Security Modeling and Analysis on Intra Vehicular Network

Jinli Zhong*, Suguo Du*, Lu Zhou*, Haojin Zhu*, Fan Cheng*, Cailian Chen* and Qingshui Xue†
*Shanghai Jiao Tong University, 200240 Shanghai, China
{Yesterday-once-more, sgdu, zhoulu}@sjtu.edu.cn, zhuhaojin@gmail.com,
{chengfan, cailianchen}@sjtu.edu.cn
†Shanghai Institute of Technology, 201418 Shanghai, China
xue-qsh@sit.edu.cn

***Abstract*—Controller Area Network (CAN), the *de facto* standard in-vehicle network protocol, prompts modern automobile an integrated system that achieves real-time interactions with roads, vehicles and people. Yet such connectivity makes it feasible to illegally access, or even attack the CAN, causing not only privacy disclosure, property damage, but also life threat. In this paper, we analyze intrinsic weakness in CAN protocol that is mostly exploited by attackers and comprehensively survey the existing attacks based on CAN interfaces. Furthermore, we propose an attack evaluation system based on attack tree model and Markov chain to assess the probability of compromising CAN and the steady state of CAN system at the presence of these attacks. Finally, we simulate new steady state when altering the difficulty of a certain attack and the results demonstrate that sometimes improving defense of an attack declines the security level of the entire system instead.**

I. Introduction

In modern automobile, CAN, a standard vehicular network protocol, is of significant importance since it is not only widely utilized by Volkswagen, Mercedes-Benz and other professional car manufacturers, but mostly applied to mainstream powertrain communications, such as controlling engine, steering, braking and so forth. Specifically, contemporary vehicle relies heavily on Electronic Control Unit (ECU) to transform informational signal obtained from sensor into actual movement done by actuator. And it is CAN that connects ECUs so that data frame, control signal, and error message could be efficiently transmitted.

However, several inherent flaws exist in CAN protocol, namely, lack of encryption, ineffective authentication, and broadcast communication, since originally it was designed to be isolated from external network[1]. But with various services and entertainment systems attached to vehicles, CAN is no longer untouchable. For instance, the majority of modern cars are equipped with OBD-II (On-Board Diagnostic) interface under the dashboard, which is mandatorily required in US[2]. Moreover, a concept of treating automobile as open-source platform for third-party applications has been proposed. Thus, the threshold of accessing CAN is lowered to a great extent that it is totally feasible to achieve attack, let alone unauthorized access. For instance, researchers remotely cut the engine of Tesla Model S, or even killed Jeep Cheroke[3] on the highway. Unlike normal internet attacks, not only property, but life is threatened. Therefore, we analyze weakness that attackers usually exploit and summarize existing attacks based on CAN interfaces as the foundation of evaluation.

Much attention has already been paid on modeling and evaluating the attacks on CAN for better defense. Lotfi Ben Othmane[4] designed own standard to qualitatively evaluate attacks such as security of devices and security of links, but it is only theoretical and empirical. Alexandros Asvestopoulos[5] introduced attack tree model to quantitatively assess security issues, but such model is inherently static and limited to independent events while attacks tend to be dynamic processes.

In this paper, we propose an innovative evaluation system based on attack tree model and Markov chain. The attack tree describes the steps of attacking CAN, the logic relationship among them, as well as the probability of compromising the CAN. However, such model could not reflect the order of attack methods, thus we further introduce Markov chain to make it dynamic considering corresponding defense as well. Furthermore, we simulate the steady state of CAN in the presence of current attacks and the influence of attack difficulty. Thus, the main contributions of the paper are as follows:

- A correspondingly comprehensive summary of existing attacks on CAN based on CAN interfaces.
- A novel attack evaluation system of CAN based on attack tree model and Markov chain.
- A simulation about the influence of each attack on the steady state of CAN system.

The rest of the paper is organized as follows. Section 2 details the CAN protocol and describes intrinsic defeats. Section 3 summarizes the existing attacks based on CAN interfaces. Section 4 employs attack tree model and Markov chain to evaluate the security issue of CAN. Finally Section 5 gives the conclusion.

II. Primer on CAN

A. Format of CAN Frame

Generally, there are 4 types of CAN frames, namely data frame, remote frame, error frame, and overload frame. Each has its own functionality, but basically shares the same format. Take data frame for example. It can further be divided

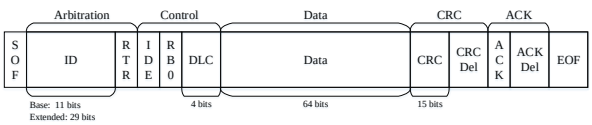

Fig. 1. Format of CAN Data Frame

into two types, the base one and the extended one, where the only criteria is the length of identifier. The base frame contains 11-bit identifier and the extended contains 29 bits as Fig.1 demonstrates. The identifier, which is unique arranged, represents the priority of message, and the 4-bit Data Length Code (DLC) exhibits the length of valid data in data field in bytes. The CRC field is the fundamental protection based on cyclic redundancy check, which is effective, but surely is not enough to resist various attacks.

B. Inherent Weakness

The format of CAN frame reveals that the communication is not secure since it is not encrypted and barely has authentication. It makes sense since real-time interaction rather than security is of vital importance in this life-critical application[6]. Besides, broadcast interaction and priority transmission are the other two common characteristics attackers usually exploit.

1) Broadcast Interaction: Physically and logically, messages on the CAN are broadcasted to all nodes, and whether to receive the message is determined by the ECU based on the identifier. Therefore, malicious components are totally capable to monitor all the information on the bus and could send data satisfying the format to every node.

2) Priority Transmission: Once a node starts sending data, it monitors CAN bus to determine whether to continue or transform into a receiver, which is called arbitration and is entirely decided by the identifier[7]. Normally, messages from different nodes would never share the same identifier, and the smaller the identifier is, the higher priority the message will obtain. Thus, it is of high probability to achieve the Denial of Service (DoS) attack, since all that the attackers need to do is constantly sending frames with small identifiers.

3) Lacking Authentication: The identity of the sender is not required in this protocol and any formatted frame could be transmitted to every node due to broadcast nature. It means that any compromised node could indiscriminately control all the other nodes through the bus, since these nodes themselves have nothing associated with defense mechanisms.

III. CAN Interfaces Exploited by Attacks

As mentioned earlier, intrinsic weakness and connectivity to the external network provide various interfaces to access the CAN, and we summarize those regularly exploited by the attackers as follows.

A. OBD-II Interface

Based on whether the path-through device[5] connected to OBD-II is wired or not, such attack could be divided into two types. To hack the wireless kind, attackers need to install malicious Access Point (AP) whose strength is far beyond the default one. Thus the device would be more willing to connect malicious AP and attackers could sniff and intercept the data on bus. In view that the cost of installing AP is reasonable and the attacker could protect himself from being discovered, the success probability is thus considered moderate.

Things change for the wired one that physical connection is required in this case, then the software, the hardware, even the attacker himself might be exposed. Only bribing or deceiving staffs in workshop could achieve the attack, thus the possibility of success seems relatively low.

B. Bluetooth System

Bluetooth system is made connected to CAN for remote control and there are basically two types of such attacks. If the attacker obtains a device paired with Bluetooth system, what he needs is to design a delicate application and deceive the owner to download and install. For instance, Stephen Checkoway installed a Trojan application on HTC Dream (G1) and once the phone connects to car's Bluetooth system, it scans for other Bluetooth connections and would send malicious packets on CAN[8]. Considering the existence of malware in the market and the attraction of deceptive advertisements, these two events seem quite feasible. However, since different manufacturers design different mechanisms in details, it is unpredictable whether the malware could successfully achieve the attack, which brings the toughness.

If lacking such device, the only method would be pairing with the car via attacker's own equipment exploiting system vulnerability. Either analyzing corresponding documents, source code or reverse engineering the Bluetooth system may discover some flaws. The former involves disclosure of commercial confidentiality which is rather difficult, while the latter is a general reverse problem such as utilizing "Bluesniff" to obtain the MAC address of Bluetooth and brute force cracking PIN[9]. Once it is connected, the situation is exactly the same as the previous one that the pre-designed application would undertake the task left.

C. Tire Pressure Monitor System (TPMS)

Drawing on researches of Ishtiaq Rouf[10] and Stephen Checkoway[8], attack on TPMS requires a customized antenna to send TPMS packet and a compromised ECU to detect specific packet. The former is extremely simple, using USRP could complete the task, yet the latter requires skills of reverse engineering to refresh or modify the firmware to trigger pre-complied data. Under this circumstance, the success rate of changing firmware seems relatively low.

D. Media Player System

Two categories exist based on whether physical insertion is demanded. For those who connect the CAN through CD, USB, or iPod, attackers would implement either document analysis or reverse engineering to find vulnerabilities and deceive owners to play the pre-designed media file as that to

the Bluetooth system. While for wireless ones that rely upon analog FM, a low power transmitter to transmit the signal with a compromised ECU is required, which is basically similar to that of TPMS.

E. CAN to GSM Adapter

According to a survey by Scania[5], only nearly one tenth of the vehicles use third-party CAN to GSM (Global System for Mobile Communication) adapters, which implies that such attack would not be exceedingly common. However, the success rate is extremely high since several lines of scripts written by personal computer could attack the GSM[11].

IV. Evaluation System

In the presence of these five attack patterns, we now propose a novel evaluation system based on attack tree model and Markov chain to assess the probability of compromising CAN, the property of the steady state, and the influence on CAN when altering the difficulty of a certain attack.

A. Evaluation Based on Attack Tree Model

The attack tree model was first proposed by Bruce Schneier[12] for describing attacks. A tree structure is used to visualize all possible steps to accomplish an attack, thus could simulate how each method is implemented to achieve the attack and find which method causes the severest damage. Moreover, logic gates (normally OR and AND gates) are utilized to demonstrate the relationship among steps. Furthermore, once a tree is established, each basic event, namely the leaf node, would be assigned a corresponding value, representing the success rate of event, and the root value would represent the security level of the whole system.

Thanks to the merits above, we now focus on how to establish the attack tree. In this case, the top event would be attacking CAN since it is the ultimate goal. Then the five patterns in Section III constitute the sub-goals, thus are described as sub-nodes. And since any pattern is capable to accomplish the attack, they are connected by OR gate. Furthermore, for instance, refreshing firmware and customizing signal generator constitute essential methods to attack TPMS, which then become leaf nodes and the indispensability guarantees connection by AND node. Finally, in a top-down manner the tree is created effectively.

TABLE I
Approximation of Probability of Success

Description	Range	Approximation
Low Probability	$0 \leq p < 0.20$	0.10
Relatively Low Probability	$0.20 \leq p < 0.40$	0.30
Medium Probability	$0.40 \leq p < 0.60$	0.50
Relatively High Probability	$0.60 \leq p < 0.80$	0.70
High Probability	$0.80 \leq p \leq 1$	0.90

In view that the model itself is more significant than particular value in this paper, for convenience, we empirically divide the success probability of attacks into five categories as shown in TABLE I. According to the narratives in the previous section, every leaf node is assigned a corresponding approximated value as success rate. Then in a bottom-up pattern, with the assistance of general rules of AND and OR calculations, the probability of compromising CAN could be obtained as Fig.2 demonstrates. Moreover, the root, the sub-node, and the leaf node are separately labeled "t", "$m_i(i = 1, 2, \cdots, 11)$", and "$b_j(j = 1, 2, \cdots, 11)$" for precise description.

B. Evaluation Based on Markov Chain

Originally, Markov chain describes transition property of random processes. For random variables $\{X_n, n = 0, 1, 2, \cdots\}$ and its state space $S = \{S_i, i = 0, 1, 2, \cdots\}$, $X_n = S_i$ represents staying S_i at time n. The theory shows that the next state depends completely on the current state, and is unrelated to any of the previous ones. Suppose $|S| = m$, and P represents the transition matrix where $P_{ij}(i \neq j)$ describes the transition probability from S_i to S_j and P_{ii} shows the probability of maintaining S_i.

The reasons why we introduce Markov chain in this evaluation system are listed as follows. Firstly, not only attacks in the attack tree, but corresponding defenses could be demonstrated by Markov chain. Secondly, it reveals the logical and practical order of basic events compensating the limitation that every event in attack tree is treated independent. For instance, the attacker must design malicious application before deceiving victims to install. Thirdly, time dimension is taken into account while attack tree is just static. And thanks to the random choices of attackers and defenders, the next state of CAN is entirely decided by the current state and however the system reaches the current state has nothing to do with the future, perfectly consistent with Markov chain. Lastly, with a proper transition matrix, the transition of states could be systematically described and the risk of CAN system could be quantitatively assessed.

1) Transition Diagram: Once the suitability of Markov chain is determined, we propose three assumptions to simplify the generation of transition diagram, which demonstrates transitions among the states. Firstly, we assume that every basic event in the attack tree has its corresponding defense as a basic event, since these attacks maintain the integrity of CAN's hardware and could thus be recovered. Secondly, we suppose that the success rate of basic event is independent of each other that whatever the current state is, the transition probability would be the same. Otherwise, we could utilize piecewise function to describe the difference since the state space is finite. Lastly, without loss of generality, we consider state transition caused by only one basic event. Any other transition could be expressed with this method. For instance, transition caused by two events could equally be described as two adjacent transitions with the same probability.

On the basis of these assumptions, we describe systematic state as a combination constituted by the state of each basic event. And it is divided into three categories, namely failed state, semi-failed state and normal state. The original situation that CAN works properly is called normal state. And once the top event happens, CAN reaches failed state. Otherwise,

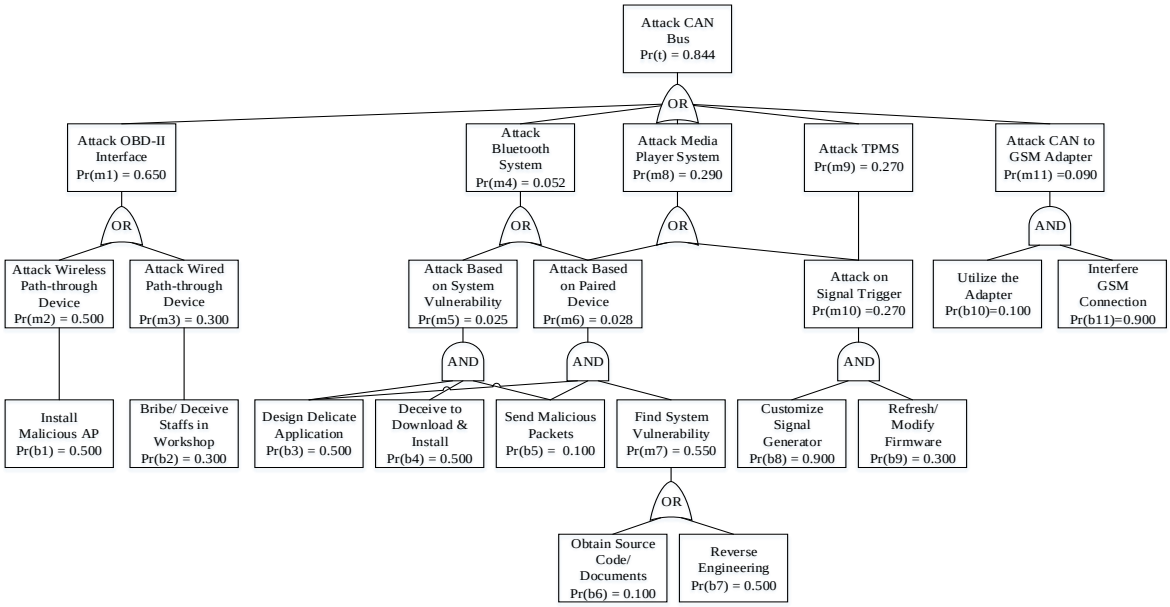

Fig. 2. Attack Tree of CAN

it is semi-failed state. Finally, considering the order of basic attack methods, that is $b_3 \rightarrow b_4 \rightarrow b_5$, $(b_7|b_8) \rightarrow b_3 \rightarrow b_5$, $b_{10} \rightarrow b_{11}$, the transition diagram is established as Fig.3.

2) Steady State: Intuitively, the transition matrix could be derived from the diagram based on another two assumptions. Firstly, we consider the fact that the more effective the attack is, the more useless the defense it implies. Otherwise, the defense mechanism would protect the system from being attacked to a great extent and under no circumstance will the attack be called effective. Thus, to simplify the calculation, we claim that the sum of success rate of any attack b and its corresponding defense B would be 1. That is

$$Prob_{attack_b} + Prob_{defense_B} = 1 \tag{1}$$

Secondly, we assume that the higher the success rate of an attack is, the higher transition probability related to the attack will be. So does the defense. Suppose the current state is S_i, and Q_i represents all the next states the system may transform to, then the transform frequency F_{ij} and the transform probability P_{ij} would be

$$F_{ij} = \begin{cases} Prob_{attack_b} & S_i \xrightarrow{b} S_j \\ Prob_{defense_B} & S_i \xrightarrow{B} S_j \end{cases} \tag{2}$$

$$P_{ij} = \frac{F_{ij}}{\sum_{k \in Q_i} F_{ik}} \tag{3}$$

The generated matrix perfectly satisfies the requirement of Markov matrix that every element is no less than 0 and that the sum of each row or each column is equal to 1, since the success rate of basic event could never be less than 0 and P_{ij} is generated via frequency normalization.

Suppose vector π describes the probability of each state when CAN reaches the balance, then π must satisfy the following prerequisite.

$$\lim_{n \rightarrow \infty} \pi \cdot P^n = \pi \tag{4}$$

Utilizing the platform of MATLAB 2015b, we find all the absolute value of P's eigenvalues are no greater than 1, guaranteeing that a steady state do exist[13]. Otherwise, the value of π would fluctuate dramatically when n changes.

C. Evaluation Results

The original probabilities are estimated and calculations involve approximations. Thus it is the range where the value is in and the trend of change rather than the specific value itself that makes more sense.

1) Probability of Compromising CAN: Generally, the risk level of CAN could be assessed by either attack tree model or Markov chain. In attack tree model, the probability of achieving top event is 0.844 while in the steady state, proportions of normal state, semi-failed states and failed states are separately 0.1641, 0.5217 and 0.3142. Therefore, both reveal that the probability to maintain working properly is relatively low in the presence of existing attacks. Nevertheless, the compromising possibility is far less in Markov chain, since the attack tree neglects the order of attack methods so that states violating such order are still included.

2) Dominant Attack Patterns: In attack tree model, attacking OBD-II has the highest probability of 0.65. Especially, compromising wireless path-through device to interfere normal communication becomes the dominant pattern due to its low cost and simple operation. Whilst attacking Bluetooth system accounts for less than 10% of the former due to its complicated process. While the steady state describes relatively high

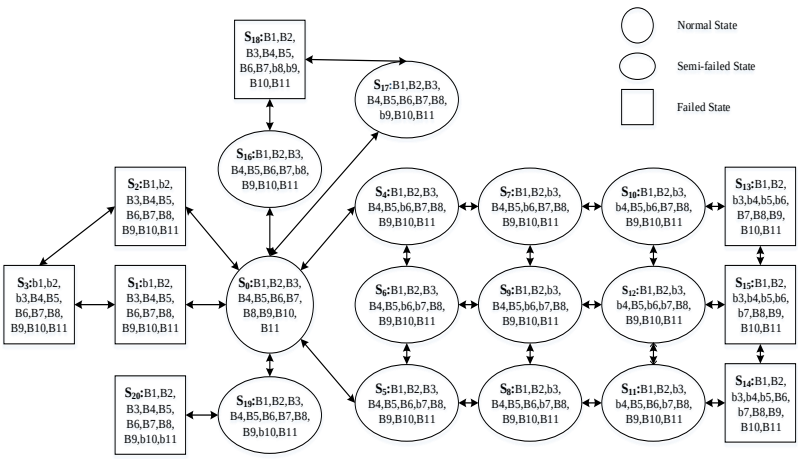

Fig. 3. Transition Diagram of CAN State

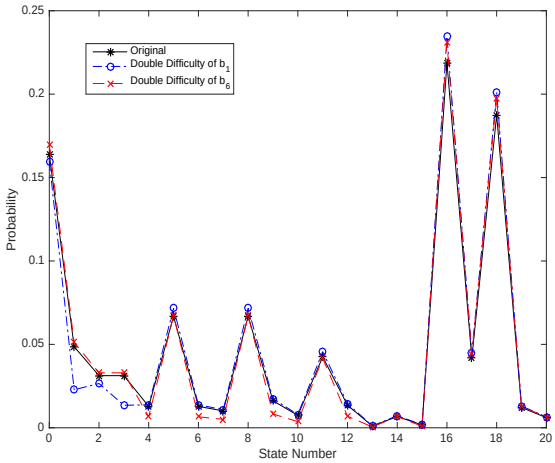

Fig. 4. Comparison on Steady State

proportion of S_{16} and S_{18}, indicating that attacks combining firmware refresh and signal generator production are most aggressive, which makes sense as well. For instance, Miller and Valasek[3] have done severe damage through rewriting firmware, otherwise, recalling cars would not be the only choice for car manufacturers.

3) Influence on Steady State: Generally, the difficulty of attack is dynamic for the development of attacker, defender, or the both, leading the system to a new balanced state. Thus we manually alter the success rate of a certain attack to simulate its influence on steady state. For instance, we separately decrease the success probability of b_1 and b_6 by 50%, and Fig.4 shows the comparison.

Every state is impacted in equilibrium, not only those directly related to the changed attack. Nonetheless, it depends whether the influence is positive even when certain attack becomes more difficult to achieve. Specifically, probability of normal state increases to 0.1698 when b_6 becomes twice harder whilst it decreases to 0.1597 when b_1 changes. The fact is that after a number of attempts, rational attacker would master the difficulty level of each attack and would tend to choose the relatively simple one, making the CAN system suffer greater risk instead. Therefore, simply improving the defense of a certain attack may increase the risk of the whole system, overall protection is demanded.

V. Conclusion

In this paper, we analyze inherent weakness in CAN protocol under the trend of connecting CAN with external network, which leads to the existing attack patterns summarized on CAN interfaces. Furthermore, we propose an innovative evaluation system based on attack tree model and Markov chain. The simulation describes the most aggressive methods and reveals that the security issue of CAN is not optimistic. Additionally, it shows that simply improving defense of a certain attack may decline security level of the entire system and overall protection is requested.

Acknowledgment

This paper is supported by NSFC (no. U1405251, 71671114) and SIT Collaborative innovation platform under Grant No. 3921NH166033.

References

[1] Y. Huang, G. H. Qin, T. Liu, and X. D. Wang, "Strategy for ensuring in-vehicle infotainment security," in *Applied Mechanics and Materials*, vol. 556. Trans Tech Publ, 2014, pp. 5460–5465.

[2] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental security analysis of a modern automobile," in *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE, 2010, pp. 447–462.

[3] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, 2015.

[4] L. Ben Othmane, H. Weffers, M. M. Mohamad, and M. Wolf, "A survey of security and privacy in connected vehicles," *Wireless Sensor Networks (WSN) For Vehicular and Space Applications: Architecture and Implementation. Springer, Norwell (accepted)*, 2015.

[5] A. Asvestopoulos, "Intrusion protection of in-vehicle network: study and recommendations," 2015.

[6] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle can," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 993–1006, 2015.

[7] K.-T. Cho and K. G. Shin, "Error handling of in-vehicle networks makes them vulnerable," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1044–1055.

[8] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive experimental analyses of automotive attack surfaces." in *USENIX Security Symposium*. San Francisco, 2011.

[9] D. Spill and A. Bittau, "Bluesniff: Eve meets alice and bluetooth." *WOOT*, vol. 7, pp. 1–10, 2007.

[10] R. M. Ishtiaq Roufa, H. Mustafaa, S. O. Travis Taylora, W. Xua, M. Gruteserb, W. Trappeb, and I. Seskarb, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *19th USENIX Security Symposium, Washington DC*, 2010, pp. 11–13.

[11] S. Wagenknecht and M. Korn, "Hacking as transgressive infrastructuring: mobile phone networks and the german chaos computer club," in *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, 2016, pp. 1104–1117.

[12] B. Schneier, "Attack trees," *Dr. Dobbs journal*, vol. 24, no. 12, pp. 21–29, 1999.

[13] V. Ramaswami, "A stable recursion for the steady state vector in markov chains of m/g/1 type," *Stochastic Models*, vol. 4, no. 1, pp. 183–188, 1988.