Security of GPS/INS based On-road Location Tracking Systems

Sashank Narain, Aanjhan Ranganathan, Guevara Noubir
Northeastern University
Researchers use spoofing to 'hack' into a flying drone

How to cheat at Pokémon Go and catch any Pokémon you want without leaving your couch

The Russians are screwing with the GPS system to send bogus navigation data to thousands of ships

GPS Spoof Hits Geneva Motor Show
Incident leaves GPS units showing a location in England and a date 17 years in the future.

Researchers commandeer £50m superyacht with GPS-spoofing
No constraints

Route Estimate with Road Constraints
Given a roadmap and assuming inertial sensor data is monitored (in addition to GPS)

Is it possible for an attacker to spoof their navigation path / final destination?
Contributions

- Developed algorithms that derive potential destinations reachable without raising an alarm
  - Leveraging regular patterns that exist in urban road networks
  - Rendering any GPS/INS based monitoring system useless

- First real-time integrated GPS/INS spoofer that accounts for traffic fluidity, lights and stop signs
  - Dynamically generates GPS spoofing signals
  - And it works in the real world!
High-level Attack Overview

- Road Network
- Graph Construction
  - Graph
  - Start and End Location
  - Spoof Routes Generation Algorithm
  - Escape Routes Generation Algorithm
  - Selected Routes
- Selected Routes
- Real-time Spoofer
A Visual Representation

Attack Algorithm
Graph Construction

- **Edges** → Intersections
  - Contains turn angle
- **Vertices** → Road between Intersections
  - Contains curvature + travel time
Intuition for Spoof Routes Generation

- Maximize Probability of Spoofing
  - Use curves + turns common in the road network
Spoof Routes Generation Algorithm

- **Extended Depth First Search**
  - **Filter routes** unlikely to reach destination
    - Define constraints for likely routes
    - Direct routes towards destination
  - **Score routes** that reach destination
    - Using turn angles and road curvature
    - Compound probability of all vertices in path
  - Select the top scoring paths

```
Input: G = (V, E), Loc(s), Loc(d), N_P
Output: S = {p_1, ..., p_N_P}

1. Initialization: S ← ∅; p ← []; v ← ∅
2. s ← getSourceVertex(Loc(s))
3. d ← getDestinationVertex(Loc(d))
4. GenerateSpoofedPaths(s, d)
5. S ← selectTopPaths(S, N_P)

6. function GenerateSpoofedPaths(s, d):
   7. p ← p + [s]
   8. v ← v ∪ {s}
   9. if s = d then
      10. S ← S ∪ {p}
   11. else
      12. for e ∈ V such that (s, e) ∈ E do
          13. if e ∈ v and Filter(s, e, p) passed then
              14. p.score ← p.score * Score(s, e, p)
              15. GenerateSpoofedPaths(e, d)
          end
      end
      16. p ← p - [s]
      17. v ← v - {s}
```
Intuition for Escape Routes Generation

- Exploit noise sensitivity of sensors
  - Accelerometers sensitive to road irregularities
  - Magnetometer sensitive to vehicle magnets
  - Gyroscopes can have significant drift
Escape Routes Generation Algorithm

- **Extended Depth First Search**
  - Ensures *spoof routes and escape routes are topologically similar*
    - Accounting for varying road curvatures and lengths
    - Renders any sensor monitoring useless
  - **Filter paths** dissimilar to spoof routes
    - Exceeded the turn count
    - Turn, Curvature or Distance is outside noise threshold

```plaintext
Input: G = (V, E), S_I
Output: N_P, E = \{p_1, \ldots, p_{N_P}\}

1 Initialization: E ← ∅; N_P ← 0; p ← []; v ← ∅
2 s ← getSourceVertex(S_I)
3 t ← getTurnsCount(S_I)
4 GenerateEscapePaths(s, t)

function GenerateEscapePaths(s, t):
    p ← p + [s]
    v ← v ∪ \{s\}
    if len(p.turns) > t then
        return
    if len(p.turns) = t then
        E ← E ∪ \{p\}
        N_P ← N_P + 1
    for e ∈ V such that (s, e) ∈ E do
        if e ∉ v and Filter(s, e, p, S_I) passed then
            p.curve ← updateCurvature(s, e, p)
            p.turns ← updateTurns(s, e, p)
            p.score ← p.score * Score(s, e, p, S_I)
            GenerateEscapePaths(c, t)
        end
    p ← p − [s]
    v ← v − \{s\}
```

Real-World Spoofer

- **Generic system** usable in many different attack scenarios
- In case of Road Networks -
  - First implementation to account for traffic fluidity, traffic lights and stop signs
  - On receipt of driver’s real (spoof) location -
    - Calculates a escape location and bearing **efficiently within ~5 ms**
    - **GPS spoofer generates NMEA packets** for escape location
    - **Magnetometer Spoofer generates magnetic field** for escape bearing
Real-World Spoofer Evaluation

- GPS lock never lost during 10 routes
- **Maximum delay of 60 ms** between spoof and escape location
- All sensor errors within range of error threshold
Evaluation Methodology

- Perform simulations for 10 global cities
  - Major transportation and logistic hubs
  - With diverse road networks
    - Structured & Grid-like -> E.g., Manhattan and Chicago
    - High variability -> E.g., London and Paris
    - Somewhere in between -> E.g., Boston and San Francisco

- Simulate 1000 routes in each chosen city
  - “Residence” to “Office” using OpenStreetMap
  - Measure -
    - Maximum Displacement from Intended Destination
    - Estimated Coverage Area of Escape Routes
Maximum Displacement from Intended Destination

- **Significant Deviation possible**
  - In grid-like cities
    - > 10 km for 50% routes
    - > 20 km for 20% routes
  - In other cities
    - > 10 km for 10% routes
    - Several routes with 30-40 km deviation
Estimated Coverage Area of Escape Routes

- **Monte-Carlo Simulations**
  - Define a circle with
    - Source as center
    - Distance from destination as radius
  - Calculate area of escape destinations
    - Within the circle
    - Assuming user walks 50m around parking

- **Possible to Cover**
  - > 30% area in grid-like cities
  - > 8% area for long routes (~10 kms)
Countermeasure - “Secure Paths” Algorithm

- Generate routes with low probability of spoofing
  - Reverse the spoof routes generation algorithm
  - Run escape routes generation algorithms
  - Choose spoof route with least escape routes
Summary & Questions?

- Developed algorithms that derive potential destinations reachable without raising alarms on GPS/INS tracking systems
  - Possible to deviate > 10 km (> 20 km) for 50% (20%) routes in grid-like cities
  - Possible to deviate 30-40 km for many routes in all cities

- First real-time integrated GPS/INS spoofer that accounts for traffic fluidity, lights and stop signs
  - GPS lock never lost during 10 routes
  - All sensor errors within range of threshold