Modelling and Security Analysis of Internet Connected Cars

Fredrik Krantz

***Abstract*—With more and more Internet connected devices comes the question, are they secure enough. If such a thing as a vehicle were to be hacked there could be dire consequences. For example if someone acquired remote control of the brakes and steering of a vehicle they could be controlled with illicit intent. Several reports have shown that the internal networks of a vehicle with the protocols and devices used today are vulnerable to different kinds of attacks. So how can we make them more safe. The first step should be to show how vulnerable these systems are. This can be done in different ways. One way is to directly try to gain access to critical devices on a physical car i.e hacking it. Another way is to perform a manual security analysis of the car to map different vulnerabilities and and try to exploit those. A different way is to model the cars internal network with a tool that could do threat modelling and simulate attacks. One tool available being SecuriCAD. This tool is made with ethernet networks in minds and computers connected to these networks. An attempt to model in-vehicular networks shows to work with some adjustments. When a generalised model of a in-vehicular network is created in SecuriCAD the simulations shows vulnerabilities that correlates with what is shown in other reports. When modeling a 2014 Jeep Cherokee, the resulting attack tree of a replay attack given by the simulations is comparatively accurate to those attack steps made by Miller and Valasek when they hacked the same Jeep model in 2015. The method used in this project could be further improved but is a good proof of concept.**

I. Introduction

An increasing amount of devices are being connected to the Internet. Things such as toasters, phones, cars and many more that you would not consider needing a connection in the first place. But with proper implementation the Internet connection could provide a higher quality of life. With the introduction of IPV6 there is an increase in available IP addresses from almost 4.3 milliard to over 34 followed by 37 zeros. There will be almost no limitation on how many devices that can be online at once. Though one of the biggest challenges that this market is facing is the implementation of proper cybersecurity. An increase in devices could lead to an increase in cybercrime like ransomware and integrity breaches. This is due to the vulnerabilities and exploits that already exists in todays software and network architecture [1]. This concern is amplified due to manufacturers focusing on getting new performance effective devices on the market as fast as possible. For example Intel's CPUs which contained a function that directly accessed the kernel memory. This is a huge security risk and the patch for this caused that function call to take twice the amount of time [1] which decreased performance.

The latest car models often come with a cellular connection to the Internet and they are more computerised than ever. Containing up to a hundred ECU:s. ECU stands for Electrical Controller Units which control brakes, airbags and parts of the engine. These are interconnected by networks. A combination of ECU:s, sensors and different network buses creates a system which for example lets sensors in the steering wheel send signals to let the power-assisted steering know when the driver is turning. The most commonly used network protocol in a car is called Controller Area Network, or CAN for short. There are several ways to breach into this network excluding an Internet connection [2]. This shows just how vulnerable a car is to exploits in several ways. How easy it is to access vital ECU:s and how dire the consequences would be if someone had access to those ECU:s. These vulnerabilities has been proved to exist numerous times, one famous example is when Charlie Miller and Chris Valasek acquired remote control of a 2014 Jeep Cherokee whilst a test subject was driving it on a highway [3].

One way to improve security in these Internet connected systems is to use advanced tools to model and analyse them. This can show which parts of the network are most vulnerable and how they can be secured.

The software SecuriCAD was used in this project for modelling and simulation purposes. SecuriCAD is a threat modelling and risk management tool in which the user is able to model a home LAN or a larger corporate network. The modeller can create a model of the topology of the network. Then assign security measurements to different objects and use the built in simulation tool to show the probability of different threats. Some threats that is available to simulate is denial of service, compromising of devices or replay attacks. The attacker can also be connected to different objects in the model based on where the modeller might think the attack will come from. Either over the Internet or if the attacker have acquired a company laptop and is directly connected to an internal LAN [4]. SecuriCAD is developed and supported by Foreseeti which is a spin-off company based on KTH research. SecuriCAD uses a Bayesian methodology to interpret the probability of simulated attacks.

A car can have several attack surfaces such as bluetooth, a radio, DVD readers, physical access to the diagnostics port and RFID systems [2]. But since this project is focusing on cybersecurity in Internet connected devices these will not be used in the modelling and analysis part. The attacks that this project focuses on is the compromise of ECU:s, replay and Denial of Service attacks on internal network buses. Compromise meaning being able to take direct control the device. These are shown to be realistic attacks that could be

[1] P. Bright, Whats behind the intel design flaw forcing numerous patches? Jan 2018. [Online]. Available: https://arstechnica.com/gadgets/2018/01/whats-behind-the-intel-design-flaw-forcing-numerous-patches/

performed on cars [5]. In the modelling process there are several limitations to be made. An internal vehicular network is complex and consist of ten to even more than 100 electronic devices. Therefore the ECU:s used in the model is reduced to the most security critical ones i.e those connected to the Internet and those who control vital functions such as braking, steering and engine control.

This report is not a guide in how to model in SecuriCAD. Although the steps taken to create the model will be shown. The aim of this report is to create a model of a cars internal network with an Internet connection and simulate attacks on that network through that connection. The simulation results will then be analysed to determine the accuracy of the model.

II. Related Work

Previous research in the ICT-security field regarding cars has focused on penetration testing, hacking and threat modelling. Reports such as ”Hacking the can bus” [2] where they tested different ways to access the CAN bus. ”A survey of remote automotive attack surfaces” [6] where they penetration tested and found exploits of a attack surface of an 2014 Jeep Cherokee shows the penetration testing and Hacking of a cars internal network. Threat modeling has also been made of cars internal networks [7] where different threat modelling methods such as STRIDE were used. Threat analysis was also done where they step by step went through all potential attack surfaces to assess the potential of an attack. Also complete security analyses has been made such as ”Experimental Security Analysis of a Modern Automobile” [5] where they explore different vulnerabilities and demonstrate different attacks on a cars internal network. These are all manual ways to test the security of a cars internal network and are performed by security experts and other workers in the ICT-security field. There is no work found that completely and without expert knowledge can threat model and security analyse a car like the proposed method in this report.

III. method

This project was done using a iterative method. A visualisation of this method can be seen in Fig. 1. This method was repeated every week from literature study to simulation. The analyse was done when the model had met the goals set in the workplan. Every week a part of the model was made and simulated.

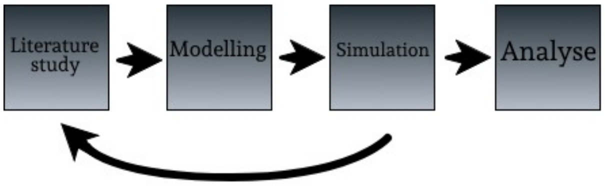


Fig. 1. Iterative method of the project

A. Literature Study

Knowledge about what a cars internal network looks like and works is vital when you try to model it. Therefore a literature study and search for information is done first in the iterative process. This was done through Google by looking for information from manufacturers, standards which are used in the industry and by looking for academic papers about the subject.

B. Modelling

Modelling is done in SecuriCAD. This is done by choosing the appropriate objects to represent the different parts of a cars internal network and connecting them together. Also the security measurements in the different objects that are used are set to the correct state based on information collected on the subject.

C. Simulation Results and Analysis

SecuriCADs built in simulation tool is then used and the results are noted. Then the next iteration begins with a literature study.

After the model has been completed the results from the latest simulation can be analysed. This is done to find the accuracy of the model.

IV. Results

A. Literature study

The first thing to consider is what does a cars internal network look like. What components does it consist of and what does the topology look like. Charlie Miller and Chris Valasek did a survey on different car models, one being the 2014 Jeep Cherokee. This topology can be seen in fig. 2. This is just one example of a topology and structure used within a car. The design can vary much depending on the manufacturer.

The internal networking of a 2014 Jeep Cherokee consists of three different networks. Two CAN and one LIN. Fig. 3 is a simple example of a CAN bus with three ECU:s connected onto it. The red and the blue line represents the physical function of the CAN. On the physical layer the CAN uses a twisted pair cable and modulates digital signals with voltage differentiation. CAN uses a protocol called CSMA/CA (Carrier-Sense Multiple Access with collision avoidance) to regulate the communication between devices [8]. This is a protocol which is also used in IEEE 802.11 (Wi-Fi). The version of CAN used in this model is called CAN-FD. It has a larger packet size and allows for some security implementations like message authentication [9]. This is not possible in standard CAN because of its small size of the package (8 Bytes).

The LIN (Local Interconnected Network) protocol that is presented in Fig.2 is a low speed network which is used when high bandwidth isn’t necessary. LIN is also cheap to implement compared to other network technologies [10].

There is also other network types that are used in a car. Though they are not used that much anymore. CAN is used to connect the most interesting ECU:s in regard to this project [2].

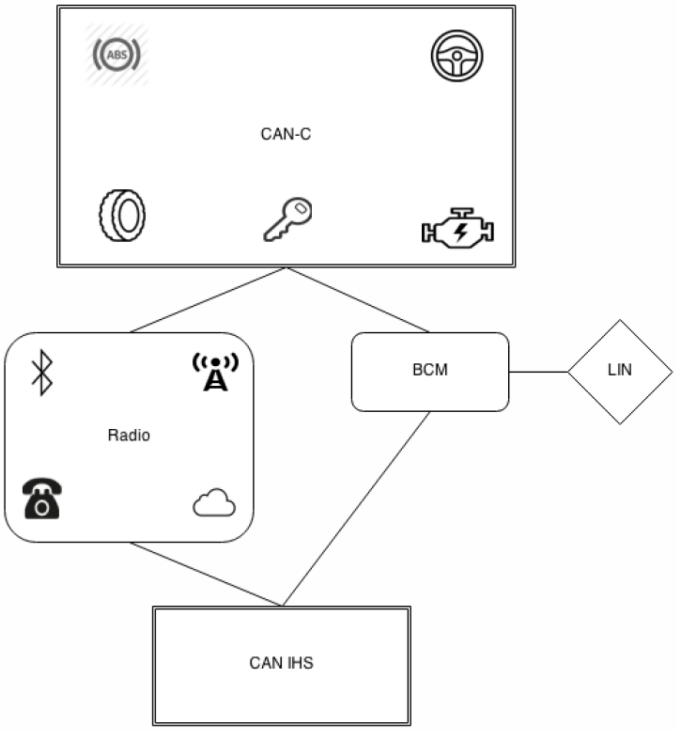


Fig. 2. Topology 2014 Jeep Cherokee made by C.Miller and C.Valasek [6]

Two of these being the MOST (Media Oriented Systems Transport) and the FlexRay protocol. MOST is used for media devices and it allows for a higher bandwidth. It is used in some models but not as much as the CAN protocol [6] [11]. FlexRay is a time-deterministic protocol with higher bandwidth that is used in time critical functions such as by-wire driving. It is expensive to implement and not used as much as CAN [6] [10].

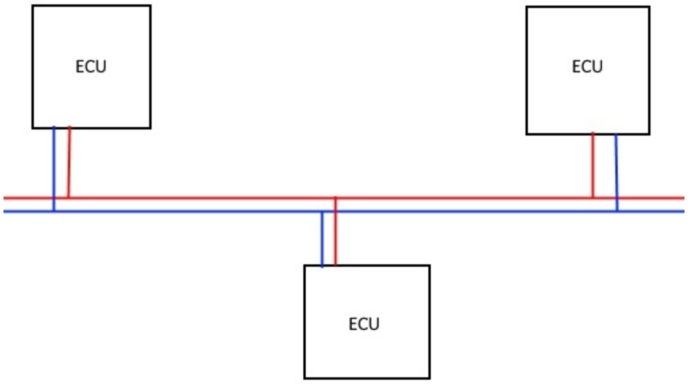


Fig. 3. Example of a CAN bus with three ECU:s connected

The purpose of these network technologies is to create a data communication channel between different devices in a car. These devices are called ECU:s (Electronic Control Units). ECU:s are embedded systems which controls electrical systems on a vehicle. They can also function as a server that handles calculations and only sends/receives data from other ECU:s that control electrical systems [2].

The Radio box in Fig. 2 is an ECU with several functions like bluetooth, cellular, radio and Internet connectivity. The Radio ECU is connected to two CAN buses. One is called CAN IHS and on that bus there are several ECU:s like those that control seat heating and door motors. They are not printed on this diagram because they are not the most critical ECU:s if they would be compromised. On the CAN-C bus on the other hand there are several ECU:s that can be considered safety critical. Like the steering, braking, engine and key control ECU:s. But also the Tyre Pressure Monitoring System ECU is considered safety critical due to it having a wireless communication interface [12].

Connected to the CAN networks are also a Body Control Module. The BCM:s function is to control and send commands to other ECU:s, so it doesn't control any physical function within the vehicle but acts as a controller for other ECU:s functions. The BCM is also connected to a LIN on which ECU:s like the rear view mirrors and lamp controllers are connected. The BCM ECU acts as a gateway ECU between two CAN buses and the LIN bus and can be compared to a ethernet switch. It repackages the data to fit the standard of different network protocols.

The software used on these ECU:s is either made entirely by the manufacturer who develops the internal communication network for a car. Or it is made using a already existing architecture standard. One such architecture is called AUTOSAR which stands for AUTomotive Open System ARchitecture. AUTOSAR is an open source standard which is developed by a partnership between a large number of automotive manufacturers [13]. AUTOSAR is the architecture that will be used in the model. Because it's open source thus easy to research and because it's becoming a world standard for automotive embedded software it should be fair to use in the model [14]. There are two versions of AUTOSAR, Classic and adaptive. Classic is used in this model due to it being the oldest and most used. AUTOSAR Classic is based on OSEK which is a standard for embedded operating systems in vehicles [15].

B. Modelling

First all objects in the model are created and assigned the correct properties. The Host object in SecuriCAD is described to be a kernel of an operating system and is used to represent PCs or servers [16]. This object is the one used to represent the ECU. On a Host object there is a required connection to a SoftwareProduct object representing the operating system. This software represents AUTOSAR. Both of these objects in SecuriCAD has a selection of security implementations that can be enabled, disabled and probability based. This can be seen in Table I and Table II. The modelling of these objects can be seen in Fig. 4.

The next step is to connect this ECU to a CAN-FD bus. This is done by using the Network object in SecuriCAD. The ECU:s is then connected to that CAN-FD object. This can

[2] Wikipedia contributors, Electronic control unit, 2018, [Online; accessed 26-04-2018]. [Online]. Available: https://en.wikipedia.org/ wiki/Electronic control unit

TABLE I
HOST SECURITY SETTINGS

Defenses	Implementation and reasoning/source for decision
ASLR Address space layout randomisation. This implementation fortifies against buffer overflow attacks.	Not implemented in AUTOSAR classic. But is on the adaptive platform [17]. Disabled.
AntiMalware Antimalware detects, removes and deters malware attacks.	Is not implemented, can be in the future [18]. Disabled.
DEP Data Excecution Prevention defends against buffer overflow by making memory areas non-excecutionable.	Avaliable on the adaptive platform [17]. Not on the classic. Disabled
Hardened This represents the procedures in which unused services, ports and hardware outlets are disabled.	No information available. Set to unset. (Default value off)
HostFirewall A firewall controls if dataflows should be blocked or allowed between hosts.	No information avaliable. Set to unset. (Default value off)
Patched Represents if the hosts are up to date with the latest security updates.	Set to on. An internet connection gives improved software support and patch availability.
StaticARPTables Maps IP addreses to MAC addresses to avoid spoofing.	Set to off. This measurements is with an ethernet network in mind, not a CAN.

TABLE II
SOFTWAREPRODUCT SECURITY SETTINGS

Defenses	Implementation and reasoning/source for decision
HasVendorSupport If the product still is supported and has access to patches.	On, the model has a internet connection and is assumed to still be supported.
NoPatchableVulnerability If it is known that the software has no patchable vulnerabilities.	No information available. Unset (default is off)
NoUnpatchableVulnerability If it is known that the software has no unpatchable vulnerabilities..	No information available. Unset (default is on)
SafeLanguages The sofware is developed in languages that perform checks to reduce the risks of buffer overflow.	No information available. Unset (default is off)
Scrutinised If the software has been thoroughly tested and checked for vulnerabilities.	No information available. Unset (default is on)
SecretBinary If there is access to the binary by an attacker. The attacker can then detect vulnerabilities. No access to the binaries makes it ”virtually impossible to find new vulnerabilities”.	No information available. Unset (default is off)
SecretSource If the sourcecode is opensource.	Autosar is opensource [13]. Set to off.
StaticCodeAnalysis The use of tools to find vulnerabilities and bugs.	No information available. Unset (default is on)

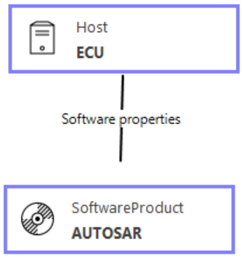


Fig. 4. A Host and a SWP object connected to each other.

be seen in Fig. 5. The Network object doesn't represent any physical device. Just the fact that all objects connected to Networks are connected to each other. The CAN-FD itself has no security measurements enabled. The measurements available on a network object is DNSSec, PortSecurity and StaticARPTables. All of these are TCP/IP related and no information was found on any comparable systems used in CAN-FD.

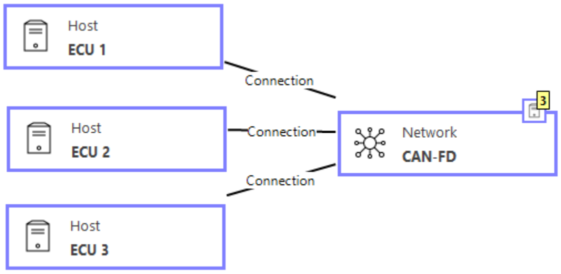


Fig. 5. Three ECU:s connected to a CAN-FD network object

The CAN protocol works in such a way that all messages from devices connected to it are broadcasted. The receiver knows which packages to pick up depending on the receiver address in the package. Therefore we need to model this broadcast behavior. This is done i SecuriCAD by using a Dataflow object. The Dataflow object represents communication between Service and Client objects. So we need to add Service and Client objects to the ECU hosts as well. A Service and Client represents the function of the ECU e.g the power-assisted steering that receives data from a sensor in the steering wheel. A ECU doesn't need to have both a Service and a Client object. But for this generalised model both is used, see Fig. 6. They also need an SoftwareProduct connected to them so an AUTOSAR object is connected to all the Client and Service objects. The Client and Service objects have one security measurement which is called Patched. It is enabled to represent the fact that the software is up to date and still supported. With a Internet connection to the car manufacturers can send out patches more efficiently.

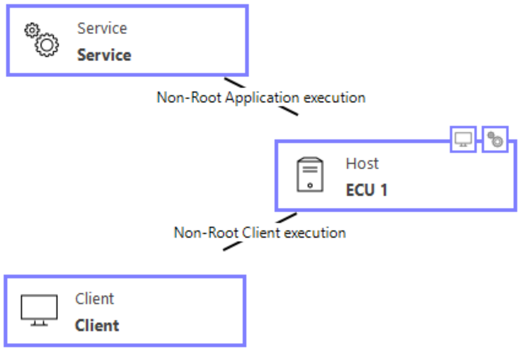


Fig. 6. A Service and a Client object connected to an ECU

The connection type that can be seen in Fig. 6 denotes how much access the Service or Client has to different commands and function calls in the operating system and kernel. The decision is to use the most secure option. Since there is no information found about how much access a service on a ECU in AUTOSAR has. In Fig. 7 the Service and Client objects as well as the CAN-FD object are connected to a Dataflow object. Simultaneously we can connect a Protocol object to the Dataflow object. The Protocol gives options to chose different security implementations that applies to the communication over the CAN-FD bus. The security measurements available is Authenticated, Encrypted and Nonce. The only one that is supported on CAN-FD and is enabled is Authenticated [9]. The other two are disabled.

All objects don't need to be shown all the time in SecuriCAD. In Fig. 6 there are small markers on the ECU which indicates that a Service and a Client is connected to it. A more complete version of Fig. 5 can be seen in Fig. 8.

Fig. 8 is still generalised with names such as ECU 1 and CAN-FD. The custom objects created in this model can be exported and saved as components to be accessed in other models. So with these objects the model in Fig. 9 is created. Here every CAN-FD bus has its own dataflow connected to its ECU:s and to the router.

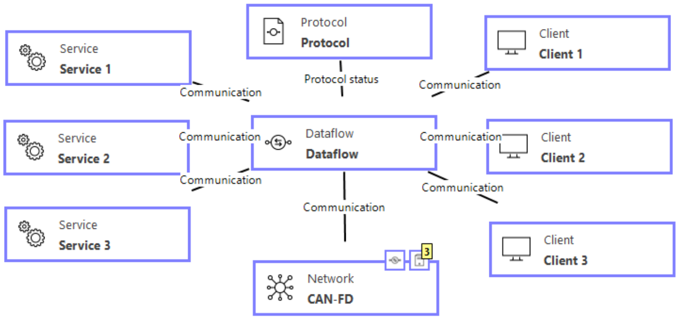


Fig. 7. A dataflow object connected to Service, client, protocol and network objects

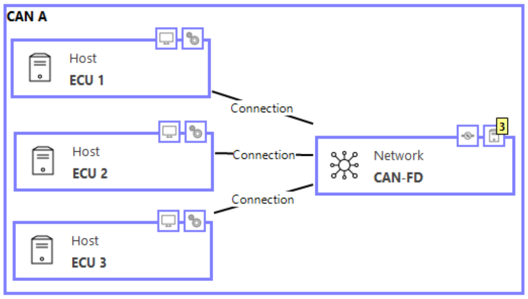


Fig. 8. A model of a single CAN bus with three connected ECU:s

In Fig. 9 all ECU:s are named after certain parts of an actual car. The ECU:s are connected onto two CAN-FD buses which have their separate names, one is called Drivetrain and one is called Chassi/safety control. This is an example on how a part of an internal vehicular network can look like [19]. All ECU:s in this model doesn't both have a Service and a Client. This decision was based on that only the ECU:s that requires to send data to other ECU:s need the Client object. Like the Driving Assistance which is an ECU without control over any

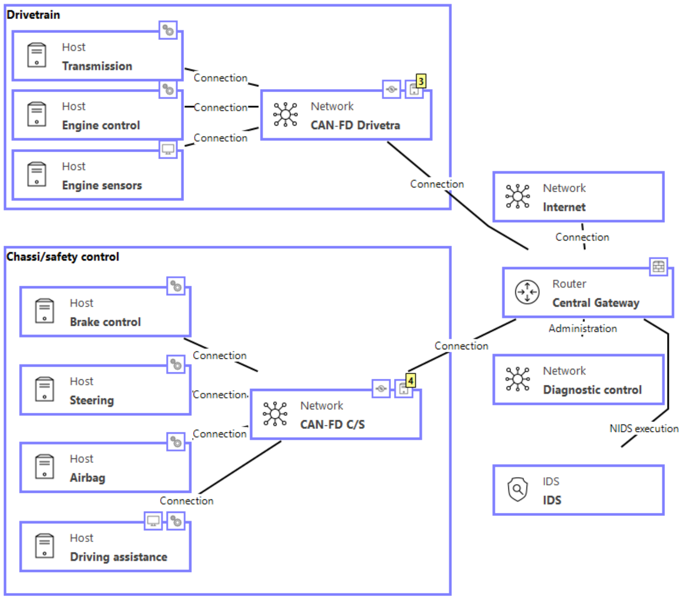


Fig. 9. A model of a cars internal network with two CAN buses

electrical devices. Its purpose is to calculate for the driving assistance functions based on input and send output to ECU:s that handles driving functions.

There is also more objects introduced in Fig. 9. A Router object called Gateway ECU, this is connected to a Firewall, an IDS (intrusion detection system) and four networks [20]. Two of these being the CAN-FD buses. One other is a administrative network that is required to use in SecuriCAD. There is also the Internet network object. Internet is made available through the use of a Connectivity Control Unit which is attached to the gateway [20]. This isn't represented by an object in the model. Just that the CGW is connected to a network called Internet.

The security measurements that can be set in this stage is in the Firewall object. The Firewall has two measurements. If it is enabled, and if the firewall ruleset is known to the modeler and configured properly. The default setting is used for these. Enabled is on and ruleset is probability:0.5 because there is no public information available in how manufacturers configure their firewalls on gateway ECU:s.

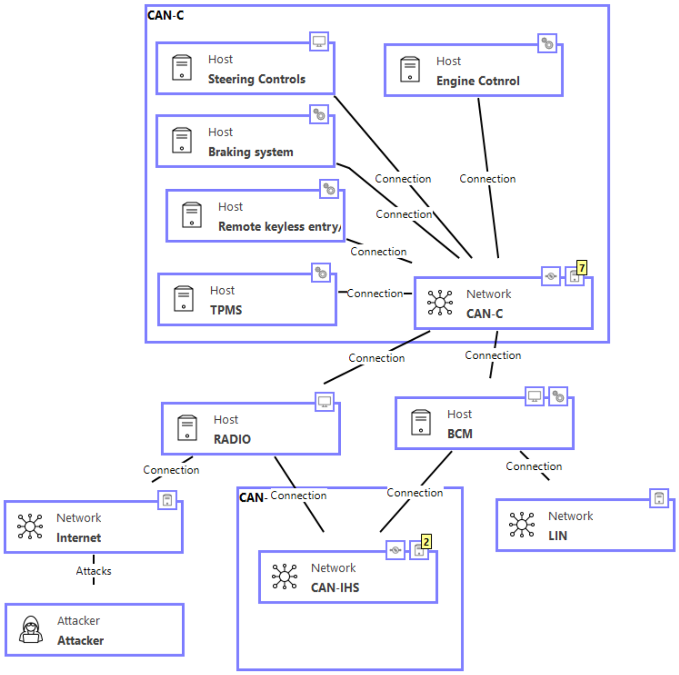


Fig. 10. SecuriCAD model of a 2014 Jeep Cherokee internal network

To summarise the security measurements that is enabled and relevant for the model in Fig. 9

- For the Host ECU and softwareProduct AUTOSAR there is Table I and II.
- No settings are enabled on the Network object.
- The Dataflow object has Authentication enabled.
- The Firewall has ruleset set to 0.5 and is enabled.
- The Service and Client has Patched set to on.
- The IDS Uses all the default values because it will not be used in the simulation. This is further explained in the Analysis section.

A model of a 2014 Jeep Cherokees internal network as seen in Fig. 2 can also be created using the objects from the generalised model with some changes. CAN is used instead of CAN-FD in the Jeep. so authentication can be removed from the security settings in the protocol object [3]. The Radio ECU:s Firewall setting was enabled, even though access was made through an open port other ports were closed. But the open ports that were found by Miller and Valasek can be represented by disabling the Hardened setting. This model can be seen in Fig. 10

C. Simulation Results and Analysis

An Attacker object must be added for the model to be complete. This attacker is connected to the object in which the attack should start from. The Attacker is connected to the Internet network with the connection type "compromise" in both the Jeep model and the generalized in Fig. 9. This is to regard the Internet as unsafe. To show results there also needs to be consequences attached to different attacks on objects. So on the Engine control, Transmission service and Brake control objects there is a 10 as consequence on the compromise attack. This is because a compromise and access to these ECU:s and service could lead to deadly road-accidents. On both CAN-FD networks the consequence for Denial of Service attacks are set to 9. this is because a DOS attack shuts down access to ECU:s on that network. That isn't as a direct threat to others in traffic as if the brakes would suddenly be fully enabled and a rear-ending could happen. Using the IDS in the model caused the simulation service to not give a report. A quick view of the attack path (Fig. 12) shows that BypassIDS that an IDS should protect from isn't existing. Therefore the IDS object was removed before running the simulations.

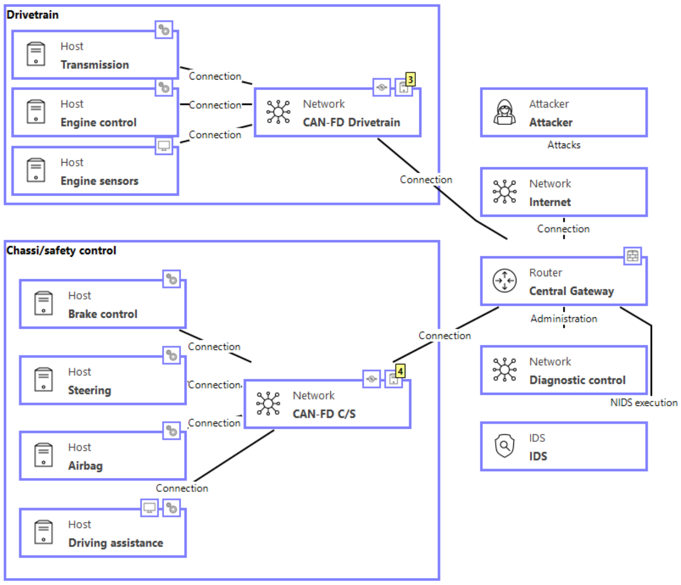


Fig. 11. The IDS was removed before the simulations were made.

Fig. 11 is the resulting model. Using this model with SecuriCADs simulation service gives the following risk assessment on Fig. 12. All attacks are considered to be of high risk.

If the firewall would be disabled completely, The risk would look like Fig. 13. All attacks considered to be in the critical

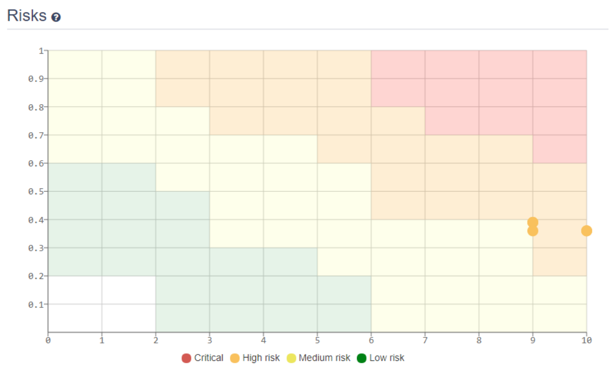


Fig. 12. Risk matrix of simulations performed on Fig. 11

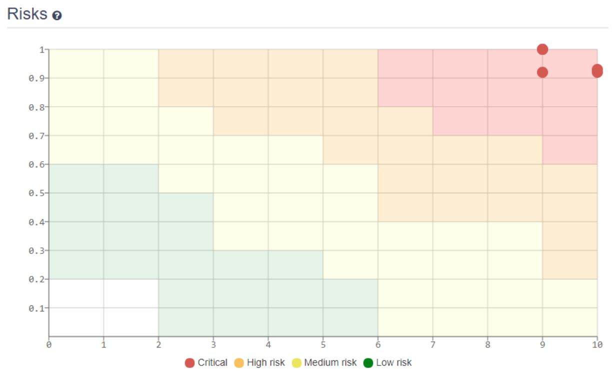


Fig. 13. Risk matrix from simulations performed on Fig. 11 with firewall disabled

zone. This shows that there isn't much beyond the firewall to ensure that the network is secure.

The report also shows the attack steps the attacker made to get to the attacks that were marked with a consequence number. The attack steps for a Denial of Service attack on the Drivetrain network can be seen in Fig. 14. This is with the firewall enabled and set at 0.5.

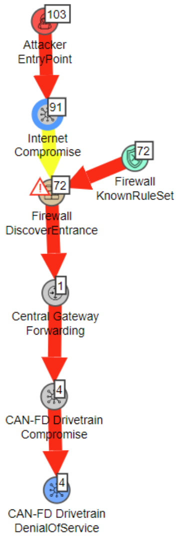


Fig. 14. Attack tree of a Denial of Service attack on a CAN-FD network

Fig. 14 is a simple attack path. It shows measurements as a green circle that can be made to further improve security. In this case it has to do with the firewalls ruleset. If the ruleset were to be set to 1.0 there would be a 0 risk for all attacks. The attack path for the compromise of an ECU can be seen in Fig. 15.

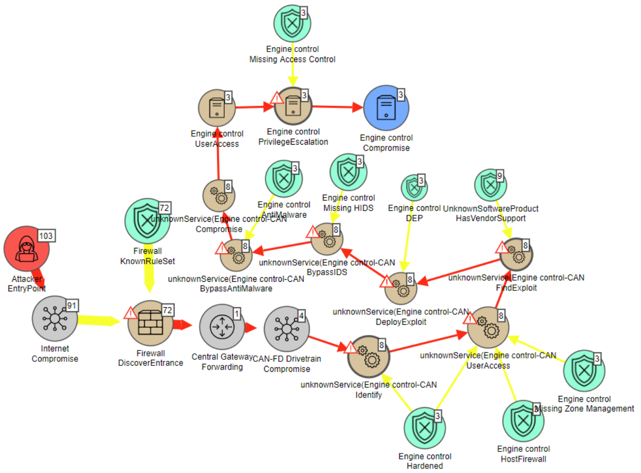


Fig. 15. Attack tree of a compromise attack on the Engine Control ECU

The attack path in fig. 15 is a bit more intricate. But it shows which parts that can be improved and gives a better picture about what security should focus on. Not just the first line of defense (firewall) but trough all parts of the network.

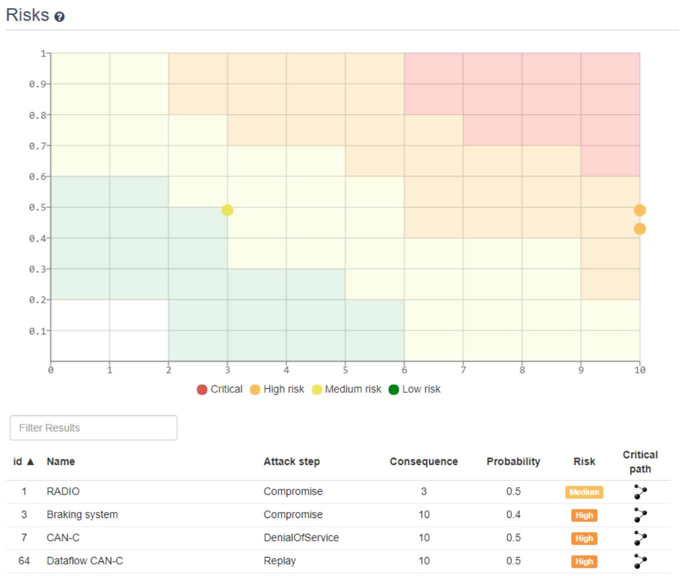
The consequences used for the Jeep model in Fig. 10 is

- 3 for a compromise of the radio ECU. This is used as a reference to see how much the probability lowers after the initial entry of the network.
- 10 for a compromise of the Braking system ECU.
- 10 for a Denial of Service attack on the CAN-C network.
- 10 for a Replay attack on the CAN-C network. This represents the actual attack made in the report by Miller and Valasek where they were able to send commands over the network unhindered [3].

Fig. 16 shows the probabilities of different attacks made on the Jeep model. From this we can see that the probability doesn't change much between the compromise of the Radio ECU and the other attacks. This shows that the network security beyond the Radio ECU is completely void.

Fig. 17 is the attack path for the Replay attack on the CAN-C bus on the Jeep Cherokee. The unknown service can be seen as the D-bus service accessed in the actual attack [3]. Just before the replay attack steps the attack path divides into two different paths. The compromise path seems more likely since ARP (Address resolution protocol) isn't used in CAN. There are several security measurements that could be implemented that is given in the attack tree. Most of them affect the Radio ECU itself, first and foremost the hardening of the ECU. This is what allowed the attack to happen in the first place [3].

The simulation results can also show the TTC (time to compromise) of an attack. The TTC of the Jeep replay attack can be seen in Fig. 18. This graph indicates how many days it



id ▲	Name	Attack step	Consequence	Probability	Risk	Critical path
1	RADIO	Compromise	3	0.5	Medium	
3	Braking system	Compromise	10	0.4	High	
7	CAN-C	DenialOfService	10	0.5	High	
64	Dataflow CAN-C	Replay	10	0.5	High	

Fig. 16. Risk matrix of attacks on the Jeep Cherokee model

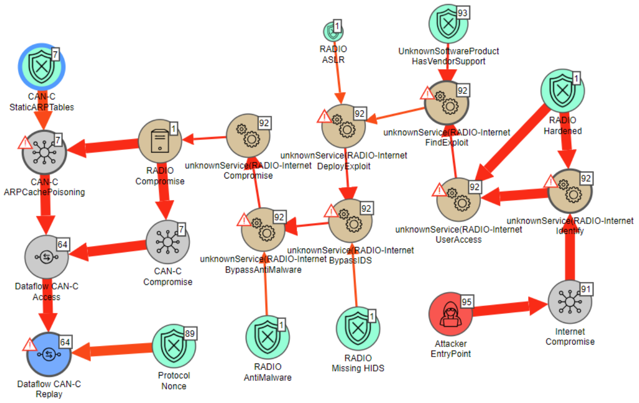


Fig. 17. Attack path of the Jeep replay attack

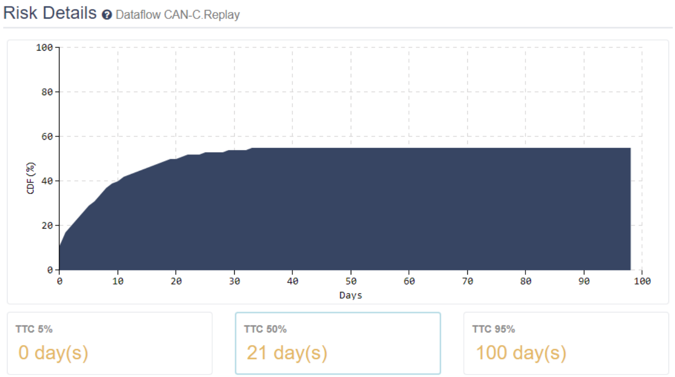
takes to reach a certain risk probability. The probability seen in the matrices is the converged end-value in the TTC graph.

V. Discussion and Conclusion

A. Discussion

For future work in this field the suggestion is to get more information about what security measurements are implemented in a vehicular network and if other attack surfaces such as bluetooth could be modeled. I had a meeting with two master thesis workers in Foreseetis office, they were working on developing new objects like the ECU in SecuriCAD. If it is successful and added to the public product it could give more accurate models and simulations of a cars internal network. These more accurate models could provide vehicle ICT-Security specialists with the tools they need to get a full view of the security status of a vehicle. It is also a great tool to use to show others with less knowledge about ICT the strength and weaknesses of security in vehicular networks.

The Risk matrices confirm that security is lacking in the internal networks of a vehicle [5] [2] [6]. The Attack path graphs shows which improvements can be made to secure these networks. These improvements is relevant even though a ECU is an embedded system and not as complex as the host object that is used in SecuriCAD, but some attack steps could be disregarded. For example the user access is something more relevant for a computer than a embedded system.



TTC 5%
0 day(s)

TTC 50%
21 day(s)

TTC 95%
100 day(s)

Fig. 18. Time To Compromise of the Jeep replay attack

As shown in the Jeep model and simulations, a replay attack is possible to model and the attack path is almost completely accurate if the UnknownService can be considered to be the d-bus service which was used in the actual attack. User access was actually granted to Miller and Valasek because the service didn't have a password restriction. Then by using physical access to a Jeep Cherokee they had bought they developed an exploit which could be used on unaltered Jeeps to replay CAN messages on the CAN-C bus. The probability given in the Risk matrix Fig. 16 for the replay attack could be up for discussion since the model is made by using previously created objects with some small changes. A small amount of research has been made on the Uconnect software that is used in the Radio ECU so there could be security implementations not considered in the model. These implementations could change the probability of the attack. But probably not the attack steps used. The TTC graph in Fig. 18 is a very precise way to try to predict the risk of a potential attack. This graph just as the risk matrix can be taken with a grain of salt. More development of SecuriCAD in regard to vehicle security could make the TTC graph more accurate and the reliability would be the same as the simulation results of computer networks.

B. Conclusion

SecuriCAD could be the go-to tool to give easy understood and holistic security analyses of a cars internal network. But further development is needed to show accurate probabilities and attack steps that is relevant to vehicular networks and embedded systems. The ECU which cannot accurately be represented by a Host and a SoftwareProduct needs to be given an own object to show the correct attack steps in a attack. The simulations are based on Bayesian statistics and needs to be further developed to give more accurate probabilities. If these things are improved SecuriCAD can be used to model and simulate attacks with the same accuracy on cars internal networks as it has on TPC/IP networks.

Acknowledgment

I would like to thank my supervisor Robert Lagerström for his help and guiding in this project. I also would like to thank Sotirios Katsikeas, Asmelash Girmay Mesele and Niklas Wiberg for their feedback on the accuracy of the model.

References

[1] M. U. Farooq, M. Waseem, A. Khairi, and S. Mazhar, "A critical analysis on the security concerns of internet of things (iot)," *International Journal of Computer Applications*, vol. 111, no. 7, 2015.

[2] R. Currie, "Hacking the can bus: Basic manipulation of a modern automobile through can bus reverse engineering," The SANS Institute, InfoSec Reading Room report series, may 2017.

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

[4] Foreseeti, "Securicad," April 2018. [Online]. Available: https://www.foreseeti.com/#securicad

[5] 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.

[6] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," *black hat USA*, vol. 2014, 2014.

[7] C. McCarthy, K. Harnett, and A. Carter, "Characterization of potential security threats in modern automobiles: A composite modeling approach," Tech. Rep., 2014.

[8] I. Standard, "Iso 11898, 1993," *Road vehicles–interchange of digital information–Controller Area Network (CAN) for high-speed communication*, 1993.

[9] T. Islinger, Y. Mori, J. Neumller, M. Prisching, and R. Schmidt, "Autosar secoc for can fd," *CAN Newsletter*, Mar 2017. [Online]. Available: https://can-newsletter.org/engineering/engineering-miscellaneous/170224_can-newsletter-magazine-march-2017

[10] Freescale Semiconductor, "In-vehicle networking," Brochure, 2006. [Online]. Available: https://www.nxp.com/docs/en/brochure/BRINVEHICLENET.pdf

[11] MOST Cooperation, "Most network," April 2018. [Online]. Available: https://www.mostcooperation.com/technology/most-network/

[12] 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.

[13] AUTOSAR, "Autosar," April 2018. [Online]. Available: https://www.autosar.org/

[14] S. Fürst, J. Mössinger, S. Bunzel, T. Weber, F. Kirschke-Biller, P. Heitkämper, G. Kinkelin, K. Nishikawa, and K. Lange, "Autosar–a worldwide standard is on the road," in *14th International VDI Congress Electronic Systems for Vehicles, Baden-Baden*, vol. 62, 2009, p. 5.

[15] D. John, "Osek/vdx history and structure," 1998.

[16] Foreseeti, *SecuriCAD 1.4 User Manual*, 5th ed., Foreseeti AB, Sveavgen 166, 3TR, 113 46 Stockholm, January 2018.

[17] AUTOSAR, "Requirements on Security Management for Adaptive Platform," october 2017. [Online]. Available: https://www.autosar.org/fileadmin/user_upload/standards/adaptive/17-10/AUTOSAR_RS_SecurityManagement.pdf

[18] I. You and K. Yim, "Malware obfuscation techniques: A brief survey," in *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on*. IEEE, 2010, pp. 297–300.

[19] H. Schweppe, "Security and privacy in automotive on-board networks," Ph.D. dissertation, Télécom ParisTech, 2012.

[20] BOSCH, "Central Gateway (CGW)," 4 2018. [Online]. Available: https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/connectivity-solutions/central-gateway-cgw/