CaTch: A Confidence Range Tolerant Misbehavior Detection Approach

Joseph Kamel; Arnaud Kaiser; Ines ben Jemaa; Pierpaolo Cincilla; Pascal Urien

2019 IEEE Wireless Communications and Networking Conference (WCNC)

Motivation

- Misbehavior Detection Systems in Vehicular Communications.
- Current MBD systems fail in distinguishing between intentional and non-intentional erroneous mobility information.
- Integrating benign sensors error in the plausibility checks would improve misbehavior detection.

Proposed Solution

- Include sensors inaccuracy/tolerance in plausibility checks.
- “Confidence Range” field in IEEE BSM and CAM for each mobility parameter.
- Uncertainty Factor *f*:

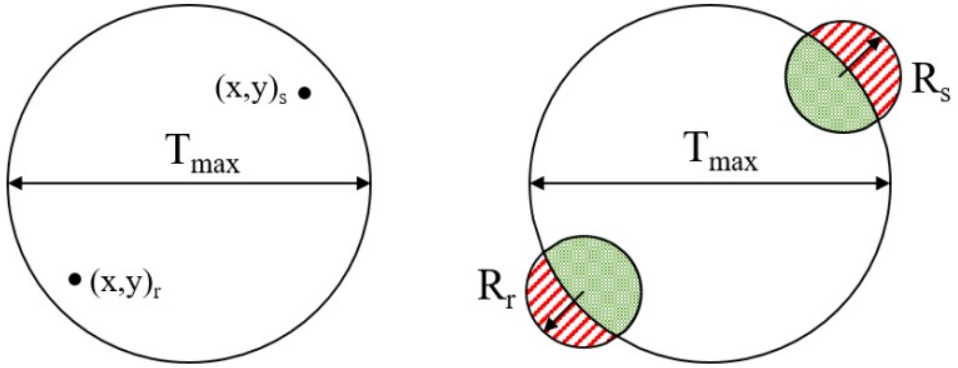
  For each check, CaTch finds Uncertainty Factor f

  f(mobility parameter, mobility parameter confidence range, plausibility thresholds)

  In the range [0,1]

  0 as malicious and 1 as no signs of misbehavior

R_x	$\triangleq$	Position confidence range in beacon x
V_x	$\triangleq$	Claimed speed in beacon x
C_x	$\triangleq$	Speed confidence range in beacon x
D_x	$\triangleq$	Claimed heading in beacon x
Δt_{ij}	$\triangleq$	Time separating beacons i and j
d_{ij}	$\triangleq$	Distance separating beacons i and j
A_x	$=$	πR_x^2



(a) Legacy Version (b) CaTCh Version

Fig. 1: Range Plausibility Check

T_{max}	$\triangleq$	Communication Diameter
$A_{T_{max}}$	$=$	$\frac{\pi T_{max}^2}{4}$
a_r	$=$	$A_{T_{max}} \cap A_r$
a_s	$=$	$A_{T_{max}} \cap A_s$

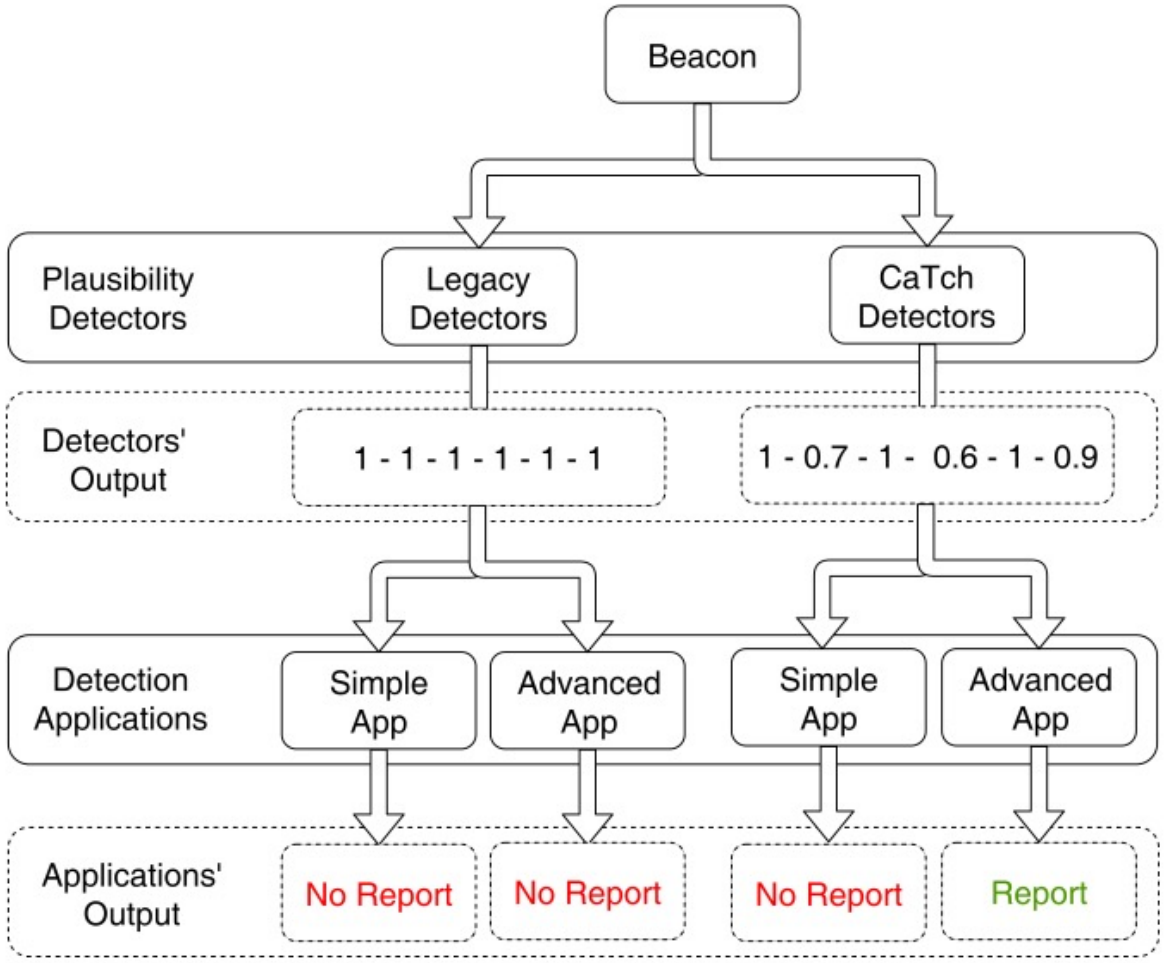
$$f = (a_r + a_s)/(A_r + A_s)$$

Proposed Solution

- Example: Position Plausibility check
- A vehicle cannot report a position out of T_{max}.
- Position Confidence Range intersection areas with comm. range area are plausible positions.
- Green is plausible, Red is not plausible.

Evaluation-Experiment Setup

- Simulator: VEINS.
  - ✓ *21 Vehicles.*
  - ✓ *Attack Density = 0.1.*
  - ✓ *Two attacks: ConstPosOffset, Sybil.*
- Detection Applications:
  - ✓ *Determines whether to report a message or not.*
  - ✓ *Simple (Threshold) Vs. Advanced (ML Detector) for both Legacy and CaTch.*
  - ✓ *CaTch Threshold: Report if f < 0.5*
  - ✓ *CaTch Advanced: MLP, Trained on history of a couple of previous messages.*
- Sensors Inaccuracies:
  - ✓ *GPS position inaccuracies.*
  - ✓ *Radom inaccuracies N(0,1).*



Evaluation-Performance Metrics

- For MBD, FN has higher gravity than FP.
- Recall/Precision trade-off. (FP/FN trade-off)
- Bookmaker Informedness: Random guessing.
- MCC: Accuracy when one class is significantly larger than the other.

	Genuine	Misbehaving
Reported	FP	TP
Not Reported	TN	FN

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$F_1\,Score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

$$BM = \frac{TP}{TP + FN} + \frac{TN}{TN + FP} - 1$$

$$MCC = \frac{TP + FP}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Evaluation-Results

- Performance depends on attack type.
- ConstPosOffset:
  - ✓ *Attack is easy detected by Simple App (No improvement for CaTch over Legacy).*
  - ✓ *ML app with CaTch improve Genuine detection with a higher precision and Lower recall. (Not beneficial at all).*
- Sybil:
  - ✓ *Attack is sending plausible data from virtual cars.*
  - ✓ *ML app with CaTch improve Attack detection with a higher recall.*

App	Scenario Detectors	Evaluation Metrics		
Threshold		**Recall**	**Precision**	**Accuracy**
	Legacy	0.7621	0.9233	0.9691
	CaTch	0.7625	0.9207	0.9689
		Δ0.1%	Δ-0.3%	Δ0.0%
		F_1Score	**BM**	**MCC**
	Legacy	0.8350	0.7548	0.8227
	CaTch	0.8342	0.7550	0.8216
		Δ-0.1%	Δ0.0%	Δ-0.1%
Machine Learning		**Recall**	**Precision**	**Accuracy**
	Legacy	0.7642	0.9375	0.9706
	CaTch	0.7498	0.9721	0.9721
		Δ-1.9%	Δ3.7%	Δ0.2%
		F_1Score	**BM**	**MCC**
	Legacy	0.8420	0.7584	0.8312
	CaTch	0.8466	0.7473	0.8400
		Δ0.5%	Δ-1.5%	Δ1.1%

(a) Constant Offset Scenario

App	Scenario Detectors	Evaluation Metrics		
Threshold		**Recall**	**Precision**	**Accuracy**
	Legacy	0.3976	0.9504	0.7468
	CaTch	0.4203	0.9457	0.7546
		Δ5.7%	Δ-0.5%	Δ1.1%
		F_1Score	**BM**	**MCC**
	Legacy	0.5607	0.3834	0.5013
	CaTch	0.5819	0.4038	0.5155
		Δ3.8%	Δ5.3%	Δ2.8%
Machine Learning		**Recall**	**Precision**	**Accuracy**
	Legacy	0.3928	0.9498	0.7446
	CaTch	0.7961	0.9102	0.8852
		Δ102.7%	Δ-4.2%	Δ19.8%
		F_1Score	**BM**	**MCC**
	Legacy	0.5556	0.3783	0.4967
	CaTch	0.8494	0.7424	0.7618
		Δ52.9%	Δ96.2%	Δ53.4%

(b) Sybil Attack Scenario

Conclusion

- Integrating the mobility parameters' Confidence Range with ML App increases the MBD ability to detect Sybil attack.
- CaTch detector outputs a real value in the range of [0,1] for each check.
  - ✓ *Legacy outputs either 0 or 1 for each check.*
  - ✓ *CaTch can be trained to generate a Fingerprint for each attack.*

Discussion

- Implementation Questions:
- *How is the Sybil attack implemented in the simulator? Are the inaccuracies added within the simulation time?*
- *Do the virtual cars include a random Confidence Range in the data they send?*
- *Would that decrease the obtained improvement in recall?*
- would including the confidence range increase FP/FN in case of sensor breakage, inaccurate/high tolerance sensors,…?
- *Only GPS inaccuracies are included. Is it ok as GPS is the main positioning system in this context?*
- Evaluation on other attacks
- ML detector
- *Simplicity, short training history*
- Generate a fingerprint for each attack
- *Is this useful?*
- Performance metrics: Undiscussed improvements in BM and MCC.

Thank you