USENIX Security '24 Artifact Appendix: RECORD: A RECeption-Only Region Determination Attack on LEO Satellite Users

Eric Jederman
RPTU Kaiserslautern-Landau, Germany
jedermann@cs.uni-kl.de

Martin Strohmeier
armasuisse, Switzerland
martin.strohmeier@armasuisse.ch

Vincent Lenders
armasuisse, Switzerland
vincent.lenders@armasuisse.ch

Jens Schmitt
RPTU Kaiserslautern-Landau, Germany
jschmitt@cs.uni-kl.de

A Artifact Appendix

A.1 Abstract

In our paper we present a novel passive attack called RECORD, which is solely based on the reception of messages to LEO satellite users on the ground, threatening their location privacy. In particular, we show that by observing only the downlink of 'wandering' communication satellites over wide beams can be exploited at scale from passive attackers situated on Earth to estimate the region in which users are located. Besides real world measurements, we conducted extensive simulative evaluations, to evaluate the impact of important variables as the observation time or the observer placement.

This artifact contains the simulation framework to perform RECORD attack simulations, the satellite antenna beam models for the simulations, the original simulation data behind the paper and the scripts for generating the evaluation graphs.

A.2 Description & Requirements

The submitted artifacts can be distinguished in two parts:

1. The simulation: The required scripts and satellite antenna beam model to execute the simulation of the RECORD attack. The main simulation script is *simulations_attackerTypes_fibo.py*. It is responsible for executing the simulations described in the paper sections 5.2, 5.3 and 5.4. The different scenarios of each paper section can be recreated by adapting a row of parameters at the end of the file.
   The iridium antenna beam models are located in *beamModel_iridium.npy* and *beamModel_iridium_noisy.npy*. The second model is a noised variant of the first, to simulate an inaccurate antenna beam model of the attacker (paper section 5.1.Simulation Setup.*Beam Model*).
2. The data evaluation: The original simulation data and the scripts to evaluate the data. The original simulation data can be found in the subfolder *simulation_data*. Depending on the parameters that are varied, sub-subfolders are used to separate the data collection in useful groups. In each sub-subfolder are multiple files, with the file name describing the individual simulation parameters.
   The python scripts named *graph_*.py* use the generated simulation data to generate the graphs, shown in the paper. An overview which script generates which graph can be found in the *README.md* in the repository.

A.2.1 Security, privacy, and ethical concerns

There are no risks for evaluators or their machines while executing the artifacts code.

For privacy consideration we have not published the original beam model, we used in the paper for the simulations and for executing the RECORD attack in the real world. We did not publish the original model to avoid a simple tracking of Iridium devices in the real world, as mentioned in section 6.6 in the paper. However we published a modified beam model and provide a comparison on the RECORD attack between the unpublished and the published beam model. This is done by the evaluation script *graph_simulation_paper_vs_published.py*, the numbers of the comparison can be found at the end of the *README.md* in the repository.

A.2.2 How to access

The artifacts are available on GitHub: https://github.com/ErJedermann/RECORD/tree/usenix24

A.2.3 Hardware dependencies

None

A.2.4 Software dependencies

- Linux based operating system.
- Python with pip.
- Libraries listed in *requirements.txt* in the repository.
- Up to date browser.

It was tested on Ubuntu 18.04 and Linux Mint 20.2, python 3.8 and 3.11, browsers Firefox 105 and Chromium 122.

A.2.5 Benchmarks

None

A.3 Set-up

A.3.1 Installation

Prerequirements: installed python with pip

```
git clone https://github.com/ErJedermann/RECORD.git
cd RECORD
git checkout usenix24
python3.11 -m venv ./venv/
source venv/bin/activate
pip install -r requirements.txt
```

A.3.2 Basic Test

There are two tests to ensure full functionality:

Test 1: The simulation:
Run `python simulations_attackerTypes_fibo.py`.
This executes the RECORD simulation, using the default parameters at the end of the script: 10 iterations, inter observer distance 100 km, observation duration 60 seconds, 3 eavesdroppers, using a noisy beam model and weak observations.

Expected execution time: about 8 minutes.

Expected console output: A live tracker of the currently simulated iteration. Which looks like:

```
starting with [100]km, [60]sec and 3eves
generic: <timestamp>: d:1/1 t:1/1 i:1/10
(...)
generic: <timestamp>: d:1/1 t:1/1 i:10/10
started at <timestamp>, until <timestamp>. took <n> sec
```

Result: After every iteration, the calculated RoIs will be appended to the respective file. Currently this is *my_simulations/duration_and_type/100kmFibo_cont-_60sec_3eves_weakEvents_noisyPrediction.csv*

Test 2: The data evaluation:
Run `python graph_simulation_attackerTypes.py`.
This loads a row of simulation data files (which will be printed in the console), reads out the data and combines it to a graph. The interactive graph is opened in the browser as a html site.

Expected execution time: 2 seconds.

Expected console output: A list of the used simulation data files. It starts with:

```
open: 100kmFibo_cont_60sec_3eves_weakEvents_noisyPredict...
open: 100kmFibo_cont_60sec_3eves_weakEvents.csv
open: 100kmFibo_cont_60sec_3eves.csv
(...)
```

Result: Open a interactive graph in the browser as html page. It not modify any files.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): *The RECORD attack is able to estimate a Region of Interest (RoI) where a satellite terminal is located. This is proven by experiments (E3) and (E4).*

(C2): *Major effects on the size of the RoI have the observation duration and the information that is available for the attacker. This is proven by experiments (E1) and (E3).*

(C3): *Minor effects on the performance have the number of observers and the inter-observer-distance. This is proven by experiments (E2) and (E4).*

A.4.2 Experiments

(E1): *[5 human-minutes + 2 compute-seconds]: Create and evaluate a graphic showing differences in observation duration on different attacker types (paper figure 10).*
Preparation: None.
Execution: Execute the python command `python graph_simulation_attackerTypes.py`
Results: The result is a graph, showing the relation between the observation time and the RoI sizes on different attacker types. With 4 hours observation duration the RoI can be reduced by a factor of 400, compared to the inital guess (528428 km^2 to 1257 km^2). Depending on the available attacker information (attacker types) a factor of more than 10k is possible (1257 km^2 to 0.0855 km^2), realistic is a factor of 14 (1257 km^2 to 92 km^2) The results are comparable to Figure 10 in the paper and the associated section 5.2.

(E2): *[5 human-minute + 2 compute-seconds]: Create and evaluate a graphic showing differences in inter observer distances and different observer amounts (paper figure 11).*
Preparation: None.
Execution: Execute the python command `python graph_simulation_receiverDistances.py`

Results: The result is a graph, showing different RoI sizes over different inter observer distances and for different observer amounts. Different inter observer distances can improve the results by a factor of 3 (557 km^2 to 183 km^2). While increasing the number of observers improves the RoI size by a factor of 2.5. The results are comparable to Figure 13 in the paper and the associated section 5.3.

(E3): *[30 human-minutes + 580 compute-hours]: Conduct simulations to verify the relations between observation duration, attacker types and the estimated RoI.*
Preparation: Adapt the parameters of *simulations_attackerTypes_fibo.py* and *graph_simulation_attackerTypes.py* according to the instructions in *simulation_setup_1.md*.
Execution: Execute three simulations and one graph generation according to the instructions in *simulation_setup_1.md*.
Results: This experiment simulates three scenarios and thereby generates all data from scratch, required for a full evaluation of the relations between observation duration, attacker types and the estimated RoI. With the new generated data, a graph is created showing the relation between the observation time and the RoI sizes on different attacker types. The new created graph will be comparable to the result graph of experiment E1. Variations up to a factor of two are expected due to the altered published beam model.

(E4): *[30 human-minutes + 840 compute-hours]: Conduct simulations to verify the relations between inter observer distances, different observer amounts and the estimated RoI.*
Preparation: Adapt the parameters of *simulations_attackerTypes_fibo.py* and *graph_simulation_receiverDistances.py* according to the instructions in *simulation_setup_2.md*.
Execution: Execute three simulations and one graph generation according to the instructions in *simulation_setup_2.md*.
Results: This experiment simulates three scenarios and thereby generates all data from scratch, required for a full evaluation of the relations between inter observer distances, different observer amounts and the estimated RoI. With the new generated data, a graph is created showing the effect of different inter observer distances and different observer amounts. The new created graph will be comparable to the result graph of experiment E2. Variations up to a factor of two are expected due to the altered published beam model.

A.5 Notes on Reusability

The framework is capable of simulating different attack scenarios of the RECORD attack: An attacker observes incoming traffic to a target device while the transmitters, the satellites, are moving. Since the attacker knows the position of the satellites and the spreading pattern (via the beam model) of the signals, he can gather information about possible locations of the target device.

By adapting the parameters in the main simulation script, many different scenarios can be simulated. This enables interested readers to dive deeper into the attack and adapt it to their own scenarios.

Another promising approach is to exchange the satellite beam model and to evaluate the attacks behaviour on different satellite systems or independent of a specific satellite system.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2024/.