...

Process finished with exit code 0
```

Less than 1 min later, 13 submissions have been processed ... **it takes seconds on private subscription**.

Use case


PandaDoc
Triage
URLhaus Database
Database Entry
Installing WebView2 Runtime - this may take a few
Session
Download
Session
Download
Session
Download
Session
Download
Session
Download
Session
Download
Session
Download
Session
Download
Session
Download
Session
Download
Session
Download
Session
Download
Session
Download
Session
Download
Session
Download

Use case