Application Distribution Model and Related Security Attacks in VANET

Navid Nikaein, Soumya Kanti Datta, Irshad Marecar, Christian Bonnet
EURECOM, Sophia Antipolis, France
{navid.nikaein, soumya-kanti.datta, irshad.marecar, christian.bonnet}@eurecom.fr

ABSTRACT

In this paper, we present a model for application distribution and related security attacks in dense vehicular ad hoc networks (VANET) and sparse VANET which forms a delay tolerant network (DTN). We study the vulnerabilities of VANET to evaluate the attack scenarios and introduce a new attacker's model as an extension to the work done in [6]. Then a VANET model has been proposed that supports the application distribution through proxy app stores on top of mobile platforms installed in vehicles. The steps of application distribution have been studied in detail. We have identified key attacks (e.g. malware, spamming & phishing, software attack and threat to location privacy) for dense VANET and two attack scenarios for sparse VANET. It has been shown that attacks can be launched by distributing malicious applications and injecting malicious codes to On Board Unit (OBU) by exploiting OBU software security holes. Consequences of such security attacks have been described. Finally, countermeasures including the concepts of sandbox have also been presented in depth.

Keywords: VANET, Application distribution, Mobile platform, Malware, Sandbox, App store, Attacker's Model.

1. INTRODUCTION

Intelligent Traffic System (ITS) promises reduced traffic congestion and improved traffic safety [1], [2]. While considering the numerous benefits of VANET, we also have to investigate the vulnerabilities leading to security attacks both dense and sparse VANETs. Main vulnerabilities in dense VANET stems from the network properties and associated protocols including (i) lack of fixed infrastructure to take care of security and privacy issues, (ii) Wireless medium (e.g. allows eavesdropping), (iii) node mobility (frequent change of topology), (iv) lack of sequence number or time stamp in basic 802.11 MAC layer (allows replay attack) and (iv) multi hop routing (e.g. routing attacks). On the other hand, sparse VANET which forms DTN in rural areas suffers from vulnerabilities like (i) long round trip delay, (ii) no end-to-end connectivity and (iii) store, carry for a while and forward nature. All these vulnerabilities account for message modification and packet inspection. Significant evaluation of security attacks [5] and development of countermeasures are very crucial when it comes to successful deployment of VANET and ITS. As an example, it is absolutely necessary to make sure that nobody can insert or modify emergency messages into the network (message tampering).

Recent studies in vehicular communication suggest more sophisticated network structure. It is also being envisioned that the on board units (OBU) will integrate emerging mobile platform(s) like Android and iOS in future and will be able to download and install applications from app stores. Today applications are being provided by trusted app stores of Apple and Google or property platform by car manufacturers like BMW. This scenario could change and applications can be hosted by a third party provider. For example, in case of Android apps the in-app capability can be enabled for a third party. Although content distribution in VANET [3], [4] has been investigated in literature, the concept of application distribution and associated threats have not yet been looked into in depth. By the term 'content' we refer to multimedia material that cannot interact with the mobile platform whereas 'application' actively does. The possibility of integrating mobile platforms in OBU announces technological advancements but the possibilities of sophisticated attacks also exist latently. The main motivation behind this work is to propose a VANET model that supports application distribution and identify the attacks that can be launched through application distributions. To achieve the objectives, we propose a VANET model that supports application distribution scenarios in both dense and sparse VANETs; state the assumptions and the steps in the application downloading process. Then we identify the attack scenarios and countermeasures are also mentioned in details.

The rest of the paper is organized as follows. Section II briefs attacker's model [6] and extends it by introducing another model. The proposed VANET model with underlying assumptions and steps encountered for application downloading are described in section III. Section IV and V illustrates the attacks in both dense & sparse VANET and the countermeasures that can be adopted to mitigate the attacks. Finally section VI concludes with some future directions.

2. ATTACKER'S MODEL

The attacker's model presented in [6] is briefed below.

- **Insider vs. Outsider**: An ‘Insider’ attacker possesses a public key certificate and has overall knowledge of the entire network. On the contrary, an ‘Outsider’ or intruder to a network has limited amount of attack opportunity.
- **Malicious vs. Rational**: Malicious attacker concentrates on affecting the network whereas rational attacker is focused on personal gain like obtaining banking details etc.
- **Active vs. Passive**: A passive attacker just eavesdrops or analyzes the traffic whereas the active counterpart can carry out more damage to the network by modifying or destroying the contents of packets intentionally.
- **Local vs. Global**: An attacker can be local or extended in scope.

Now we introduce a new attacker’s model by considering the fact that whether the attacker is real or in the absence of an attacker, someone perceives some phenomenon as attack.

- **Real vs. Illusionary**: We consider the fact that an attacker can be real or illusionary. In case of active attacks, there exists a real attacker who performs such attacks. But sometimes the notion of an attack depends largely on the perception. Even in absence of an attacker, some behavior of wireless links or vehicles may appear as an attack to the vehicle. For example, due to poor routing in a sparse network, it may happen that a node may not receive entire response message from the node with whom it is communicating. But to OBU of vehicle, it might appear as a routing attack where intermediate nodes are selectively dropping packets. The behavior of the vehicle and its driver will largely depend on the perception of the situation. In this case, there is no real attacker. Another case could be the active advertisements of commercial companies which may appear as spam and consequently they are filtered out.

3. VANET MODEL FOR APPLICATION DISTRIBUTION

Figure 1 portrays the VANET model which divides the entire VANET into four different domains. VANET Domain V2V includes the mobile nodes (vehicles/humans) and communications among vehicles. The VANET Heterogeneous Access Domain integrates heterogeneous wireless access networks with IP based core network and takes care of network selection strategy, load balancing and communications among vehicles and infrastructure. VANET Core Network Domain acts as a bridge between gateways and IP Exchange (IPX)/IP Backbone that interacts with the server cloud residing in VANET Application Server domain. This model allows proxy app stores to act as proxies of app store servers that are located in the cloud or the third party servers. The realistic feature is included to (i) support application distribution in a sparse VANET where connection could not be established to the app stores or cloud servers and (ii) reduce VANET traffic considerably by avoiding connection to app store for every application download request. The entire Figure 1 represents dense VANET whereas sparse VANET is represented by the same model without the shaded part i.e. by vehicles and few RSUs only. The proposed VANET model can very well support the emergency situations and reduce traffic congestion. But in this work, we focus on the application distribution scenario and the security threats associated with it.

3.1 Assumptions of the VANET Model

Following are the assumptions of the proposed VANET Model.

1. Vehicles and RSUs have limited local storage where the cloud servers/app stores have unlimited storage.
2. All the communications are routed primarily through RSUs. In case the wireless link among RSUs and vehicles are down for any reason, the communication is done through 4G/WIMAX.
3. The servers are synchronized with GPS geo-synchronized timing.
4. The vehicles have GPS positioning to provide the location co-ordinates.
5. The routings for V2V, V2I and I2V communications are based on geo-position routing [7]. Standard routing protocols (wired or wireless) are adopted between RSUs, different app stores & cloud servers. The RSUs perform message switching between different routing protocols.
6. The cloud servers, the third party servers and potentially proxy app stores should have certificates provided by a trusted certification authority (CA).
7. The proxy app stores could exist in all domains and include vehicles, RSUs and gateways.
8. The request to download an application contains the following fields – unique MAC address of the OBU, location co-ordinates, and direction of movement, speed, a time stamp and the application name.

3.2 Steps in application distribution

Here we provide an overview of the procedure of application distribution and put forward two possible scenarios. Firstly, a vehicle sends a ‘GET’ request to download a certain application (e.g. an Android/iPhone application), or update of OBU OS/software and the proxy app stores or cloud servers or app stores grant the download permission to the vehicle. Secondly, car manufactures or 3rd party providers ‘PUSH’ important updates to vehicles (auto update). The application download requests are primarily routed through the RSUs for both dense and sparse VANETs. But in case of very sparse network, the request is answered only by a vehicle within the same network. It may also happen that even in a dense VANET, a vehicle tries to avoid downloading applications from clouds because (i) sometimes clouds keep record of users and requested applications to develop marketing solutions, (ii) to obtain a cracked version of paid applications from proxy app stores and (iii) to avoid the cost of communication. The application download process can be summed up as below.

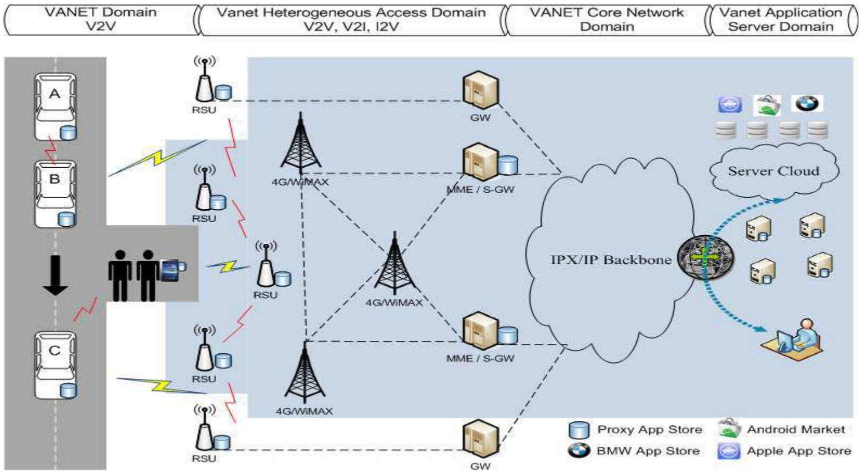

Figure 1. VANET Model.

1. A vehicle requests to download an application directed to RSUs.
2. A RSU that receives the message, checks if it has the desired application in proxy app store. If it is found, the RSU routes the application to the vehicle. Else, the request is forwarded to cloud servers and app stores which on retrieving the application, routes it back to the vehicle.
3. If an application is not found in clouds or app stores or request fails due to timeout, the vehicle is notified about the same, and then it broadcasts the request again.
4. Any proxy app store including vehicles or RSUs could potentially respond to the request.
5. The requestor will select the appropriate proxy.

4. ATTACKS THROUGH APPLICATION DISTRIBUTION IN VANET

In this section we describe the possible attacks through application distribution. The attacks like DoS, replay attack, Sybil Attack and routing attacks has been addressed in the existing literature [11], [13]. But we intend to focus on attacks which are more specific to the scenario of app distribution.

4.1 Malware

Distribution of malwares disguised as a useful application is a sophisticated attack in VANET. Publishing malwares in app stores may be easy if the apps store has no strict screening policy for application publishing. It is even more difficult to extend such a screening policy for proxy app stores. Thus it becomes easier for an adversary to publish malwares. Following describes the possible security attacks through application distribution.

- Attacks like Denial of Service, Illusion attack can be performed remotely. Malwares can compromise proxy app stores to remotely launch routing attacks. Update attacks could be launched by adding malicious codes to the previously published useful apps and releasing the malware to app stores.
- Spywares can analyze the packets transmitted and received and gather user specific information in order to model user behavior. Then attackers can remotely launch specific attacks to modify the user behaviors.
- Another interesting case is the generation and transmission of false warning messages in order to launch a ‘false information attack’.
- Since it is assumed that the routing system is geo-position routing, the vehicles must provide its location co-ordinates for application download. But in case of compromised OBU, the malware is capable of providing false co-ordinates such that the vehicle never receives any application while downloading.
- An intelligent application can track a particular user by receiving and actively analyzing the location coordinates that the user is broadcasting. This leads to a major threat to location privacy.

- Application download process is vulnerable to session hijacking in which an adversary gets unauthorized access to the download process.
- If the malware is able to spoof MAC address of a vehicle, then it can send an application download request with MAC address and location co-ordinates of another vehicle. In this case the other vehicle (that might be controlled by the attacker) gets the application. Such attack leads to disclosure of user sensitive information.
- A malware free application could exploit the in-app capability of mobile platforms and remotely connect to a compromised server in order to install a malicious plug-in.

4.2 Software attacks

Software coding flaws like buffer overflow and incorrect input injection can lead to software based attacks which can lead to injection of malicious codes into the operating system of OBU. Thus an attacker can control the OBU remotely and cause unexpected program behaviors. As consequences of such attacks the update functionality of mobile OS or software update can be disabled to avoid detection or Trojans can be downloaded or DoS attacks can be launched by continuously transmitting noise. According to [8], now-a-days the OBU contains 100MB of embedded codes. It is quite expensive and difficult to test the entire code. Thus if there are any security holes present, that can be reverse engineered and exploited.

4.3 Spamming and Phishing attacks

As mentioned previously, spamming can be performed using a malware from a compromised OBU. It can also be done by an adversary who keeps on sending spams from his own vehicle or by rouge RSUs. Spamming is difficult to control as VANET lacks central administration due to lack of infrastructure. Rouge RSUs can send spams to vehicles elevating the risk of increased transmission latency. Once an OBU is compromised, it can start spamming the vehicles in the neighborhood with adware, false message, spyware and viruses. The attacker can impersonate an RSU and trick other drivers to reveal their personal information by phishing attack. In this case, the attacker broadcasts a message informing the recipients they have been chosen for some offers (like free download of some applications or software updates). In order to avail those opportunity, the user must share some personal information and thus the attacker can gather personal details.

4.4 Malicious intermediate node

This scenario brings out a sophisticated attack for sparse VANET. We assume that there exists at least one malicious node between the node requesting a download and the app store. The intermediate node checks proxy app store for the application on receiving download request. It forwards the application to the sender if available. Else the node stores and carries forward the request until it can retransmit the request to another proxy app store. It can provide the source with a malicious version of the requested application to compromise it. In case a node wishes to buy an application, the malicious node carries payment information to the app store. During that time it can actually perform Deep Packet Inspection to understand the nature of payment process and payment details like credit card numbers.

4.5 Malicious source node

Another attack scenario specific to sparse VANET considers that the source node is malicious which tries to compromise the intermediate node(s) by pushing malwares. The malwares activate themselves once stored in local storage and try to compromise the OBU by software attacks. Once succeeded, such attack can be propagated over an entire sparse VANET to compromise most of the nodes. Another consequence of such attack could be launching a combination of attacks mentioned previously.

5. COUNTERMEASURES

Efforts have already been invested to research the countermeasures of security attacks [9], [14]. Conventional security measures include authentication, use of digital signatures and access control policies. Intrusion detection systems are also being deployed in VANET but only as second base of defense. Here we will discuss the main countermeasures that can be effectively deployed against the mentioned attacks.

5.1 Antivirus

The cloud servers and (proxy) app stores must scan all the applications using updated antivirus softwares before publishing to make sure that the stored applications are malware free. If any malware is found, that should be pulled out of the storage immediately and a notification message should be sent to other (proxy) app stores and cloud servers.

5.2 Permission based approach

Before installing an android application, all the required permissions must be checked carefully. Any application asking ‘out-of-context’ permission(s) should be avoided. For example, an application that shows map of a city should not normally ask privileges to receive and send messages; wipe all data of OBU.

5.3 Sandbox

Sandbox approach can identify potential malwares in applications and can exist in all OBUs. In this approach, an execution environment is created that mimics the target mobile platforms and the suspicious applications are executed. Then their behaviors are studied and validated accordingly.

5.4 Testing of OBU OS and softwares

To protect against software attacks, the OBU operating systems and softwares must be tested while designing and developing for possible bugs. Later, if any bug is found, updates must be announced for downloading through authenticated app stores/servers.

5.5 Tamper proof OBU

It can be useful to counter software attacks, the integrity of the system should be checked and the system should at least be tamper evident if not tamper resistant.

5.6 Defense against session hijacking

SSL/TLS can be used to protect from session hijacking but SSL/TLS require high computation, thereby increasing the computational overhead of vehicles. Thus proper countermeasure against session hijacking remains an open research topic.

6. DISCUSSION

In this paper we have contributed a new attacker's model to [6] and demonstrated new attacks that can find their way as consequences of application distribution in VANET and DTN. Another significant contribution lies in the proposed VANET model that can support the application distribution. It is implicit that successful deployment of the proposed application distribution scenario depends to a lot extent on security. We believe that more researches are needed to identify new security threats and propose better countermeasures. Without a tamper proof OBU and authentication schemes, no classical security mechanism can protect vehicular communication for misbehaving or selfish nodes. Some security issues like countermeasures of session hijacking, location privacy [12] remain open research topics. Our current research is pin pointed on (i) secure application distribution and (ii) design and development of security architecture of the proposed VANET model.

ACKNOWLEDGE,EMT

This work is funded by French Research project WL-BOX-4G Pole SCS.

REFERENCES

[1] P.P. Ashtankar, S.S. Dorle, M.B. Chakole, A.G. Keskar, “Approach to avoid collision between two vehicles in intelligent transportation system,” in 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET), 2009, pp. 598–603.

[2] Ping Li, Li-Min Jia, A-Xin Nie, “Study on Railway Intelligent Transportation System Architecture,” Proceedings of Intelligent Transportation System, 2003, vol 2, pp. 1478—1481.

[3] F. Soldo, C. Casetti, C.F. Chiasserini, P. Chaparro, “Streaming Media Distribution in VANETs,” IEEE GLOBECOM, 2008, pp. 1 – 6.

[4] M. Sardari, F. Hendessi, F. Fekri, “Infocast: A New Paradigm for Collaborative Content Distribution from Roadside Units to Vehicular Networks,” 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09, pp. 1 – 9.

[5] M. Raya, P. Papadimitratos, J.P. Hubaux, “Securing Vehicular Communications,” In IEEE Communications Society, Vol. 13, Issue 5, pages 8 – 15, 2006.

[6] Maxim Raya and Jean-Pierre Hubaux, “Securing vehicular ad hoc networks,” Journal of Computer Security 15 (2007) pp. 39–68.

[7] M. Mauve, A. Widmer, H. Hartenstein, “A survey on position-based routing in mobile ad hoc networks,” In IEEE Network, Vol. 15, Issue 6, 2001.

[8] Manfred Broy, “Challenges in Automotive Software Engineering,” Proceedings of the 28th International Conference on Software Engineering, Pages 32–42. ACM Press, 2006.

[9] Y. Hu, D. Johnson, and A. Perrig, SEAD: Secure Efficient Distance Vector Routing in Mobile Wireless Ad-Hoc Networks. Proc. of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02), pp. 3-13, 2002.

[10] Jeremy J. Blum, Andrew Neiswender, and Azim Eskandarian, Denial of Service Attacks on Inter-Vehicle Communication Networks, 11th International IEEE Conference on Intelligent Transportation Systems Beijing, China, October 12-15, 2008

[11] S. Carter and A. Yasinsac, “Secure position aided ad hoc routing protocol”. Proc. Of the LASTED International Conference on Communications and Computer Networks (CCN02) pp. 329 -334, 2002.

[12] N.W. Lo, H.C. Tsai, “Illusion Attack on VANET Applications – A Message Plausibility Problem,” In IEEE Globecom Workshops, 2007, pp.1–8.

[13] J. Douceur, “The Sybil Attack,” In First International Workshop on Peer-to-Peer Systems, pages 251–260, 2002.

[14] S.M. Safi, A. Movaghar, M. Mohammadizadeh, “A novel approach for avoiding wormhole attack in VANET,” In Second International Workshop on Computer Science and Engineering, 2009, pp. 160—165.