SHIELDNET: An Adaptive Detection Mechanism against Vehicular Botnets in VANETs

Mevlut Turker Garip, Jonathan Lin, Peter Reiher, Mario Gerla
Department of Computer Science, University of California Los Angeles
{mtgarip, jonathan.lin, reiher, gerla}@cs.ucla.edu

***Abstract*—Vehicular ad hoc networks (VANETs) are designed to provide traffic safety by enabling vehicles to broadcast information—such as speed, location and heading—through inter-vehicular communications to proactively avoid collisions. However, the attacks targeting these networks might overshadow their advantages if not protected against. One powerful threat against VANETs is vehicular botnets. In our earlier work, we demonstrated several vehicular botnet attacks that can have damaging impacts on the security and privacy of VANETs. In this paper, we present SHIELDNET, the first detection mechanism against vehicular botnets. Similar to the detection approaches against Internet botnets, we target the vehicular botnet communication and use several machine learning techniques to identify vehicular bots. We show via simulation that SHIELDNET can identify 77 percent of the vehicular bots. We propose several improvements on the VANET standards and show that their existing vulnerabilities make an effective defense against vehicular botnets infeasible.**

***Index Terms*—Vehicular Ad Hoc Networks, VANET Security, Vehicular Botnets, Vehicular Botnet Communication, Intrusion Detection, Machine Learning, Reputation-Based Security**

I. Introduction

Many traffic accidents are caused by unsafe driver actions due to insufficient traffic information [20]. In vehicular ad hoc networks (VANETs), vehicles exchange traffic information through Basic Safety Messages (BSMs) [21], which contain current speeds, locations, directions, etc. Vehicles use this information to prevent collisions by automated and prompt reactions to abrupt traffic events. However, the wide acceptance of VANETs depends on their security since attacks on them might have fatal consequences, unlike most other systems.

One of the most powerful adversaries against VANETs is vehicular botnets. We pioneered the concept, argued its feasibility, and demonstrated the first vehicular botnet attack—namely a *congestion attack*—and its effectiveness in [6]. This attack can cause traffic congestion on any road of interest, making it and other roads surrounding it virtually unusable. We also presented BOTVEILLANCE in [8]—a vehicular botnet surveillance attack—which violates one of the most fundamental requirements of VANETs, location privacy. We finally demonstrated RIoT in [9]—the first attack against Internet of Things (IoT) devices using vehicles—which can compromise a significant percentage of the IoT devices in an area of interest by taking advantage of the mobility and collective communication range of vehicular bots. Considering the dangers vehicular botnets impose through such attacks, it is crucial to eliminate them for the safety and privacy of drivers.

In our earlier work, we designed the first vehicular botnet communication protocol, GHOST [7], which exploits the existing vulnerabilities in the VANET standards to avoid detection. The standards assign finer granularity than necessary to some fields in BSMs, particularly *speed, latitude, longitude* and *positional accuracy*. GHOST splits and injects secret messages into the least significant bits of these four fields. It remains hidden because the magnitude of the fine granularity in these fields makes the variations in their values caused by the injections less than the natural variations. While GHOST is not the only possible mechanism for vehicular botnets to coordinate their activities, future mechanisms are very likely to be similar to GHOST since using the VANET control channel for vehicular botnet communication is the stealthiest approach—given that the control channel is already standardized to be frequently used by everyone [21]. Since only BSMs are allowed to be sent through this channel, future mechanisms will also be forced to work with BSMs to transmit botnet messages, which will have similar side effects to GHOST. Since vehicular bots have to use GHOST or a similar communication protocol to secretly coordinate their attacks, similar to Internet botnets, the most effective defense against such cooperative adversaries is targeting their communication protocol to identify them. Instead of trying to detect which specific vehicular botnet attack is being performed and defend against it, going after the common mechanism among such attacks—vehicular botnet communication—provides a defense against not only the known threats but also the future ones. In this paper, we target GHOST since any detection mechanism against it can also largely be used against future mechanisms anyway due to the aforementioned reasons.

In this paper, we present SHIELDNET, a detection mechanism against vehicular botnets, which applies machine learning techniques to search for evidence of GHOST usage and identify the participants as possible vehicular bots. Since the only indication of GHOST usage is the effects that its secret message injections have on the values of the aforementioned BSM fields, each machine learning algorithm of SHIELDNET is chosen based on the expected change pattern in each corresponding field value to be able to detect the anomalies. Our work is the first implementation of a defense against vehicular botnets. After identifying the most applicable machine learning algorithms for the BSM fields, we determine their best configurations. However, SHIELDNET is designed in an adaptive manner, meaning that its machine learning algorithms can easily be replaced or extended without affecting its reputation-based vehicular bot identification component. We also discuss the vulnerabilities in the VANET standards in more depth and show via simulation that GHOST remains infeasible to detect even with SHIELDNET as long as these vulnerabilities exist. We then implement the vulnerability fixes so that our defense is evaluated with the improved standards.

In Section II, we discuss the existing defense approaches for Internet botnets and if they are applicable to the problem of vehicular botnet detection. In Section III, we describe the GHOST protocol and present the design details of SHIELDNET. In Section IV, we discuss the configurations of the machine learning algorithms for the evaluation of SHIELDNET, and show the accuracy of our detection mechanism via simulation. In Section V, we describe the vulnerabilities in the standards that GHOST exploits and that need to be fixed as future work. In Section VI, we conclude with the contributions of our work to future research on VANET security.

II. Related Work

The concept of botnets has received much research attention due to the high impact of these networks of compromised machines. Internet botnet detection, particularly, is a well-studied subject and there are numerous approaches proposed to tackle this problem. Most botnet detection techniques are based on passive network traffic monitoring and analysis [5] [19], similarly to our approach. These techniques attempt to detect botnets by identifying their command and control (C&C) channels that are used for their coordination; some C&C protocol models are presented in [2]. They try to achieve this by searching for anomalies in network traffic [10] [14]—such as high latency, high traffic volumes, and traffic on unusual ports—or specifically in DNS traffic [1], which are caused by C&C activities. However, the C&C channel of GHOST neither causes these network anomalies nor uses the same network architecture as Internet botnets. Vehicular botnet messages are already injected into BSMs, which are broadcast to everyone with a high frequency as per the VANET standards. This way, the C&C activities of vehicular botnets do not change the existing network conditions in any way. Therefore, Internet botnet detection approaches are not applicable to our problem.

In the context of VANETs, there is no existing work in the direction of detecting vehicular botnets. Our work is the first step towards identifying the unique characteristics of vehicular botnets to detect them, and the first implementation of such a detection mechanism. We determine the standard and most applicable machine learning algorithms [3] [4] [13] based on the expected change patterns in the injected BSM field values and the format of each field, and use these algorithms to detect anomalies in them. The scope of this paper is not designing new machine learning algorithms; our novelty comes from designing an adaptive vehicular bot detection framework that houses them. The framework provides organized and efficient data collection for the built-in machine learning algorithms, and uses their outputs for its reputation-based vehicular bot identification mechanism. Its adaptive design makes the addition of new machine learning algorithms or the removal of the built-in ones very easy for future researchers.

The vehicular bots that are identified by our detection mechanism would naturally need to be removed from the network. Since there are already many existing mechanisms for the eviction of misbehaving vehicles—such as [15] and [16]—which are surveyed in [17], designing a new one is out of the scope of this paper.

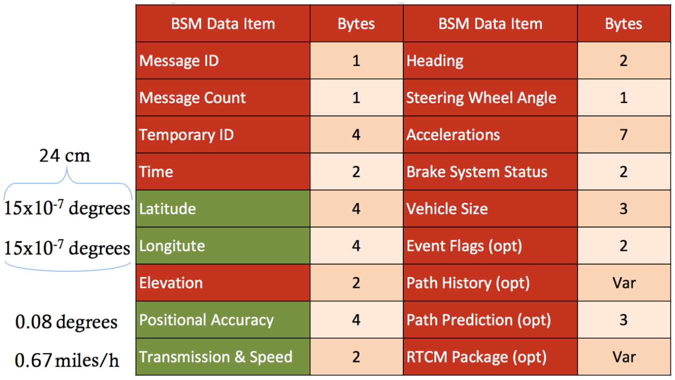
III. SHIELDNET

A. Overview

Here we first discuss the GHOST protocol and which of the injected BSM fields are suitable to be used for vehicular bot detection. We then describe the design details of SHIELDNET—the machine learning algorithms applied on the suitable fields to detect anomalies in their values and the reputation-based analysis of these anomalies to identify vehicular bots. SHIELDNET is intended for use by authorities with the means to monitor the exchanged BSMs in the whole *shielded area*. It is designed in an adaptive manner so that authorities can easily replace the machine learning algorithms applied on the data whenever they feel necessary without affecting the reputation-based identification of vehicular bots. We use a labeled dataset, which is a large set of BSMs—collected during our vehicular botnet simulation—where the injected ones are marked, for the investigations and evaluation in this paper; however, SHIELDNET uses unsupervised machine learning algorithms and does not require any labeled data to identify vehicular bots.

B. GHOST

There are two wireless channels in VANETs that vehicles use to send messages: control and service channel. The control channel is reserved only for broadcasting BSM messages, while the service channel is shared among all the other applications that are not related to traffic safety. Vehicular bots need a communication channel that they can frequently use to coordinate their attacks without raising any flags. They are not allowed to send any messages other than BSMs in the control channel, and it might raise suspicion if there is a continuous traffic in the service channel among only the same set of vehicles. GHOST enables vehicular bots to conceal the botnet messages inside the broadcast BSMs. Since BSMs are already frequently broadcast (every 100 msec [21]) to everyone, vehicular bots can piggyback their messages onto them without raising suspicion. They achieve this by splitting their two-byte botnet messages into four parts and injecting them to the least significant four bits of the speed, latitude, longitude and positional accuracy fields in their outgoing BSMs. The original botnet message can then be reconstructed from these four fields by the receiver bots. The injected messages will look like random noise to authorities monitoring the BSM broadcasts even if they can be identified, since they are encrypted with a random scheme that changes periodically. Further details of the GHOST mechanism can be found in [7].



	BSM Data Item	Bytes	BSM Data Item	Bytes
	Message ID	1	Heading	2
	Message Count	1	Steering Wheel Angle	1
	Temporary ID	4	Accelerations	7
24 cm	Time	2	Brake System Status	2
15x10^{-7} degrees	Latitude	4	Vehicle Size	3
15x10^{-7} degrees	Longitute	4	Event Flags (opt)	2
	Elevation	2	Path History (opt)	Var
0.08 degrees	Positional Accuracy	4	Path Prediction (opt)	3
0.67 miles/h	Transmission & Speed	2	RTCM Package (opt)	Var

Figure 1. BSM fields where botnet messages will be injected into and the effect of the injection on the field values

Figure 1 shows the content of a BSM according to the VANET standards. The injected fields are highlighted and the theoretical maximum changes in their values due to the injections are shown; the theoretical maximum change in a field value is the change when 0xF is injected while the field has 0x0 as its least significant four bits. The vulnerability in the standards that allows GHOST to stay hidden is the fine granularity assigned to these fields. Due to this vulnerability, even the theoretical maximum changes GHOST makes in the field values remain well under the tolerated natural variations. In fact, experiments reveal that the observed maximum and average changes caused by GHOST are significantly lower than the theoretical maximum changes, as shown in Figure 2.

	Average	Max	Theoretical Max
Latitude (d)	5.02E-07	7.67E-07	15E-07
Longitude (d)	5.25E-07	7.73E-07	15E-07
GPS Position (cm)	8.08	12.11	24
Positional Accuracy (d)	0.02	0.03	0.08
Speed (miles/h)	0.29	0.40	0.67

Figure 2. Changes in the values of the injected fields during the experiment

Detecting the GHOST usage by searching cleartext botnet messages in the BSM fields or checking extreme variations in their values is infeasible due to all the aforementioned reasons. Therefore, SHIELDNET uses machine learning approaches to detect the anomalies in the fields by predicting their values based on the features and learned behavior specific to them. Since there is no predictable behavior for the positional accuracy field—the noise in the calculation of the vehicle's heading relative to true north—only the speed, latitude and longitude fields are used for the anomaly detection.

C. Anomaly Detection in Speed

Prior to applying machine learning to detect anomalies in speed values, we performed statistical analysis on the labeled dataset to look for a predictable pattern in how the GHOST injections might effect these values. We first investigated if the injections yield higher or lower speed values on average than the non-injected values. Since the effect of the four-bit injections on the speed values in two-byte granularity is very low as aforementioned, we calculated the average of values represented by the four least significant bits of the speed field for all the injected and non-injected BSMs. We observed that these two averages were nearly identical, meaning that there was no predictable pattern. We repeated the same analysis for the one, two and three least significant bits as well but the outcome was the same. We then investigated if there was a bit pattern associated with the injections in the speed values, that is, if the injections cause more ones or more zeros in the binary speed values on average than the non-injected values. We calculated the average number of ones and the average number of zeros in the speed field for all the injected and non-injected BSMs. Since the content of the injection is random due to the botnet message being encrypted with a random scheme that changes periodically, this analysis did not reveal any predictable pattern as well. Due to the unpredictability of the effects of the injections caused by the randomness in their nature, the only effective approach to detect anomalies in speed values is learning a vehicle's recent mobility pattern and determining the plausibility of its following speed broadcasts based on how well they fit this pattern.

SHIELDNET uses a machine learning algorithm, specifically moving average using *discrete linear convolution* [3], to model each vehicle's mobility pattern based on its speed advertisements and detects outliers with a stationary standard deviation. Each outlier is then reported to the reputation-based vehicular bot identification mechanism.

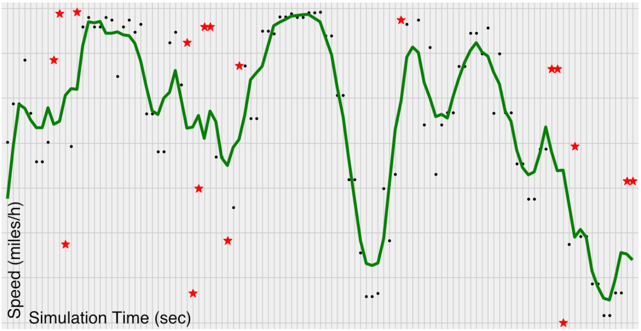


Figure 3. Model of a vehicle's mobility and outliers during the simulation

SHIELDNET creates a separate mobility model for each vehicle and performs anomaly detection simultaneously for all of them. Figure 3 shows the mobility model created for one of the vehicles in the simulation (green line). The model is continuously updated with each speed advertisement from the vehicle received by authorities monitoring the BSM broadcasts in the *shielded area*. Each advertised speed value (black dot) is tested against the model built so far and marked as an outlier (red star) if it does not fit the model based on the configurable stationary standard deviation. If it is not considered as an outlier, the model is then updated to include this new speed value in its convolution.

D. Anomaly Detection in Latitude and Longitude

Detecting anomalies in latitude and longitude values by learning their change patterns is impractical for several reasons. First, depending on the heading, only the latitude or only the longitude or both might change and the amount of change in each is determined by both the heading and speed. As a result, the convolution approach alone would not be able to predict their next values. Second, even if another machine learning approach could incorporate the effect of heading and speed in its model, there is another factor in addition to the effect of injections that might be considered as an anomaly: GPS error. Therefore, the only effective approach to detect anomalies in latitude and longitude values is learning the GPS error pattern to be able to distinguish the effect of the injections from the GPS error as anomalies.

The level of GPS error is determined by the environmental factors such as interference, obstacles, etc. Therefore, vehicles under the same environmental factors are expected to have similar GPS errors. In other words, a vehicle having a significantly different GPS error than other vehicles under the same environmental factors would indicate an anomaly—likely to be caused by the injections into the vehicle's latitude and longitude values. In order to detect such anomalies, GPS errors in the same environment need to be clustered together based on their similarity. Since being in the same environment is represented as having both similar latitudes and longitudes, 2D clustering approaches would fail to model one of these dimensions required: latitude, longitude and GPS error.

SHIELDNET uses the 3D version [13] of DBSCAN [4]—a density-based cluster discovery algorithm for large spatial databases with noise—for dividing the spatial GPS errors into multiple 3D clusters based on their similarity and proximity.

SHIELDNET's GPS error calculations start for each vehicle when its first two BSMs, say BSM_1 and BSM_2, are received. Using the speeds and headings advertised in BSM_1 and BSM_2, as well as BSM_1's and BSM_2's timestamps, BSM_2's expected latitude and longitude—lat_{exp} and $long_{exp}$—are calculated. The GPS error for the vehicle would then be the Euclidian distance between lat_{exp} and $long_{exp}$ and the actual latitude and longitude advertised in BSM_2—lat_{adv} and $long_{adv}$. The next GPS error for BSM_2 and BSM_3 is calculated using BSM_2's lat_{exp} and $long_{exp}$ instead of its lat_{adv} and $long_{adv}$, which are inaccurate due to the GPS error.

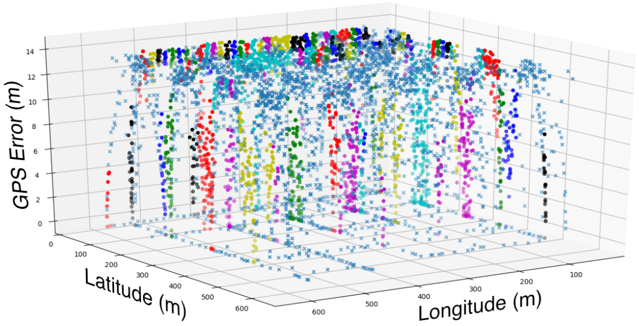


Figure 4. 3D clusters of the GPS errors and outliers calculated by DBSCAN

Figure 4 shows how the calculated GPS errors are divided into multiple spherical 3D clusters (different colored circles). Clusters are created based on the configurable maximum cluster radius and minimum number of points required within this radius to form a cluster. Figure 4 depicts only the data belonging to a subset of the *shielded area*. The latitude (y-axis) and longitude (x-axis) values in the graph are the distance in meters from the bottom and left edge of the *shielded area*, respectively. Each calculated GPS error is tested against the model built so far and marked as an outlier (blue cross) if it cannot belong to a cluster. The outlier is then reported to the reputation-based vehicular bot identification mechanism.

E. Reputation-Based Identification of Vehicular Bots

SHIELDNET performs a reputation-based analysis on all of the outliers reported by the built-in machine learning algorithms in order to identify vehicular bots. This identification mechanism is designed to provide authorities with a list of vehicles that are most likely to be bots and to constantly update this list while the machine learning algorithms simultaneously generate new outlier reports.

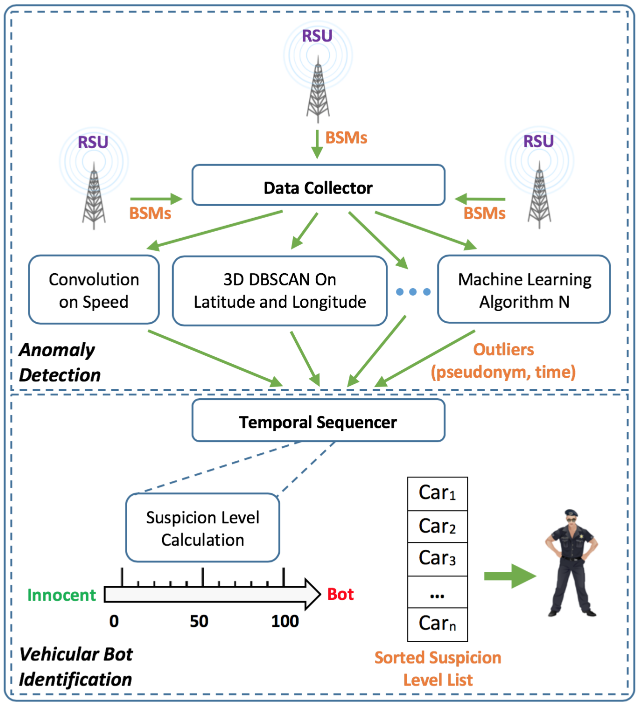

RSU
BSMs
RSU
RSU
Data Collector
BSMs
BSMs
Convolution on Speed
3D DBSCAN On Latitude and Longitude
Machine Learning Algorithm N
Anomaly Detection
Outliers (pseudonym, time)
Temporal Sequencer
Suspicion Level Calculation
Innocent
Bot
0
50
100
Car1
Car2
Car3
...
Carn
Vehicular Bot Identification
Sorted Suspicion Level List

Figure 5. Diagram of SHIELDNET's components and their interactions

Figure 5 shows the individual components of SHIELDNET, how they communicate with each other and the input/output of each component. These components operate in a pipelined manner; each component immediately processes the partial data inputted and forwards the result to the next component while the new data is constantly being inputted. This pipeline starts with the *data collector* where all the BSMs being broadcast in the *shielded area* are inputted. Authorities collect this data by using network sniffers with a high communication range—most likely the Roadside Units (RSUs). The *data collector* then organizes the inputted BSMs and passes them to the machine learning algorithms as input. There can be any number of machine learning algorithms at the next step in the pipeline and it might change over time. Therefore, the *data collector* is designed in an adaptive manner to detect the current algorithms and forward the data accordingly. After BSMs are passed to each machine learning algorithm, outlier detection is performed on the data relevant to the algorithm and outliers are reported to the *temporal sequencer*. Each outlier report is a pair of the pseudonym (identifier) of the vehicle that broadcasts the outlier and the timestamp of this broadcast. Based on these reports, the *temporal sequencer* calculates the suspicion level for each vehicle, which determines its likelihood of being a bot. Finally, the *temporal sequencer* sorts the list of vehicles by their suspicion levels and marks all the vehicles with suspicion levels that are higher than a threshold as bots. The pseudonyms of these vehicles that are likely to be bots are then given to authorities for the appropriate action to be taken towards either recovering their system to its factory settings or removing them from the network.

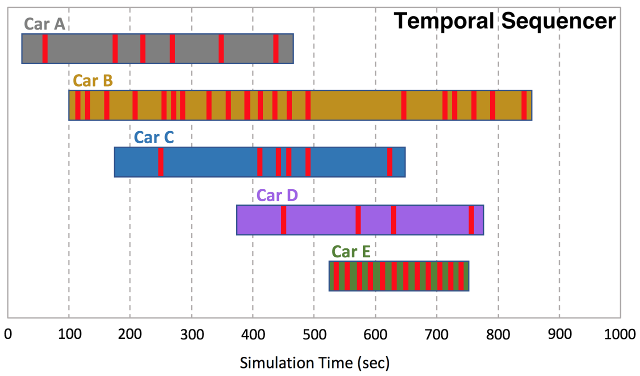


Figure 6. Temporal sequencing of the outliers to calculate suspicion levels

The mechanism for the temporal sequencing of the outliers reported by the machine learning algorithms is depicted with an example in Figure 6. Starting from the first BSM, every broadcast of each vehicle in the *shielded area* is placed in the *temporal sequencer* ordered by its timestamp (different colored bars), which captures both the duration that each vehicle is in the *shielded area* relative to others and the total number of BSMs received from each vehicle. The BSMs with an outlier reported by any of the machine learning algorithms are marked by the *temporal sequencer* (red lines) according to the pseudonyms and timestamps located in the outlier reports. These marked BSMs are then used for calculating the outlier percentage of each vehicle—which is the percentage of the marked BSMs of a vehicle over all of its broadcast BSMs—to be used in the suspicion level calculation.

Outliers might occur for every vehicle due to the false positives caused by the machine learning algorithms. As a result, an innocent vehicle might sometimes have the same outlier percentage as a vehicular bot. However, due to the nature of GHOST and vehicular botnet attacks, injections and the outliers caused by them are likely to be frequent during certain times whereas the outliers for innocent vehicles tend to be more scattered. For example, even though Car A and Car C in Figure 6 seem to have the same outlier percentage, Car C is more likely to be a bot due to this reason. Therefore, the *temporal sequencer* divides the BSM and outlier report data into multiple time intervals (dashed lines) and performs the outlier percentage calculations for each interval separately. If a vehicle's outlier percentage for an interval is above a threshold, the vehicle gets marked as a violator for that interval. The *temporal sequencer* then calculates the suspicion level for a vehicle by finding the percentage of the intervals for which the vehicle is a violator over all the intervals that the vehicle's BSMs span. The *temporal sequencer*'s use of percentages—outlier percentage of a vehicle instead of its total number of outliers and percentage of a vehicle's interval violations instead of the total number of interval violations—normalizes the effect of vehicles being in the *shielded area* for different durations on the resulting suspicion levels. For example, in Figure 6, it enables Car E to be correctly assigned a higher suspicion level than Car B.

After the calculation of the suspicion level for a vehicle, it is immediately added to the *temporal sequencer*'s sorted suspicion level list and authorities are updated with the new list. If the suspicion level of the vehicle is above a threshold, it gets identified as a bot and authorities deal with it alongside the other identified bots in the list. Note that none of the outlier percentage calculation, suspicion level calculation or vehicular bot identification mechanism depends on any specific types or number of machine learning algorithms. This is because, much like the *data collector*, the *temporal sequencer* is also designed in an adaptive manner so that the types and number of the machine learning algorithms used for anomaly detection can easily be changed as long as there is at least one algorithm that outputs outlier reports.

IV. Evaluation

We used Veins [18] (which combines the SUMO and OMNeT simulators) to evaluate SHIELDNET. SUMO is responsible for simulating realistic vehicular traffic while OMNeT simulates the IEEE 802.11p standard [21]. Each simulation was 30 minutes long and a total of 1500 cars passed through the *shielded area*. The percentage of vehicular bots over the total number of cars was 20%.

Since the vulnerability in the VANET standards that makes GHOST infeasible to detect is the excessive fine granularity in the injected fields, we evaluated SHIELDNET using both the BSM traces with the normal granularity that is compliant with the current standards and the traces with the decreased granularity. We decreased the granularity of the injected fields by a magnitude that makes the effect of the injections more noticeable and that we believe does not affect traffic safety; the details are explained in Section V.

We evaluated SHIELDNET separately for each machine learning algorithm, that is, there was only one active machine learning algorithm in each simulation detecting anomalies in the values of the specific injected BSM field(s) it is responsible for. This way, as we observed the capabilities provided by

SHIELDNET, we also evaluated the effectiveness of each machine learning algorithm individually. Each algorithm was tested with both the normal and decreased granularity BSM traces. The accuracy metric is defined as the percentage of vehicles that are actually bots in the list of identified vehicular bots SHIELDNET reports to authorities.

Convolution	Standard Deviation	Sliding Window Size	Interval Length (sec)
Normal Granularity	0.1	25	140
Decreased Granularity	0.5	5	40
3D DBSCAN	Max Cluster Radius (m)	Min Number of Points	Interval Length (sec)
Both Granularities	4	8	100

Figure 7. Optimal configuration parameters of the machine learning algorithms for the normal and decreased granularity

Figure 7 shows the optimal parameter values used in the experiments for the components of SHIELDNET—standard deviation and sliding window size for the convolution, maximum allowed radius for the clusters and minimum number of data points required to form a cluster for the 3D DBSCAN, and interval length used in the *temporal sequencer*. The factors that affect the optimal parameter values are different for each component. The optimal configuration of the convolution is determined by the used granularity since the sensitivity required to detect the outliers caused by injections changes with the granularity. On the other hand, the optimal configuration of the 3D DBSCAN is independent of the granularity but rather governed by the geography and its environmental factors that affect GPS errors and the best way to model them. Finally, the optimal interval length for the *temporal sequencer* is decided based on the frequency of the outlier reports from the machine learning algorithms; the more frequently outliers are reported, the smaller the interval length should be.

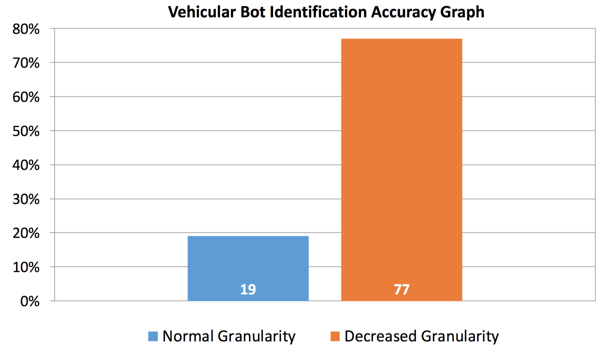


Figure 8. Accuracies of vehicular bot identification using convolution for the normal and decreased granularity

The vehicular bot identification accuracies are shown in Figure 8 for the normal and decreased granularity when only the convolution is used for anomaly detection. The results confirm that, with the normal granularity in the speed field as per the current VANET standards, GHOST remains infeasible to detect since 19% accuracy is just a statistical result of 20% of all cars in the simulation being bots. When the granularity is decreased, the experiments reveal that, with our approach, the time-series speed data indeed follows a sufficiently predictable pattern to detect injections and SHIELDNET can identify the vehicular bots with 77% accuracy.

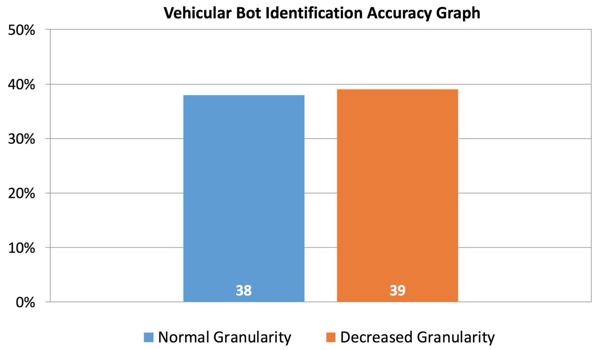


Figure 9. Accuracies of vehicular bot identification using 3D DBSCAN for the normal and decreased granularity

Figure 9 shows the vehicular bot identification accuracies for the normal and decreased granularity when only the 3D DBSCAN is used for anomaly detection. The results substantiate the effectiveness of our approach of clustering GPS errors in 3D based on latitude and longitude; despite the normal granularity, 38% identification accuracy is significantly more than a statistical result of the bot percentage of 20%. On the other hand, decreasing granularity does not improve the accuracy that much due to the low limit of granularity decrease in the latitude and longitude fields that is possible without affecting traffic safety. This issue and an existing vulnerability related to the nature of GPS errors—which adversely affect the accuracy—are explained in Section V.

The accuracy of SHIELDNET is also affected by the values of thresholds used in the *temporal sequencer*—the outlier percentage threshold to determine interval violators and the suspicion level threshold to identify vehicular bots. We currently use a static outlier percentage threshold, which is the same for all intervals, and a static suspicion level threshold. As future work, implementing a mechanism that determines the best threshold values dynamically based on changing network activity will significantly improve the SHIELDNET's vehicular bot identification accuracy.

V. Vulnerabilities in VANET Standards

The significant contributions of SHIELDNET to VANET security need to be complemented with improvements to the VANET standards since GHOST cannot be detected by SHIELDNET with the current fine granularity in the injected fields—each increment is in the unit of 0.02 m/s ($\approx$ 0.04 miles/h) for speed and 10^{-7} degrees for latitude and longitude [21], which are the fields that the anomaly detection is performed on. Therefore, we decreased the granularity of these fields by a factor of 16, which we believe is the minimum granularity that does not affect traffic safety; a further decrease would cause each increment in the speed field to be in the unit of more than $\approx$ 1 mile/h, and each increment in the latitude and longitude fields to represent more than $\approx$ 1 meter in distance. The experiments showed that, while this granularity decrease by a factor of 16 was sufficient to make the injections on the speed field noticeable, the variations in GPS errors were still generally higher than the effects of the injections on the latitude and longitude fields. In other words, GHOST could not be detected without a significant false positive rate by using only the latitude and longitude values. Therefore, the standards should further be improved so that the additional granularity decrease in the latitude and longitude fields would not impact traffic safety (e.g., enforcing an extra safety distance). An obvious attacker response to the granularity decrease in the injected fields would be decreasing the size of the botnet messages to make injections less noticeable. However, this would significantly impact the functionality of GHOST due to the consequent increase in duration for fully synchronizing the information needed for attacks among vehicular bots, and decrease in the number of attacks that could be performed simultaneously, details of which are located in [7].

Another existing vulnerability that affects the success of the anomaly detection in latitude and longitude values is the unpredictability of GPS errors. There are several factors determining these errors such as satellite orbits, inaccuracies in satellite time, atmospheric effects, signal blockage, multi-path effect and radio interference [11]. Randomness in these factors makes the GPS error for an individual vehicle unpredictable at any given time [12]. However, since vehicles close to each other would be subjected to similar environmental conditions, they would have similar GPS errors. Our 3D DBSCAN model exploits this phenomenon effectively to detect the anomalies that are caused by the injections, as aforementioned. While our approach compensates for the randomness in GPS errors considerably, there is still a significant unpredictability in radio interference levels. Even vehicles close to each other might sometimes experience very different radio interference levels due to the changing number, positions and usage pattern of nearby interfering devices, causing a high false positive rate in the 3D DBSCAN. Therefore, as future work, techniques for smoothing random GPS errors [12], prediction models for local radio interference caused by user devices, and/or methods for normalizing the effects of radio interference locally can be built and incorporated into our 3D DBSCAN model in order to decrease the differences in GPS errors of vehicles close to each other. Otherwise, even though SHIELDNET already has an identification accuracy of 77% using only the convolution on speed values, attackers might come up with ways to better utilize the latitude and longitude fields for injection, reducing the use of the speed field and thus decreasing the probability of getting detected.

VI. Conclusion

In this paper, we presented SHIELDNET—an adaptive detection mechanism designed to defend against vehicular botnets. It is a vehicular bot identification framework that employs a set of machine learning algorithms to detect the use of GHOST, a vehicular botnet communication protocol. SHIELDNET is the first implementation of a defense mechanism against vehicular botnets. We used the standard machine learning algorithms that are the most suitable for our anomaly detection approaches to evaluate SHIELDNET. We discussed the effectiveness of these approaches and explained how easily our adaptive framework can be extended with new machine learning algorithms. We showed via experimentation that GHOST remains infeasible to detect if the standard granularity in the injected BSM fields is not decreased. We described how we decreased the granularity of these fields without affecting traffic safety and demonstrated that SHIELDNET can identify 77 percent of the vehicular bots with the decreased granularity. We discussed the vulnerabilities that still exist in the VANET standards and need to be fixed. We recommended several solutions to improve the accuracy of the anomaly detection in GPS errors as future work.

References

[1] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi. Exposure: Finding malicious domains using passive dns analysis. In *NDSS*, 2011.

[2] C. Cho, E. Shin, and D. Song. Inference and analysis of formal models of botnet command and control protocols. In *ACM CCS*, 2010.

[3] P. Choudhary. Introduction to anomaly detection. https://www.datascience.com/blog/python-anomaly-detection, 2017.

[4] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996.

[5] M. Feily, A. Shahrestani, and S. Ramadass. A survey of botnet and botnet detection. In *IEEE SECURWARE*, 2009.

[6] M. T. Garip, M. E. Gursoy, P. Reiher, and M. Gerla. Congestion attacks to autonomous cars using vehicular botnets. In *NDSS*, 2015.

[7] M. T. Garip, P. Reiher, and M. Gerla. Ghost: Concealing vehicular botnet communication in the vanet control channel. In *IEEE IWCMC*, 2016.

[8] M. T. Garip, P. Reiher, and M. Gerla. Botveillance: A vehicular botnet surveillance attack against pseudonymous systems in vanets. In *IFIP WMNC*, 2018.

[9] M. T. Garip, P. Reiher, and M. Gerla. Riot: A rapid exploit delivery mechanism against iot devices using vehicular botnets. In *IEEE VTC*, 2019.

[10] G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *17th USENIX Security Symposium*, 2008.

[11] H. Jamal. Sources of errors in gps and their correction. https://www.aboutcivil.org/sources-of-errors-in-gps.html, 2017.

[12] J. Jun, R. Guensler, and J. Ogle. Smoothing methods to minimize impact of global positioning system random error on travel distance, speed, and acceleration profile estimates. *Transportation Research Record: Journal of the Transportation Research Board*, (1972):141–150, 2006.

[13] S. Kafle. Dbscan. https://github.com/SushantKafle/DBSCAN, 2017.

[14] A. Karasaridis, B. Rexroad, and D. Hoeflin. Wide-scale botnet detection and characterization. In *HotBots*, 2007.

[15] T. Moore, M. Raya, J. Clulow, P. Papadimitratos, R. Anderson, and J. P. Hubaux. Fast exclusion of errant devices from vehicular networks. In *IEEE SECON*, 2008.

[16] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J. P. Hubaux. Eviction of misbehaving and faulty nodes in vehicular networks. *IEEE Journal on Selected Areas in Communications*, 25(8):1557–1568, 2007.

[17] D. A. Rivas, J. M. Barceló-Ordinas, M. G. Zapata, and J. D. Morillo-Pozo. Security on vanets: Privacy, misbehaving nodes, false information and secure data aggregation. *Journal of Network and Computer Applications*, 34(6):1942–1955, 2011.

[18] C. Sommer. Veins: Vehicles in network simulation. http://veins.car2x.org, 2015.

[19] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your botnet is my botnet: Analysis of a botnet takeover. In *ACM CCS*, 2009.

[20] U.S. Department of Transportation. National motor vehicle crash causation survey. *DOT HS 811 059*, 2008.

[21] Wireless LAN Working Group. Wireless access in vehicular environments. *IEEE Standards*, July 2010.