Automotive Cybersecurity

6

Cybersecurity is the body of technologies, processes, and practices designed to protect computers, data, networks, and programs against intrusion, damage, or unauthorized access by cyberattacks. Therefore, this chapter begins, in Sect. 6.1, with an overview of automotive cybersecurity issues subdivided into ten subsections. It focuses on the scale and complexity of vehicles cyber and physical components' vulnerability to a variety of security challenges, intrusions, threats, and malicious cyberattacks whose intent is to disrupt communication, steal sensitive information or records, and impair the functioning of the system, identifying the risk level as a function of likelihood and consequences. Hence, a solid theoretical foundation for cybersecurity of vehicle cyber-physical systems is introduced too, based on concepts of artificial intelligence, deep neural networks (DNN), and deep learning (DL), control theory, epidemic theory, game theory, graph theory, and the importance of cybersecurity w.r.t. different kinds of attack scenarios, for example, the spear phishing attack. Section 6.2 introduces information technology security in automotive cyber-physical systems (CPSs) and the measures taken to ensure that automotive cyber-physical systems remain secure while interacting with other digital systems connected to a controller area network (CAN) system bus. It also describes the characteristics of today's attack taxonomies. As a logical next step, Sect. 6.3 focuses on hacking, automotive attack surfaces, and vulnerabilities and summarizes the anatomy of attack surface intrusion points in vehicles and the associated risks. Therefore, vehicle security depends on a variety of different methods and tools that systematically perform security testing, such as functional security testing, fuzzing, penetration testing, and others. Section 6.4 discusses intrusion detection, described as the detection of any set of actions that attempts to compromise the integrity, confidentiality, or availability of a system, as well as intrusion prevention, actions which attempt to prevent a detected intrusion from succeeding. Different detection methods for different kinds of intrusion types are described, including numerous static, dynamic, and hybrid methods for prevention. Section 6.5 discusses security and functional safety with regard to wireless mobile and sensor networks, platform

D. P. F. Möller, R. E. Haas, *Guide to Automotive Connectivity and Cybersecurity*,
Computer Communications and Networks,
https://doi.org/10.1007/978-3-319-73512-2_6

security, cloud computing, and data security, as well as functional safety. Section 6.6 includes several examples of car hacking. Section 6.7 contains a comprehensive set of questions on automotive cybersecurity topics, and finally followed by references and suggestions for further reading.

6.1 Introduction to Cybersecurity

The rapid growth in the development of computing technology and the Internet is having a huge impact on today's lifestyles, making day-to-day tasks easier and more convenient through wireless connection technologies. However, there is also a negative impact of this growth due to the emergence of new types of cybercrime being conducted through the use of information technology and communication (ICT). As ICT is increasingly used as a tool for committing crimes, security is a critical factor for the continued acceptance of the digital transformation and as part of the cyberspace defense against cyberattacks. Cyberattacks are facilitated by or committed using computers, networks, smart hardware devices and others, where they are agents, facilitators, or targets of the crime (Gordon and Ford 2006).

The cyber-physical systems (see Sect. 5.1 in Chap. 5, and Möller 2016) which are being used to embed the manifold of driver assistance systems, and safety and control systems into today's automobiles depend on sophisticated software to carry out specific functionalities. They develop quickly and increase in complexity, integrating communication, computing, and control into an infrastructure which plays a dual role with regard to the cyber and physical components used. Due to their scale and complexity, the cyber and physical devices of mission-critical automotive components are vulnerable to a variety of security challenges, intrusions, threats, and malicious cyberattacks. The purpose of these attacks is, for example, to:

- Compromise the functioning of the embedded cyber-physical system
- Denial of service
- Disrupt communication
- Steal sensitive information or records
- And others

Furthermore, the worldwide availability of the Internet allows cyber criminals to launch attacks worldwide on both cyber and physical system components from anywhere, at anyplace, at anytime. As a result, these cyber criminal attack-related security challenges require effective techniques for detecting, preventing, and recovering from cyberattacks. However, the main objective of automotive cybersecurity with regard to cyberattacks is to:

- Detect
- Deter
- Avert

This includes both previously known and unknown potential cyberattacks. Hence, cybersecurity is a body of knowledge about technologies, processes, and practices developed to protect networks, computers, programs, and data from cyberattacks, damage, or unauthorized access.

The traditional security approach has been to focus the most resources on the most crucial system components and to protect them against the biggest known threats. This necessitates leaving some less important systems or system components undefended and vulnerable to attack with regard to less dangerous known risks. Such an approach is insufficient when it comes to the current transformations in digitization as automakers embed automotive cyber-physical systems (CPS) to enhance and create new paradigms, such as connected cars and mobility services, which require extensive internal transformation across automakers' operations. Therefore, cybersecurity is one of the cross-cutting issues in automotive ICT because it is fundamental that authorized messages be delivered at anytime and at the right time to the right place without any disturbance or malicious attack.

Automotive cyber-physical systems (ACPSs) are engineered systems that are built from, and depend upon, the seamless integration of computational algorithms and physical objects composed of sets of wireless networked components, including sensors, actuators, control processing elements, and communication devices. Thus, using these smart and highly reliable automotive CPS, one must carefully consider the possible vulnerabilities of these systems which may result in potential security problems. In fact, concerns with the security of automotive CPS include malicious attempts through cyberattacks to:

- Intercept
- Defect
- Disrupt
- Fail

These types of attacks affect a large group of mission-critical systems or system components, which could result in the denial of available services, the theft of data, and could cause various types of damage.

Cybersecurity, from a general perspective, also deals with risk analysis, i.e., once a risk for an unauthorized intrusion has been identified, an analysis is carried out to determine the likelihood (probability) of the risk occurring and the consequence (impact) of that risk should it occur, which often is called risk quantification.

Modern vehicles can be targets of cyberattacks because of their complexity. Premium segment vehicles typically contain:

- $\geq$100 embedded electronic control units (ECUs)
- $\geq$2 miles of cable
- $\geq$100 million lines of software code
- $\geq$5 in-vehicle networks

Table 6.1 Risk level as a function of likelihood and consequences

	Likelihood		
Consequences	Highly likely	Possible	Unlikely
High	High	High	Medium
Moderate	High	Medium	Low
Low	Medium	Low	Low

This causes the ICT security requirements to dramatically increase. Therefore, the level of risk needs to be calculated as a function of likelihood and consequences. Table 6.1 illustrates the identification of a risk level with regard to the likelihood and consequences.

To define proper guidelines, automotive (vehicle) cybersecurity requires a well-defined risk analysis strategy. Automotive cybersecurity is vulnerable, and risk is an unequal vulnerability. The type and amount of risk depends on, for example, the:

- Cyberattacker's motivation
- Internal, local, and remote attacks
- Magnitude of hazards when security is compromised
- Vulnerability of system security
- And others

Vulnerabilities are weaknesses which allow a cyber attacker to reduce a system's information. With regard to vehicle cybersecurity, vulnerabilities include:

- Hazards to the lives of drivers and passengers
- Hazards to real-time operation
- Limited computational performance
- Limited vehicle external connectivity
- Unpredictable attack scenarios and threats
- Large number of components/parts from many different suppliers

The automotive industry is on the edge of a digital transformation, driven by trends such as:

- Emergence of new growth markets, such as services
- Increasing need for greater fuel economy
- New opportunities with regard to connectivity and its security
- Rapidly changing consumer behavior
- And others

In order to remain competitive and proactively address these trends, automakers embrace innovations specific to vehicle cyber-physical systems. The digital

transformation is playing a key role in taking the automotive industry into the future. Digital transformation across the automotive industry's ecosystem focuses on:

- Evolution of processes:
  - Optimal capacity planning and production
  - Reduced product development time and costs
- Evolution of products:
  - Increasing complexity and role of software
  - Move toward providing connected vehicle services
- New customer and original equipment manufacturer (OEM) relationships:
  - Better customer engagement and higher retention
  - Higher productivity through analytics and business intelligence
- New mobility solutions embedded in existing business models:
  - New service formats which focus on the holistic customer experience
  - New service and business models through cloud access
- Supply chain management:
  - Better component traceability and reduced warranty or recall costs
  - Greater supply chain visibility and reduced risks

Automakers are aware of the need to develop a new portfolio of capabilities and flexibility to generate value propositions for new customers or to transform their use models. Thus, enhancing and creating new features, such as those related to connected cars and mobility services, requires extensive digital transformation across automakers' operations. However, the ongoing trend of digitization has led to exponential growth in the volume of data generated. The real value is derived from the insights that businesses are able to draw from this data rather than from the information per se. Hence, this data is also of interest to cyber criminals. This demands an answer from automakers how to defend against the growth of intrusion points that results in manifold difficulties, such as:

- High endurance and long vehicle life cycles in which cyberattacks increase compared to the computational vehicle performance
- The difficulty of monitoring the status of automotive electronics with regard to limited vehicle external connectivity compared to traditional ICT-based systems
- Unpredictable cyberattack scenarios and threats
- Unpredictable hazards to the lives of drivers and passengers
- Difficulty of updating security software with regard to limited external connectivity of vehicles compared with traditional ICT-based systems

Therefore, with the increasing use of CPS for mission-critical operations in the automotive domain, cybersecurity issues must always be at the forefront of design. A new paradigm for automotive design and manufacturing is required, which can be stated as security by design (see, for example, German Industry 4.0 Platform (URL1 2018)). Cybersecurity is a challenging, comprehensive, interdisciplinary task and a major concern in today's automotive industry because it is imperative that anomaly

and vulnerability as a consequence of cyberattacks be detected, identified, and resolved for the protection of the vehicle's mission-critical systems. The determination of the intrusion method is especially important so that the regular operation of the mission-critical vehicle system will remain undisturbed. Cybersecurity requires coordinated efforts across CPS responsible for the manifold of vehicle functionalities, with respect to:

- Application security
- Computing security
- Data security
- Intrusion security
- Network security

Nevertheless, one of the most problematic aspects of cybersecurity is the fast and constantly evolving nature of security risks because cyberattacks are becoming more sophisticated and possess the ability to spread in a matter of seconds. Therefore, it is essential to provide the necessary tools to detect, classify, and defend against the various types of cyberattacks. Cybersecurity professionals argue that the traditional approaches to securing vehicle CPS information can become unmanageable because the threat environment can become too complex.

The majority of today's anomalies and vulnerabilities in automotive electronic control systems (ECUs) are a result of their network-based accessibility, which makes them vulnerable to remote cyberattacks. Accessibility provides an entrance for launching cyberattacks on ECUs, enabling new categories of vulnerability with regard to communication network channels:

- Interception
- Replacement
- Removal of information

Hence, at the most basic level, a cyberattack requires some form of access to the targeted system, and this is normally followed by some kind of exploit. The effects of the exploit phase can include data breaches such as:

- Defective system operation
- Denial of service (DoS)
- Destruction of data systems
- Disclosure of data
- Exfiltration of data
- Information removal or corruption
- Modification of data
- Unauthorized data access
- And others

which may cause the CPS to fail in fulfilling its mission-critical operations. This type of vulnerability can be traced back to the way in which the cyber and the physical components of automotive CPS electronic control units (ECUs) are integrated. In this vulnerable space, the cyber component provides computational and control supports, facilitates the fusion and analysis of data received from various sources, and controls data for the overall operation of the respective vehicle systems.
In contrast the access phase of a cyberattack can be broken down into two forms:

- Attacks that require some kind of user action or error of emission
- Attacks that are executed automatically, without any user action required to facilitate them

Every cyberattack has a life cycle w.r.t. its impact as described in Table 6.2, which may help to understand what the cyberattacker has done, as well when and where and also create questions like "What did the cyberattacker do?," "Is the cyber attacker still active now?," and others.

Remote network access facilitates highly productive interaction among the various physically distributed or concurrent collaborating units of vehicle cyber-physical systems, as well as the efficient overall vehicle system management as an integral part of cyber components. This accessibility, however, also allows the easy launch of cyberattacks.

Cyberattacks not only have tremendous impact on the cyber part of a system, but they also cause the physical part of a cyber-physical system to fail because physical infrastructures are weak with regard to security. One such weakness in the infrastructure of vehicle cyber-physical systems consists of sensor nodes which make up many components, each of which is subject to physical capture. A cyberattacker can remove or destroy the sensor node creating a monitoring gap and disrupting transmission of system-critical data. Nevertheless, the major security realm of cyber-physical systems in vehicles is the cyber part.

A classification and categorization of cybersecurity risks has recently been done by (Johnson 2016), shown in Table 6.3.

Table 6.2 Generic cyberattack life cycle (Johnson 2016)

Attack phase	Description
Data exfiltration	Attacker extracts data hacked
Installation	Attacker installs malicious SW on the target system or network
Lateral movement	Attacker moves from access point in other systems or networks
Maintain persistence	Attacker may maintain a presence on compromised systems or networks or install backdoors that allow repeated access in future
Obtain credential	Attacker obtains root or administrator privileges
Penetration or access	Attacker access the target system or network
Reconnaissance	Attacker scopes the target and develops his attack plan

Table 6.3 Classification and categorization of common cybersecurity risks (Johnson 2016)

Cybersecurity risk class	Common categories
Network and web-facing app Attacks	Code Injection, cross-site scripting, man-in-the-middle attack, sniffing, WiFi penetrations;
Malware attacks	Adware, attack ware, crime-ware, spyware
Social engineering attacks	Face-to-face, pharming, phishing, social media
Hacking attacks	Access control breaches, cloud side channel attack, domain name server redirects, password hacking
Denial of service (DoS)	(D)DoS flooding, hostage taking, wipers and overwriting
Advanced persistence attacks	Botnets, cloud nets, industrial worms, malnets, rootkits

6.1.1 Cybersecurity and Vulnerability

As cyber technology evolves, the number of tools available for launching cyberattacks increases. This means that cyberattackers upped their strategies for complex attack. Therefore, a major concern when studying data processing in distributed environments, such as automotive (ECUs), deals with the problem of how to model vulnerability to an intent-based cyber criminal adversary threat. Most traditional IT solutions follow the common assumption that all components are well disciplined to follow the protocol properly, with only one exception – that an adversary may keep a record of all intermediate data processing. Such an assumption substantially may underestimate the capability of adversaries, and thus makes it difficult to defend against adversaries that are behaving arbitrarily.

Like any other new technology field, most of the effort seems to be focused on mapping solutions from existing technologies, such as sensor nodes, which share the networked operation and low capability characteristics with cyber-physical systems. Hence, a solid theoretical foundation for cybersecurity of vehicle cyber-physical systems (ECUs) can be introduced based on concepts such as:

- Artificial intelligence (AI) and deep neural networks (DNN)
- Control theory
- Epidemic theory
- Game theory
- Graph theory

The aim of these concepts is to provide a holistic perspective on security, as shown in Fig. 6.1, to avoid adversary threats that consider both the cyber and the physical components.

6.1.2 Artificial Intelligence

The term artificial intelligence was coined in 1956 by John McCarthy and was defined as the science and engineering of making intelligent machines. Universal

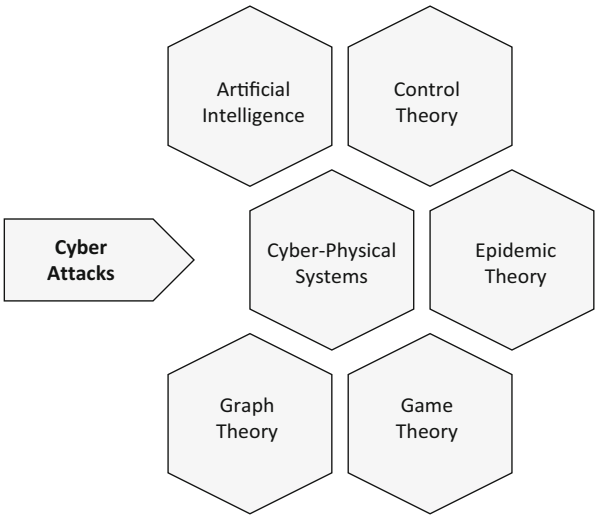

Fig. 6.1 Holistic perspective on cybersecurity

intelligence is the study of how to make machines do things which people do better. In computer science, an ideal intelligent machine is introduced as a flexible rational agent that perceives its environment and takes actions that maximize its chance of success at an arbitrary goal. Furthermore, the term artificial intelligence is likely to be applied when a machine uses cutting-edge techniques to competently perform or mimic cognitive functions that are intuitively associated with human intelligent behavior, such as learning and problem solving. In summary, artificial intelligence can be understood as:

- Academic field of study on how to create machines and software that are capable of intelligent behavior
- Constituted by machines and/or software
- Study and design of intelligent agents, whereby an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success

With the pace and amount of cyberattacks, human intervention is simply not sufficient for timely cyberattack analysis and initiation of an appropriate response, especially, when the adversarial threat is carried out by intelligent agents, such as computer worms or viruses. Combatting these cyberattacks can be done with methods delivered through artificial intelligence.

6.1.2.1 Artificial Neural Networks

Artificial neural networks (ANNs) are models inspired by biological neural networks used to estimate or approximate functions depending on a large number of inputs

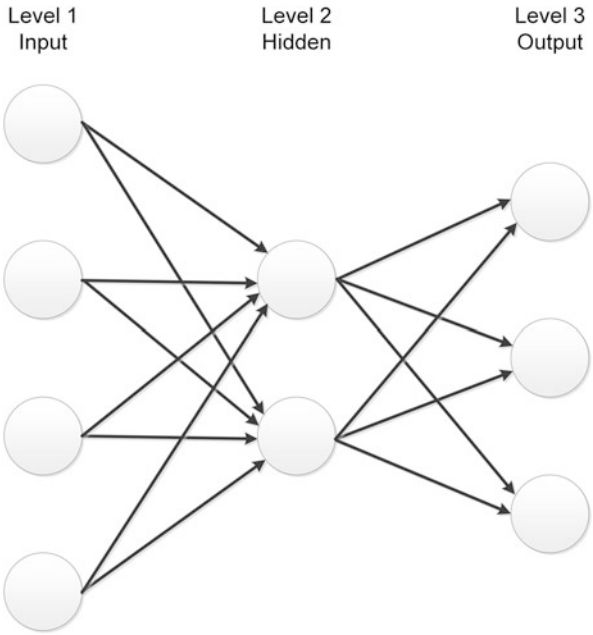

Fig. 6.2 Architecture of an artificial neural network with its three layers: input, hidden, output

which are interconnected with a group of nodes, as shown in Fig. 6.2, where arrows represent connections from the outputs of artificial neurons to the inputs of other ones.

Artificial neural networks are typically based on:

- *Architecture Body*: Specifies variables involved in the network and their topological relationships.
- *Activity Rules*: Represent local rules which define how the activities of the neurons change in response to each other.
- *Learning Rules*: Specify the way in which the artificial neural network's weights, $w_{i,j}$; $i,j = 1, \ldots, m, n$, change with time. Usually learning rules depend on the activities of the artificial neurons. They may also depend on the target values supplied by a training phase and on the current value of the weights, $w_{i,j}$, as shown in Fig. 6.3.

From Fig. 6.2, it can be seen that ANNs are massively parallel distributed entities made up of processing units (nodes), as shown in Fig. 6.3, which have the capability for storing experimental knowledge and making it available for use in monitoring anomalies behavior in cyberspace. The nodes are also effective against hidden adversary threats. A general flowchart depicting the monitoring of anomalic behavior in ANNs is shown in Fig. 6.4.

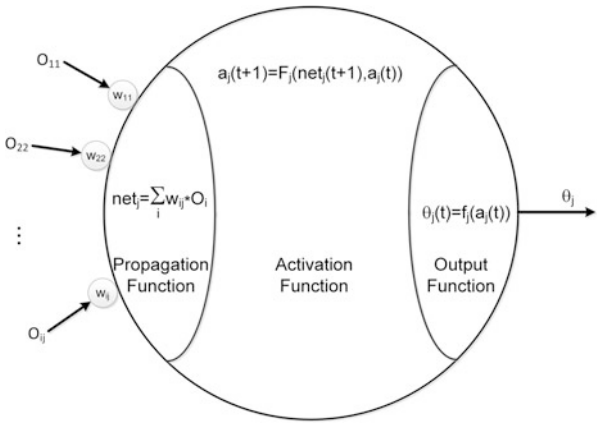

Fig. 6.3 Node structure of an artificial neural network

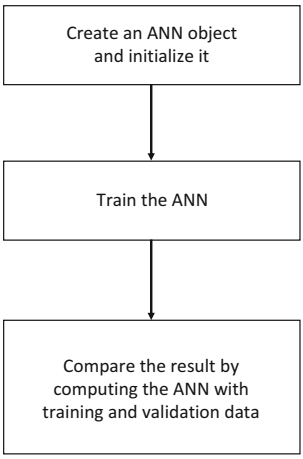

Fig. 6.4 Processing workflow of an ANN

6.1.2.2 Evolutionary Algorithms

Evolutionary algorithm represents a generic population-based metaheuristic optimization algorithm using mechanisms inspired by biological evolution, such as:

- Mutation
- Recombination
- Reproduction
- Selection

Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the quality of the solution. For example, a crossover or mutation needs to be carried out with probability, p, for which a simple MATLAB program looks as follows:

Table 6.4 Common functions related to random numbers

Distribution function	C/C++	Java	MATLAB
Normal distribution $N(0,1)$	rand_max	nextGaussian	randn
Random permutation between 1 and integer n	./.	./.	randperm
Round toward infinity	ceil	ceil	ceil
Uniform distribution $U(0,1)$	(float)rand()	math.random	rand

```
%Operator M is carried out with probability p
If rand < p
Operator M
End
```

with $rand \sim U(0,1)$ for the uniform distribution, as shown in Table 6.4. The density function of a uniform distribution random number in the range (0,1) denoted as $\xi \sim U(0,1)$ is as follows:

$$p(\xi) = \begin{cases} 1 & 0 < \xi < 1 \\ 0 & \text{otherwise} \end{cases}$$

Selection solutions are randomly chosen from current solutions and determined whether one could be selected.

6.1.2.3 Fuzzy Sets

A fuzzy set is a class of objects with a continuum of grades of membership. Such a set is characterized by a membership function. Thus fuzzy sets assign to each object a grade of membership ranging between zero and one. In this regard, a set is a collection of objects that belong to some definition of a membership. Thus, a fuzzy set, A, in $\boldsymbol{X}$ is characterized by a membership function, $\mu A(\boldsymbol{x})$, which associates, with each point in $\boldsymbol{X}$, a real number in the interval [0,1], with the values of $\mu A(\boldsymbol{x})$ at $\boldsymbol{x}$ representing the grade of membership of $\boldsymbol{x}$ in A. Thus, the closer the value of $\mu A(\boldsymbol{x})$ is to unity, the higher the grade of membership of $\boldsymbol{x}$ in A. The notions of complement, convexity, inclusion, intersection, relation, union, and others are extended to such sets, and various properties of these notions in the context of fuzzy sets have been established.

For example, the union of two fuzzy sets A and B with respective membership functions $\mu A(x)$ and $\mu B(x)$ is a fuzzy set C, written as $C = A \cup B$, whose membership function is related to those of A and B by

$$\mu C(x) = \max(\mu A(x), \mu B(x)), \quad x \in X$$

It should be noted that $\cup$ has the associative property, that is

$$A \cup (B \cup C) = (A \cup B) \cup C$$

6.1.2.4 Genetic Algorithm

A genetic algorithm (GA) is an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. Thus, it represents an intelligent exploitation of a random search used to solve optimization problems. Randomized genetic algorithms are by no means random; they exploit historical information to direct the search into the region of better performance within the search space. Genetic algorithms are based on an analogy of the genetic structure and behavior of chromosomes within a population of individuals using the following characteristics:

- Each successive generation becomes more suited to its environment.
- Individuals in a population compete for resources and mates.
- Individuals who are the most successful in each competition produce more offspring than those individuals that perform poorly.
- Genes from good individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.

After an initial population is randomly generated, the genetic algorithm evolves the three operators:

- *Selection*: Equates to survival of the fittest
- *Crossover*: Represents mating between individuals
- *Mutation*: Introduces random modifications

Hence, this machine learning approach imitates the process of natural selection which can be used for generating rules for classification of adversarial cyberattacks and developing specific rules for defending against specific types of cyberattacks.

6.1.2.5 Intelligent Agent

Agent theory is concerned with the question of what an agent is and the use of mathematical formalisms for representing and reasoning about the properties of agents. Agent architectures can be thought of as software engineering models of agents concerned with the problem of designing software (or hardware) that will satisfy the properties specified by agent theory. More in general, an agent can be introduced as an entity that perceives its environment through sensors and acts upon its environment through effectors, as shown in Fig. 6.5.

Thus, an intelligent agent can be viewed as an autonomous cognitive entity with standard boundaries and interfaces which understand its environment, can work by itself, and has an internal decision-making system that acts globally around other agents. Therefore, an intelligent software agent acts independently and in the interests of the user. They are used in various fields of application, for example, to control unmanned aerial vehicles, dynamic vehicle routing, route optimization in freight traffic, and others.

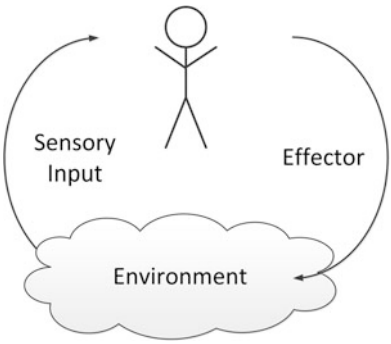

Fig. 6.5 Agent interacting with its environment

There are basically three different classification options for software agent types:

- *Autonomous Agent*: Is an entity that makes its own choices about how to act in its environment without any influence from a leader or global plan
- *Cooperating agent*: Involved in performing action of a plan to be executed through cooperation with the plan agent and/or other agents
- *Learning Agent*: Evaluate their actions independently in each iteration step and thus act differently in the next step

These agent attributes may occur individually or in combination. In this regard, smart agents are the highest level of intelligent agents.

In the case of a multiagent system, a group of autonomous mobile agents cooperate with each other in a coordinated and intelligent manner to plan and implement appropriate responses in case of unexpected events, such as defending against adversarial cyberattacks that an individual agent cannot solve.

6.1.2.6 Artificial Intelligence Methods

Artificial intelligence methods are helpful to detect, evaluate, and respond to cyberattacks as required for intrusion detection and prevention with regard to their specific features, as shown in Table 6.5 (Dilek et al. 2015).

An intrusion detection and prevention system is a part of software that monitors network or system activities for anomalous or malicious activities or policy violations, meaning it identifies possible adversarial intrusions and also tries to prevent them. For this reason, it contains four functionalities:

- *Analyzing*: Being able to provide efficient security against serious cyberattacks
- *Detecting Cyberattackers*: Detecting an attempt to change the system behavior which has to be realized in real time while the adversarial cyberattack is in progress (or immediately afterward)

Table 6.5 Advantages of artificial intelligence techniques suitable for intrusion detection and prevention

Technology	Feature
Evolutionary algorithm	Ant colony optimization
	Learning classifier system
Fuzzy sets	Interoperability to the environment
	Robustness of interpolative reasoning mechanics
Genetic Algorithm	Adaptability to the environment
	Flexible and robust global search
	Parallelism, allowing evaluation of multiple schemas at once
	Optimal solutions even for complex problems
	Robustness
Intelligent agent	Adaptability to the environment and user preferences
	Collaboration; awareness that human user can make mistakes, provide uncertain information, or omit important information; thus, the agent should not accept instructions without consideration and should check inconsistencies with the user
	Helpfulness; they always attempt to accomplish their tasks, having contradictory objectives
	Mobility
	Rationality in achieving their objectives
Neural net	Intuitiveness, since it mimics a biological neuron
	Intrusiveness, as they are an abstraction of a biological neural network
	Learning by example
	Nonlinearity, handling complex nonlinear functions
	Parallelism in information processing
	Resilience to incomplete data
	Versatility and flexibility with learning models

- *Monitoring in Real-Time*: Determining that a cyberattack is in progress (or immediately afterward) while minimizing false-positive alarms
- *Responding*: Reacting with regard to preventing the execution of the cyberattacker's attempt and generating reports to an a priori decided management level

The desired characteristics of an method must anticipate all possible forms of adversarial cyberattacks. The artificial intelligence intrusion detection and prevention system features are capable of detecting:

- *Buffer Overflows*: A cyberattack gaining process control or crashing another process by overflowing the other process' buffer.
- *Denial of Service (DoS)*: A cyberattack that prevents legitimate traffic or requests for network resources from being processed or responded to by the system. This cyberattack usually transmits a huge amount of data to the network. It is so busy handling the data that regular service cannot be

provided. After gaining access to the cyber-physical system, the cyberattacker can always further intrude by (Wang et al. 2010):
  - Flooding a cyber-physical controller or the entire sensor network with traffic until a shutdown occurs due to the overload
  - Sending invalid data to a cyber-physical controller or system network which causes abnormal termination or malicious behavior of services
  - Blocking traffic, which result in a loss of access to network resources by authorized objects or entities
- *Worm Detection*: A self-replicating program propagates without using infected files. Worms usually propagate through network component services on computers or through email(s).

In the case of distributed wireless communication networks or sensor nodes, intrusion detection and prevention through intelligent agents are combined with mobile agents. This adds mobility features for monitoring suspicious cyber activities as part of an adversary's cyberattack resulting in better intrusion detection and prevention (see Sect. 6.4).

Intrusions will probably identify vulnerable weak points in cyber-physical systems which can be easily attacked. Thus, vulnerability is a vehicle cyber-physical system susceptibility or flaw. Many vulnerabilities are documented in the Common Vulnerabilities and Exposures (CVE) database supporting management of vulnerabilities discovered, thus enhancing performance with regard to the variety of functions applied to intrusion detection and prevention systems, such as:

- Classifying
- Identifying abnormal activities through statistical analysis
- Installing and operating traps to record information about intruders
- Managing audit trails and highlighting user violation of policy or normal activity
- Mitigating
- Monitoring users and system activities
- Recognizing known attack patterns in system activities
- Remediating

The CVE is maintained by the MITRE Corporation, a not-for-profit organization that operates research and development centers sponsored by the US Department of Homeland Security (DHS) Computer Emergency Readiness Team (US-CERT) Office of Cyber Security and Communications (CS&C) (URL1 2016). It lists common names for publicly known vulnerabilities. Some of these vulnerabilities are specific to a platform, operating system (OS), application, or system; but some are generic and can apply to any system. Currently, more than 50,000 vulnerabilities are identified in the CVE system. The use of CVE has been standardized by the International Telecommunication Union (ITU), the National Institute of Standards and Technology (NIST), and other standards bodies.

Some vendors provide tools that test a component or system for known vulnerabilities which vary in their approaches and coverage. For each vulnerability, the tool may implement tests that attempt to use the vulnerability as a hacker might break into a system, stop a system from functioning, or manipulate the system function in an undesirable way. Automakers may need to modify some of these tools to work in the automotive network environment to test every cyber-physical system, network infrastructure device, gateway, OS, and other integrated components. As vehicles are becoming increasingly connected, it is essential to take a holistic view of security from inside the vehicle, through the mobile vehicle networks, to the IT backend infrastructure (IXIA 2014).

A recently published review about applications of artificial intelligence techniques to combat cybercrimes (Dilek et al. 2015) gives a good overview of published research papers applying artificial intelligence techniques in intrusion detection and prevention of cyberattacks to different kinds of cyber infrastructures which are highly vulnerable to intrusion and other threats.

6.1.2.7 Deep Neural Networks and Deep Learning

A novel intrusion detection and defense system approach against cyberattacks is based on deep neural networks (DNNs) to enhance the security of vehicular networks. The DNN can be trained with probability-based feature vectors that are extracted from the improbability of each class discriminating normal and attack data, to identify malicious attacks to vehicles. This technique adapts to recent advances in deep learning, initializing the respective parameters through unsupervised pre-training of deep belief networks (DBN) improving the detection accuracy. In reality it can be very difficult to extract high-level, abstract features from raw data because many of the factors of variation may influence every observable piece of data.

When it is nearly as difficult to obtain a representation as it is to solve the original problem, representation learning does not, at first glance, seem to help. In this regard, deep learning (DL) can solve the central problem in representation learning by introducing representations that are expressed in terms of other, simpler representations. DL allows building complex concepts out of simpler concepts. The idea of learning the right representation for the data provides one perspective of DL. Another perspective on DL is that depth allows the computer to learn a multistep computer program. Each layer of the representation can be thought of as the state of the computer's memory after executing another set of instructions in parallel. Hence, DL is composed of multiple processing layers to learn representations of data with multiple layers of abstraction. It discovers intricate structures in large data sets by using the backpropagation algorithm to indicate how a system should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. In this regard the backpropagation algorithm computes the gradient of an objective function with regard to the weights of multilayer stack architecture.

Meanwhile, DL has become more useful as the amount of available training data has increased. Thus, DL has solved increasingly complicated applications with

increasing accuracy over time, and has been successfully used in commercial applications, but was often regarded as being more an art than a technology and something that only an expert could use, until recently (Goodfellow et al. 2016). Thus, DL can scale machine learning being able to understand high-dimensional data with rich structures. Therefore, DL can take an input from a rich high-dimensional distribution and summarize it with a categorial label, for example, what CPU time is required executing an algorithm of a mission-critical system. Assuming the DL classification algorithm discards most of the input and produces a single output or a probability distribution over values of the single output, DL may be able to recognize an anomaly which shows the characteristics of a possible intrusion on the mission-critical system.

In this regard one interesting research direction is determining how distributed representations can be trained to capture the relations between entities. These relations enable to formalize facts about objects and how objects interact with each other. For example, in mathematics a binary relation is a set of ordered pairs of objects. Pairs that are in this set are said to have the relation while those who are not in the set do not (Goodfellow et al. 2016). In this regard anomaly time stamps in the execution of a mission-critical cyber-physical system identified by a DL-based intrusion detection system do not have the relation of the regular time stamps and are not in the set of regular time stamps and thus result in the identification of an intruder cyberattack situation through associative reasoning.

Associative reasoning is arguably one of the most essential intellectual capability of humans. It is the way we can reflect on ourselves. As knowledge of ourselves evolves, we find ourselves literally as spectators of our own development. The basic concept of associative reasoning is that everything is connected and networked but believing that everything is connected and networked is not so easy to understand, because everything seems to be disjointed, chaotic, and separated. The reason for this is simple, because normally humans do not know how the associations and links in their brain work. However, associating everything with anything, we can create and think on levels that we previously thought to be impossible.

John Locke describes in his essay "Concerning Human Understanding" that the task of logic is to examine the nature of the signs that the mind uses to make things intelligible to communicate them. In this regard DL and associative reasoning can be understood as the ways developing successful cybersecurity systems. For this purpose a well-defined syntax and semantic for formulating of excerpts and conclusions are required.

6.1.3 Control Theory

Cyber-physical systems (see Sect. 6.5.1) are able to connect the cyberspace and the physical space in an unprecedented manner through their increased sensing, networking, and computation capabilities. However, such connectivity options have also provided rich opportunities for adversaries to perform potential malicious cyberattacks. Therefore, control theory plays an important role in the analysis and

design of cyber-physical systems (Möller 2016) with regard to issues related to data imperfection and effects on control system performance. Data imperfection can be assumed to include:

- Delays
- Packet drops
- Quantization

These are inadequate for characterizing the possibility that transmitted data may not be true data collected by sensors or calculated by controllers because they could already have been manipulated by cyberattackers. This has raised questions relating to the secure control of cyber-physical systems. Therefore, traditional security aims to identify system anomalies and design strategies under the assumption that the system anomalies are of certain types of malicious cyberattacks; being either benign or random is not appropriate. Sophisticated cyberattackers are able to design strategies specifically to exploit vulnerabilities of the cyber-physical control system resulting in system abnormalities that are far away from random. Hence, some more formal methods can be chosen, such as:

- Control with shared processors
- Mission-critical components' privacy
- Verification and validation with timing constraints

The sensor of the control system transmits its measures at every preassigned time stamp. Then the controller calculates the control input by making use of the successfully received sensor measures. In control of cyber-physical systems, sensor data received must be consistent with the physical system behavior. If not, an adversary's cyberattack will be detected and, thereafter, removed. Therefore, the challenge for the cyberattacker is to degrade the control performance while sending data consistent with the physical part of the cyber-physical system. In contrast, the challenge for the defender is to identify if the received data is consistent with the physical part of the cyber-physical system in use.

Assume that a cyberattacker cannot be detected resulting in a trade-off between surreptitiousness and performance degradation. Therefore, the question to be answered is how to quantify surreptitiousness; or in other words, what is the performance degradation for a given level of surreptitiousness? The possible options for cyberattackers are:

- Surreptitiousness through cyber-physical system structure
- Surreptitiousness through statistical properties of the noise

Let the performance metric be the average estimation error covariance. In the absence of a cyberattack, the error covariance is $p(k)$; and in the presence of a cyberattack, the error covariance is $\bar{p}(k)$.

If the cyberattacker tries to enhance the intrusion without being detected, the following error covariance (Gupta 2016) is received

$$\bar{p}(k) = \lim_{k\to\infty} \sup \frac{1}{k+1} \sum_{n=0}^{k} \bar{p}(n).$$

With regard to this equation, an observer-algorithm can be embedded in the control system for evaluation of the data received in order to decide between two use cases:

$$UC_0 : \text{No cyberattack detected}$$

$$UC_1 : \text{Cyberattack detected}$$

Surreptitiousness can be measured by

$$p(\text{Decide } UC_i | UC_i) \to 0$$

and the probability of a false alarm can be described by

$$p(\text{Decide } UC_i | UC_0)$$

A cyberattack is then called surreptitious if no intrusion detection with property exists

$$p(\text{Decide } UC_i | UC_0) < p(\text{Decide } UC_i | UC_1)$$

A cyberattack is called ε-surreptitious for any $0 < \delta < 0.5$ if no intrusion detection exists such that (Gupta 2016)

$$p(\text{Decide } H_i | H_1) > 1 - \delta$$
$$p(\text{Decide } H_i | H_0) \leq O\left(e^{-k \times \varepsilon}\right).$$

Thus, for a given probability, p, of a missed detection, the probability of a false alarm cannot decay faster than exponentially with the rate $k \times \varepsilon$ as the number of measurements, k, increases.

6.1.4 Epidemic Theory

Modeling epidemic diseases can be done with regard to their basic principles:

- *Basic Reproduction Rate (R_0)*: Measures the transmission potential of a disease by counting the number of secondary cases following the introduction of an

infection into a totally susceptible population. The basic reproductive rate is affected by several factors:
  - Duration of infectiousness
  - Probability of infection being transmitted during contact
  - Rate of contacts in the host population

  For an epidemic to occur in a susceptible population, R_0 must be >1; i.e., the number of cases is increasing.
- *Effective Reproductive Rate*: Estimates the average number of secondary cases per infectious cases in a population made up of both susceptible and non-susceptible hosts. Introduced as the number of secondary infections generated by a typical infective rate reduced by the fraction of the host population that is susceptible.
- *Herd Immunity*: Occurs when a significant portion of the population has been vaccinated, which provides protection for unprotected individuals. The herd immunity threshold is the portion of a population that needs to be immune for an infectious disease to become stable in that community. If this is reached, then each case leads to a single new case; and the infection will become stable within the population.
- *Epidemic*: An increase in the frequency of occurrence of a disease in a population above its baseline or an expected level in a given period is a mathematical approach which follows three main goals:
  - *Determine* mechanisms to control and stop epidemic and study their influence on the process.
  - *Predict* the course of an epidemic in the future, which includes, among others, the final size of the epidemic and the convergence time to the steady state.
  - *Understand* the mechanisms for spreading the epidemic and how different parameters influence its course.

Hence, an epidemic model consists of a set of assumptions about the nature of the population of interest and the spreading mechanism. Assumptions with regard to the population of interest usually belong to the following categories introduced by (Daley and Gani 1999):

- *General Structure of the Population*: Population can be homogeneous such that every individual reacts to infection and spreads infection in the same manner. There can be several different:
  - Homogeneous populations
  - Stratas interacting
  - Completely heterogeneous populations
- *Population Dynamics*: Set of individuals can be closed or open. In a closed set, the number of individuals does not change over time so there are no new:
  - Births
  - Deaths
  - Emigrations
  - Immigrations

- *Disease Status of an Individual*: Individual can be:
  - A carrier without symptoms
  - Incubating
  - Infectious
  - Immune
  - Removed
  - Susceptible to infection

In 1927, Kermack and McKendrick (1927) established a deterministic epidemic model with a fixed population of N individuals and three important states: susceptible-infected-recovered (SIR). The results constitute a benchmark for a range of epidemic models. Thus, their main result treats the epidemic threshold as an important value to separate epidemics from small infections. The deterministic SIR model, with x denoting the fraction of susceptible, y the fraction of infected, and z the fraction of recovered, results in the following equations introduced by:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -\beta \cdot x \cdot y; \quad \frac{\mathrm{d}y}{\mathrm{d}t} = \beta \cdot x \cdot y - \gamma \cdot y; \quad \frac{\mathrm{d}z}{\mathrm{d}t} = \gamma \cdot y; \quad \frac{1}{x}\frac{\mathrm{d}x}{\mathrm{d}t} = -\frac{\beta}{\gamma}\frac{\mathrm{d}z}{\mathrm{d}t}$$

where β denotes the pairwise rate of infection and γ is the removal rate. For this system of equations, different cases can be considered:

- *Survival and Total Size*: Assuming the infection stops spreading, the fraction of susceptible that was never infected is x_∞, the fraction of individuals ultimately removed is $z_\infty = x_0 + y_0 - x_\infty$, and z_∞ is a unique root of the equation:

$$N - z_\infty = x_0 + y_0 - z_\infty = x_0 e^{-z_\infty \frac{\beta}{\gamma}}$$

  where x_0, y_0 are initial fractions of susceptible and infected nodes.
- *Threshold Theorem*: A major outbreak occurs if, and only if:

$$\frac{\mathrm{d}y}{\mathrm{d}t}|_{t=0} > 0$$

  which is equivalent to $x_0 > \gamma \cdot \beta$.
- *Second Threshold Theorem*: If x_0 exceeds $\gamma \cdot \beta$ by a small value, then the final fraction of susceptible left in the population is approximately

$$x_\infty = \frac{\gamma}{\beta} - \rho$$

  and $z_\infty \approx 2\rho$.

Whether a major outbreak occurs depends on the initial condition, like the fraction of susceptibles, at the start of the epidemic. Dependency of the spread on

the initial condition is a specific feature of the SIR model; in susceptible-infected (SI) and susceptible-infected-susceptible (SIS) models, the steady state does not depend on initial conditions.

In computer networks, epidemic modeling is mostly applied in the following areas:

- Epidemic algorithms and information dissemination in distributed networks (Chakrabarti et al. 2007; Eugster et al. 2004)
- Modeling computer virus and worm propagation (Kephart and White 1993)
- Propagation of faults and failures

Today, viruses and worms use different methods for spreading and different security vulnerabilities. Computer viruses are defined as small programs that can reproduce and copy themselves on other systems or on other files. The worm does not need user intervention to spread out. Most worms do not destruct the infected host computer, but some of them do. The destructive worm propagation model is derived based on a worm that writes data at a random point of a hard disc after, e.g., every 10,000 scans, until the infected computer crashes. Scanning worms are one of the most prosperous types of malware. They spread out quickly and automatically. However, they are also easy to detect and stop, leaving the Internet to stealthier types of malware. New worm types use social networks to spread. With the introduction of new web applications for the exchange of information and data, the number of cybersecurity incidents has increased.

Epidemic algorithms for information dissemination are also referred to as gossip dissemination, a computer-to-computer communication protocol. These epidemic algorithms are simple and easy to deploy, and mathematical tools allow the system behavior to be predicted. Usually, the information is either spread out forever, modeled by SI; or each node spreads the information for some time, and then it stops, following the SIR model (Eugster et al. 2004). Unreliable networks which use gossip algorithms can be modeled with an *SIS* model.

The epidemic dynamic model for disease propagation can be used for characterizing worm propagation, assuming that each computer is in one of the following states:

- Immune
- Infected
- Vulnerable

An immune computer cannot be infected by a worm. A vulnerable computer becomes an infected computer after being infected by a worm. The spreading mechanism – the cyber intrusion attack – determines exactly how the infection is transmitted.

6.1.5 Game Theory

Game theory is a mathematical method for studying decision-making scenarios with the interaction of at least two or more players. Such an interaction scenario includes:

- Participants
- Sets of possible utility payoffs which are called a game
- Sets of rational actions that each participant can take

In a real game, each player strives to pursue the best possible objectives by choosing courses of rational actions based on knowledge or expectations or another player's action. In game theory, game-theoretic models are studied which are abstractions facilitating the understanding of various classes of real-life situations, the so-called model of intent, which can be a utility function.

Definition 6.1
Given any pair of actions, i and j, in a set of possible actions, A, $u(i)$ and $u(j)$ refer to utility functions of i and j, which can adhere to $u(i) > u(j)$ if, and only if, the decision-maker prefers i over j.

The utility function is used to express the ordinality but not the quantity of preferences. Therefore, the player cannot know how much the decision-maker prefers i to j. Based on this characteristic, a decision maker's preferences could be represented by multiple different utility functions.

With regard to cybersecurity, one can postulate a system which incorporates the defender, D, and the attacker, A. In a case where a cyberattack is launched by multiple attackers, one has to write $A_1,\ldots, A_m$. Cyberattackers can be classified from a more general perspective as smart insiders and naive attackers. Insider threats occur when individuals within an organization misuse their privileged access to cause a negative impact on the attacked system in terms of (Nurse et al. 2014):

- Availability
- Confidentiality
- Integrity

Therefore, insider threats are an ever-growing problem in today's world of the Internet of Things (IoT), where everything is a device that may be used to access, store, and share sensitive data. The in-depth knowledge insiders possess of the security practices and monitoring policies place organizations in dire situations if these cyberattacks are executed. Thus, identifying insiders is a significant challenge and part of international research work with regard to:

- Anomaly detection of suspicious and malicious insider activity
- Identification of behavioral factors
- Recognition of signatures in cyberattacks

But smart cyber criminal insiders are afraid of being detected and, therefore, try to make optimal attacking decisions. Thus, their strategy may vary, for example, by choosing a mixed strategy which randomly chooses between two choices according

to a probability distribution, which results in a utility function as introduced by Jin et al. (2012) as:

$$u_A = \begin{cases} 1, & \text{if attacker launches an undetected cyber attack;} \\ -\beta_A, & \text{if attacker launches a detected cyber attack;} \\ 0, & \text{if attacker abstains;} \end{cases}$$

where u_A is the cyberattacker utility function and β_A is a predetermined insider preference parameter. Since the insider is afraid of being detected, one can assume $\beta_A > 0$.

Naive cyberattackers may bring blindly significant damage to a system by launching a cyberattack without fear of being detected. In this case, the naive cyberattacker realizes that the defender is weak, he can start attacking the system, and he will always succeed. If the naive cyberattacker is technically more sophisticated, anomaly detection has to be chosen for intrusion detection. Thus, the defender, D, not only detects the incoming adversary's threats using the anomaly detection technique but makes a proper trade-off between the detection rate and the false-positive rate.

Let $\gamma \in [0, 1]$ be a trade-off parameter such that the higher the value of γ, the smaller the false-positive rate and, hence, the smaller the detection rate. Normalizing γ such that the probability for detection of an adversary's attack is $(1 - \gamma)$ means that all cyber criminal attacks will be detected when $\gamma = 0$; however, a large number of false positives will be issued. When $\gamma = 1$, no cyberattack will be detected; and no false positive will be generated. Thus, the defender, D, has two objectives: (1) to detect as many attacks as possible and (2) to reduce the number of false positives. For each cyberattacker A_i $(1 \leq i \leq m)$, the loss of a defender, D, due to a cyberattack from A_i *be* $I_A(i) \in [0,1]$ which results in the loss of the defender associated with A_i and can be written as (Jin et al. 2012):

$$I_A(i) = \begin{cases} 1, & \text{if } A_i \text{ launches an attack that is undetected;} \\ b, & \text{if } A_i \text{ launches an attack that is detected;} \\ 0, & \text{if } A_i \text{ abstains;} \end{cases}$$

where $I_A(i)$ is the loss of D due to a cyberattack from A_i and b refers to a detected cyberattack, whereby $b \geq 0$ captures the potential cost for the defender to repair damages caused by the detected cyberattack. In case $b \geq 1$, an undetected adversary's cyberattack leads to even greater damage; elsewhere the defender could simply abandon any detection effort (Jin et al. 2012). According to the definition of the trade-off parameter, γ, if A_i chooses to attack, then the expected loss of the defendant object is $E[I_A(i)] = \gamma + (1 - \gamma)\ b$ if A_i chooses to abstain, $I_A(i) = 0$.

Besides the intent-based view on smart and naive cyberattackers and defenders, the taxonomy of games shows that game theory generally can be divided into two classes:

- *Cooperative Games*: Two players can bond together depending on specific promises or relationships between them.
- *Noncooperative Games*: Players are only allowed to make decisions independently based on two kinds of models:
  - *Strategic Games*: Implying strategic interdependence of players in a decision-making environment whereby each decision of a player is affected by one or all of the other players. These models consist of a strategic set of players, the possible actions of each player, and preferences, such as payoff functions reflecting the probabilities of winning for each player.
  - *Extensive Games*: Specifies a more inclusive form, called game tree, to explicitly depict the order of play and choices that players make at each node.

Furthermore, interactions between players represented by cyberattackers and defenders can be modeled as a noncooperative, non-zero-sum dynamic game with incomplete information, which considers the uncertainty and the special properties of multistage attacks. The model for this scenario is an approach along a special game tree where the adversary is the leader and the defender is the follower. Hence, multiobjective optimization methods are used to predict the adversary's best actions at each decision node. The defender also keeps tracking the adversary's actions, updates his knowledge of the adversary's behavior after each detected cyberattack, and uses his knowledge to update the prediction of the adversary's future actions (Luo et al. 2010).

Assumptions about perfect information do not hold true in real life and have to be expanded for a stochastic game model so that it is able to capture more realistic scenarios, as the player knows the system's true state at a particular moment in time with some probability of error, i.e., at any given point in time, the true state and a player's perception can potentially be different.

Assuming a constraint of imperfect information, the best strategy for a player considering other players' choice of strategies can be computed assuming the defender can compute his best strategy for reaching the Nash equilibrium of a stochastic game for which it is assumed that the defender's sensor is imperfect. For Nash equilibrium, no player can improve his payoff by unilaterally switching to a different strategy. It is implicit that the defender knows that the error probability of his sensor and the players' objectives are directly opposite, i.e., it is a zero-sum game (Shiva et al. 2010) indicating the existence of the equilibrium.

Definition 6.2
The Nash equilibrium represents an action profile for all players in a game with the property that no single player, I, can obtain a higher payoff by choosing a different action from a_i given every other player, j, adheres to a_j.

In (Sastry et al. (1994), a decentralized learning of the Nash equilibrium multiperson stochastic game with incomplete information is introduced, where after each play, the payoffs to individual players are random variables. Nothing is known regarding the distribution of the random payoffs. For learning optimal

strategies, the game is played repeatedly. The primary interest lies in (asymptotically) learning equilibrium strategies, in the sense of Nash equilibrium, with regard to the expected value of the payoff. For the decentralized learning algorithms developed after each play, each of the players updates his strategy based solely on his current action or move and his payoff. None of the players has any information regarding the existence of other players. Thus the game is played with imperfect information.

6.1.6 Graph Theory

Graph theory was introduced very early by Leonhard Euler (1707–1783) when he was asked to find a path that crosses over each of the seven bridges in Königsberg exactly once. Today, graph theory is used for finding communities in networks to detect hierarchies of substructures. In general, graph theory is a mathematical notation used to model pairwise relations between objects. In this context, a graph, $G = (V, E)$, is a pair of vertices (or nodes), V, and a set of edges, E, assumed to be finite, i.e., $|V| = m$ and $|E| = n$. Assuming $V(G) = \{v_1; v_2; \ldots; v_m\}$ with, e.g., $m = 5$, and $E(G) = \{c_1; c_2; \ldots; c_n\}$ with, e.g., $n = 6$, the corresponding graph is shown in Fig. 6.6.

From Fig. 6.6, it can be seen that graphs are used for designing topological properties for complex networks, e.g., to shape or optimize a network's dynamic performance measures, whereby nodes represent program statements and directed edges represent control or data dependencies between the nodes.

Studying complex network design is germane to cybersecurity with regard to theoretically controlling fraud detection and network intrusion detection. Both require methods for calculating the regularity of a graph to detect behavior anomalies which indicate intrusion detection. Intrusion detection systems have been widely used to detect malicious behavior in network communications and hosts. Thus, intrusion detection and its management are an important capability for distributed intrusion detection solutions, making it possible to integrate and deal with different types of data or collect and synthesize alerts generated from multiple hosts located within the distributed network system environment. Hence, defending complex networks against intrusions is very difficult because a defender must be able to locate the paths into the network and prevent adversaries from using them, while

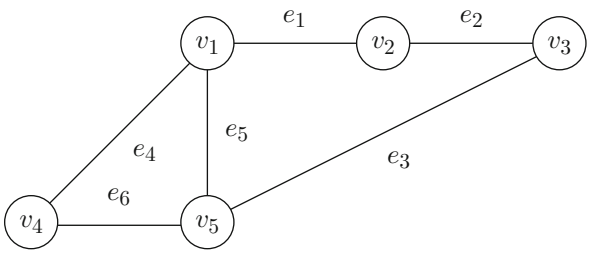

Fig. 6.6 Simple graph

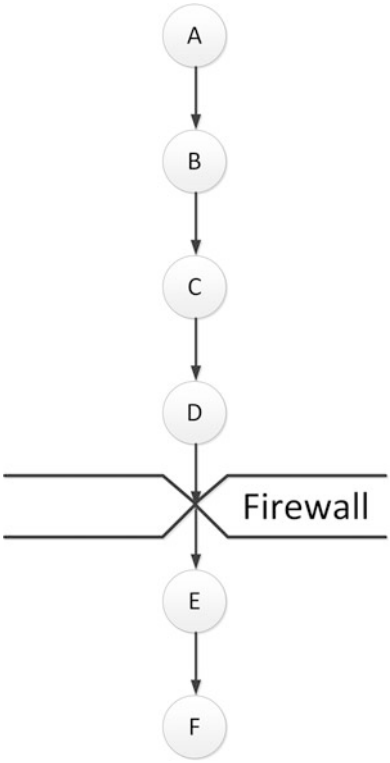

Fig. 6.7 Simple example of a network with a firewall

adversaries need to find only one unprotected path. Therefore, attack graphs are a valuable method for network defenders, illustrating paths an adversary can use to gain access to a targeted network. Defenders can then focus their efforts on patching the vulnerabilities and configuration errors that allow adversaries the greatest amount of access and work to secure those vulnerabilities.

Let's assume a simple network with six nodes, which correspond to states, and edges, which correspond to vulnerability instances. The nodes are class-divided by a firewall, as shown in Fig. 6.7.

The adversary may intrude undetected from Node *A* and can directly compromise Nodes *B*, *C*, and *D*. Assuming the adversary cannot traverse the firewall and compromise Nodes *E* and *F*, thereby completing the process by embedding malware on Node *F*, the attack graph has avoided infection of the mission-critical Node *F*. This is shown in the reachability matrix in Fig. 6.8, for the simple network depicted in Fig. 6.7, where a row represents a source interface on a node, a column represents a target port on a destination interface, and each cell indicates whether or not the source can reach the target.

Fig. 6.8 Reachability matrix; for details see text

	B	C	D	E	F
A	x	x	x		
B	x	x	x		
C	x	x	x		
D	x	x	x		
E				x	x
F				x	x

Thus, the attack graph workflow, in general, consists of the following parts:

- Correlation quality
- Cyber attack graph construction and its visualization to improve alerts
- Information gathering

The example in Fig. 6.7 uses automatic vulnerability extraction and creation of an attack graph based on unified data models. They identify typical structures of deliberately designed networks, allowing security/vulnerability analyses to be customized specifically for these networks. Thus, it can be determined whether deliberately designed networks have favorable or unfavorable security/vulnerability properties; and response strategies based on these characteristics can be developed. With some enhancement, the deliberate design methods can be used by network engineers to allocate available resources to improve security or reduce vulnerability. But the design methods must be extended in such a way that the performance metric for design includes security and/or vulnerability measures, in addition to other metrics for the network's dynamical performance. Hence, vulnerability and system information in attack graphs can be used to prioritize and tag incoming intrusion detection alerts. Therefore, the attack graph is used during the correlation process to select and optimize correlation results to protect critical resources in networked environments which can be achieved by quantifying the likelihood of potential cyberattacks.

With regard to the problem of probabilistic incorrect computing caused by shared dependencies in nodes, the methodology for security risk analysis based on attack graph nodes and a common vulnerability scoring system allow quick calculation of the probability of cyberattacks. In this context, the method used is the dependency graph for sets of events because dependence is a common feature of a relationship between objects. The relationship between objects can be modeled as a graph, with nodes and edges corresponding to objects and links, respectively. Thus, dependence is measured based on the description of this graph which means that the dependence degree of *Object A* on *Object B* is the probability p of B determining A. Suppose the dependence degree of *Object A* on Object B is $Dep(A \leftarrow B)$, which describes the degree of the determinant. In the case of a cyberattack on the directed edge from A to B, cyberattacker A introduces malware to B, so A is dependent on B.

With the dependency graph method, it is possible to measure the dependence of A on B by computing the sum of the dependent values on each path which results in the computation of the dependence degree $Dep(A \leftarrow B)$.

6.1.7 Importance of Cybersecurity

As the world becomes increasingly more interconnected through digital transformation, users must pay more attention to the security of their digital connections, since the past decade has witnessed a remarkable increase in the use of digital technologies. However, the newest wave of digital technologies is different. This has been accompanied by the fast, constantly evolving spread of security risks. It seems as though every week there are new headlines about cyberattacks bringing an organization's computers or network to its knees, with the resulting bad publicity and embarrassing revelations appearing as front-page news. This raises the question of how to protect organizations and systems from these issues.

The best protection is the development and implementation of plans and procedures to improve intrusion detection *and* prevent/eliminate vulnerabilities. One way to demonstrate the need for those types of procedures is to perform a cybersecurity audit. A better process is to send a clear request for proposal to potential audit suppliers which may move the process forward much more effectively.

The traditional approach in cybersecurity is to focus on the most crucial systems and/or components and to protect them against the biggest known threats, leaving some less important system components undefended and exposed to some less dangerous risks. This approach is insufficient for the currently expanding digital networked systems environment. The reason that cybersecurity professionals believe that traditional approaches to securing cyber-physical systems information are becoming unmanageable is because the possible threat environment has become extremely complex. In this regard, cyber-physical systems (see Sect. 6.5.1) have been identified as vulnerable to cyberattacks because of their network-based accessibility, which makes them vulnerable to remote access. Thus, the consolidation of cyber and physical components within cyber-physical systems enables new categories of vulnerability to develop with regard to:

- Interception
- Replacement
- Removal of information from communication channels

This results in malicious attempts by cyberattackers to affect cyber-physical systems operations by:

- Capture
- Disruption
- Defect creation
- Failure

The reason for this vulnerability can be traced back to the way in which cyber and physical components are integrated into sensor and communication networks. Sensor networks consist of many tiny components, each of which is subject to physical capture. Communication networks are systems of interconnected units that structure information exchange while allowing access to digital technology. This is becoming more and more essential when considering the extreme daily use of smartphones, tablets, gadgets, and other smart devices. Using today's new digital technology, it is easy to access a better quality of information, in greater quantity, at faster speeds via the Internet. But the vulnerability of this cyber-based infrastructure is a huge problem on which cyber criminals are capitalizing through attacks on sensory and communication networks. Thus, cybersecurity is both a critical area and one that is the most vulnerable to exploitation in the context of very complex supply chains and cyber-based operational infrastructures. In the vulnerable space, cyber components provide:

- Computing
- Control software
- Processing
- Sensory support

They facilitate the analysis of big data received from various smart sources, social media collaboration, and a cyber-physical system's overall operation. Therefore, a single successful cyberattack on a critical system node, if unmitigated, can have the potential to affect a significant number of important operational capabilities resulting in (see Sect. 6.1):

- Defective operation
- Denial of service (DoS), a common attack in the cyber domain
- Destruction and exfiltration
- Information corruption
- And others

Hence, cyberattacks causing denial of service may occur by creating an artificial mechanism that keeps the targeted systems unnecessarily busy, delaying or denying regular operational system services, which may be avoided if the intrusion method can be determined, and measures are taken to defend against it. Therefore, the software needs to be designed for the appropriate level of security from the outset; and some cyber-physical systems may need to be checked for resilience before being used. However, numerous solutions are available that analyze patterns and signatures in program codes and behavior of program executions in order to identify the presence of malicious agents or malware, helping system administrators to disable them. The techniques used for intrusion detection (and prevention) can be classified as follows (Zeltser 2015):

- *Behavior Detection*: Observes program execution and attempts to detect malware by looking for suspicious behavior(s), such as:
  - Unpacking of malcode
  - Modifying host files
  - Observing keystrokes

  Noticing such intrusions allows antivirus tools to be activated and the presence of previously undetected malware on the protected system to be detected. Therefore, behavioral detection makes the use of antivirus tools an intrusion prevention technique.
- *Cloud-Based Detection*: Detects malware by collecting data from protected systems and analyzing it on the provider's infrastructure. This is usually done by capturing relevant details about the file(s) and its execution on the endpoint of a line and providing them to the cloud engine for processing. Moreover, the vendor's cloud engine can derive patterns related to malware characteristics and behavior by correlating data from multiple systems. Hence, a cloud-based engine allows individual users of the antivirus tools offered to benefit from the experience(s) and knowledge of other members of the cloud community regarding intrusion detection and prevention.
- *Heuristics-Based Detection*: Detects malware by statically examining files for suspicious characteristics without an exact signature match. Thus, an antivirus tool might look for the presence of rare instructions or junk code in the examined file(s). The antivirus tool might also emulate running the file to trace what it would do if executed, attempting to do this without noticeably slowing down the running system. A single suspicious attribute might not be enough to mark the file as malicious. Finding several such characteristics, however, might exceed the predetermined risk threshold, leading the antivirus tool to classify a file as malicious.
- *Signature-Based Detection*: Uses key aspects of the examined file(s) to create a static fingerprint of known malware. A signature could represent a series of bytes in the file(s). It could also be a cryptographic hash of the file(s) or its section(s). This method of malware detection has been an essential aspect of antivirus tools since their inception; it remains a part of many antivirus tools to date, though its importance is diminishing. A major limitation of signature-based detection is that this method is unable to mark malicious files for which signatures have not yet been developed. Thus, modern cyberattackers frequently mutate their creations to retain malicious functionality by changing the file's signature.

In general, antivirus vendors have to incorporate multiple layers into their tools to keep up with the intensifying flow of malware samples, as relying on a single approach is no longer a viable option. Malicious files can do anything any other program/file can, such as:

- Erasing a stored file
- Stopping a running program
- Writing a message on a computer screen
- And others

Moreover, malicious files may do nothing at all immediately; they can be embedded to lie dormant, undetected, until some event triggers the file to act. The trigger used can be any of the following, some combination of these, or a random situation.

- Condition
- Count
- Date
- Event
- Time
- Time interval

In fact, malicious file(s) can pose different threats each time or nothing most of the time with something dramatic on occasion. Malicious files (code) can touch everything the user can touch and in the same ways. Users typically have complete control over their own program code and data files; they can read, write, modify, append, and even delete them. However, malicious files (code) can do the same, without the user's permission or even knowledge. There are different types of malicious files, (code) as shown in Table 6.6, which can be used to introduce cyberattacks.

The term virus was coined because the affected system reacts like a biologically infected system, meaning it infects other healthy components/systems by attaching itself to the program code of the respective component/system and either destroying it or coexisting with it. The infection usually spreads at a geometric rate, eventually overtaking an entire system and spreading to all other connected systems.

A common means of virus activation is an attachment to an e-mail message. In this attack, the adversary tries to convince the recipient of an e-mail message to open

Table 6.6 Types of malicious files (code)

File/code type	Characteristics
Logic bomb	Triggers action when a specific condition occurs, such as time, date, count, interval, or some combination of these
Rabbit	Virus or worm that self-replicates without limits with regard to exhausting computing resources
Time bomb	Triggers action at a specified time
Trap door	Allows unauthorized access to functionality
Trojan horse	A login script that solicits the user's login and password and passes the identification information on to the rest of the system for login processing
Virus	Transient or resident viruses are known. A transient virus has life depending on the host's life. A resident virus locates itself in a memory; it can then remain active or be activated as a stand-alone program
Worm	Propagates copies of itself through the network and operates through the network. In comparison, a virus spreads through any medium but usually uses copied programs or data files

the attachment. Once the viral attachment is opened, the activated virus can run its intended task. The virus can be executable code embedded in an executable attachment, but other types of files are equally dangerous. For example, objects, such as graphics or photo images, can contain code to be executed by an editor, so they can be transmission agents for viruses. In general, it is safer to force users to open files on their own rather than automatically.

In the simplest case, a virus inserts a copy of itself into the executable program file before the first executable instruction. Then, all of the virus instructions execute first; after the last virus instruction, control flows naturally to what used to be the first program instruction. Such a situation is shown in Fig. 6.9 (Pfleeger et al. 2015). It should be mentioned that this kind of attachment is simple and effective because the cyberattacker does not need to know anything about the program to which the virus will attach, and often the attached program simply serves as a carrier for the virus. The virus performs its task and then transfers to the original program.

Let's assume the cyberattacker wants to prevent the virus from being detected. He arranges for the virus to attach itself to the program that constructs the listing of files on the disk. If the virus regains control after the listing program has generated the listing but before the listing is displayed or printed, the virus could eliminate its entry from the listing and falsify space counts so that it appears not to exist. This is called a surrounding virus and is shown in Fig. 6.10 (Pfleeger et al. 2015).

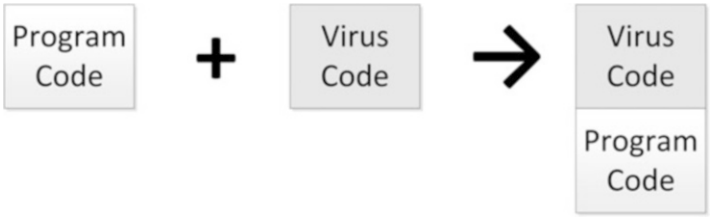

Fig. 6.9 Virus appended to a program code

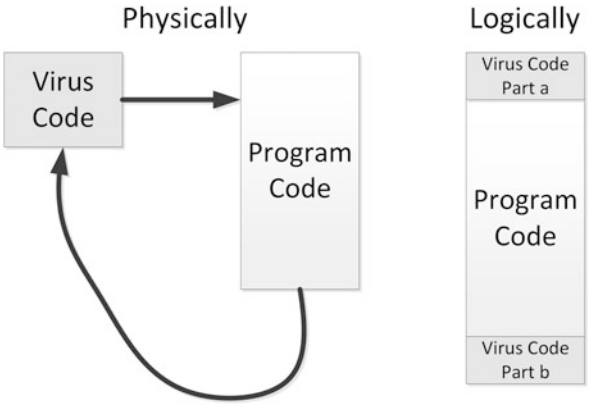

Fig. 6.10 Virus surrounding a program code

Fig. 6.11 Virus integrated into a program code

Finally, a third situation occurs when the virus replaces some of its target, integrating itself into the original code of the target. This situation is shown in Fig. 6.11 (Pfleeger et al. 2015) where the cyberattacker has to know the exact structure of the original program to know where to insert which pieces of the virus. Finally, the virus can replace the entire target, either mimicking the effect of the target or ignoring the expected effect of the target and performing only the virus effect. In this case, the user is most likely to perceive the loss of the original program.

The only way to prevent virus infection is to not share executable code with an infected source. Nevertheless, there are several techniques for building a reasonably safe community for electronic contact, including the following (Pfleeger et al. 2015):

- *Use Only Commercial Software Acquired from Reliable, Well-Established Vendors*: The good name of even highly reputed enterprises which have significant reputations could be seriously damaged by even one bad incident, so they go through some degree of trouble to keep their products virus free and to patch any problem-causing code right away. Similarly, software distribution companies are careful about products they handle.
- *Test All New Software on an Isolated Computer*: Test new software first on a computer with no hard disk, not connected to a network, and with the boot disk removed. Run the software and look for unexpected behavior. Test the computer with a copy of an up-to-date virus scanner created before running the suspect program. Only if the program passes these tests should it be installed on a less isolated machine.
- *Open Attachments Only When Knowing Them to Be Safe*: An attachment from an unknown source is of questionable safety. Also an attachment from a known source but with a peculiar message may not be trustworthy.
- *Make a Recoverable System Image and Store It Safely*: This clean version will allow a secure reboot because it overwrites the corrupted system files with clean copies. For this reason, the image has to be kept write-protected during reboot. For safety reasons, an extra copy of the safe boot image may be helpful.
- *Make and Retain Backup Copies of Executable System Files*: In the event of a virus, the infected files can be removed and reinstalled from clean backup copies (stored in a secure, offline location).

- *Use Virus Detectors (Virus Scanners) Regularly and Update Them Daily*: Many of the virus detectors available can both detect and eliminate infection from viruses. Several scanners are better than one because one may detect the viruses that others miss. Scanners search for virus signatures; they are constantly being revised as new viruses are discovered. New virus signature files, or new versions of scanners, are distributed frequently. Virus detector signature files should be kept up to date.

As more highly technological devices are introduced to the public, the more the demand for security rises. For this reason, various security schemes have been proposed, such as:

- Anomaly detection
- Probabilistic dependence graph
- Smart tracking firewall

Anomaly detection is a method of detecting anomalous behaviors or data. It mainly focuses on detecting intrusive methods based on their anomalous activities, those that are outside of the regular activity profile in a system. There are several possible approaches to tackling this challenge. The first approach is to focus on the behavior of insider cyberattacks and the design of new anomaly detection methods which utilize solid models of what acceptable behavior is and what a cyberattack is, thereby avoiding a high number of false-positive alarms. They may be caused by typical behavior that is actually normal and authorized, since normal behavior may easily and readily change (Dilek et al. 2015). Other limitations refer to the following properties (Barika et al. 2010; Bitter et al. 2010; Patel et al. 2010):

- Anomaly detection has to be able to characterize normal patterns and create a model of normal behavior; wide-ranging training sets of normal system activities are needed. Any change in a system's normal patterns must lead to a necessary update of the knowledge base.
- If intrusion detection and prevention inaccurately classifies a legitimate activity as a malicious one, the results can be very unfortunate since it will attempt to stop the activity or change it.
- Intrusion detection, no matter how efficient, may be disabled by cyberattackers if they can learn how the system works.
- In heterogeneous environments, there is an issue of integrating information from different sites.
- Another problem involves supplying intrusion detection that will conform to legal regulations, security requirements, and/or service-level agreements in the real world. First, however, the intrusion method must be identified so that the regular operation of the cyber-physical system will remain undisturbed.

The second approach in anomaly detection is to not to revise the existing anomaly detection techniques but to build upon them using novel game theory techniques to

exploit the inside intruder's weakness, in particular the fear of detection (see Sect. 6.1.5). But cyber criminals have always new ideas to disguise harmful data and overcome network protection measures. In this regard they use advanced bypass methods to deliver exploits or other malicious content to a vulnerable destination in a way that makes traffic seem normal and pass through security controls. Because multiple log levels are used which allow to overcome easily most security solutions.

The dependence graph (see Sect. 6.1.6) is a directed graph representing the dependencies of several nodes toward each other. For a given a set of nodes S and a transitive relation $R \subseteq S\ x\ S$ with $(a, b) \in R$, modeling a dependency a needs b to be evaluated first. Hence, the dependency graph is $G = (S, T)$ with $T \subseteq R$ and R as transitive closure of T. Fault detection and localization in systems are methods with which dependability can be measured to ensure a secure function. However, fault event diagnosis systems are not equipped, in any case, to detect fault events due to malicious attacks or naturally occurring events. To resolve these issues, a probabilistic graphical approach can be used that spatially correlates information from the systems and statistical hypothesis testing. A Gaussian Markov Random Field (GMRF) can be used to model a system's random variables and study their dependencies. The dependence graph illustrates the connections using a Markov Random Field (MRF) that is induced by a minimal neighborhood system by inserting an edge between sites that are neighbors. The Gaussian random variables can then be used to approximate fault diagnostics due to malicious intrusions (Landrum et al. 2014).

A smart tracking firewall is a security method for preventing intrusions by malicious nodes that infiltrate a secure wireless mesh network, which is a communication network made up of nodes organized in a mesh topology. It is also a form of wireless ad hoc network. The mesh nodes in the network have the ability to locate and deposit previously intruded nodes into either a blacklist or a graylist. A node blacklisted by a client cannot communicate with the client by either sending or receiving information. A mesh node can archive a malicious node into the graylist when neighboring nodes send alerts about a blacklisted node (Landrum et al. 2014).

Besides the common means of virus activation through an attachment to an e-mail message, the spear-phishing attack is a real particularly perfidious new cyberattack form. This is a mail to a recipient that looks like a message from a friend or colleague. It may, for example, point to a topic field on which the addressee is currently working on and the mail simply refers to the name of a study or publication that may be of interest to the addressed. The e-mail received does not contain a link to a specific page on the Internet nor an attached PDF document, only a final short greeting. In this form of a cyberattack, the attacker knows not only the personal mail address of the attacked person but also details from the attacked person's personal and professional settings, which may easily be filtered out of the so-called social networks, since many people today surrender a lot of themselves. In this way, a cyberattacker succeeds in establishing the identity of a friend or colleague, so that the mail appears completely harmless, which makes it almost impossible to recognize the attack. Since the mail contains no link and no file, the addressed person may google the note mentioned in the mail. In this way, the attacked person accesses

a page prepared by the attacker in which the spy software installed by the attacker is deposited, which from now on scans data from the then infected computer without the attacked person's knowledge. Often the affected person only notes months later that his computer was hacked.

6.1.8 Automotive IT and Cybersecurity

The automobile industry is currently undergoing an unprecedented wave of innovation, as automakers are pioneering innovative technologies that make vehicles safer than ever before. Besides this, the automobile industry is also undergoing a radical transformation from the traditional automaker's business into a digital electronic component manufacturer's business, enhancing and creating new features. This so-called digital transformation is not only redefining business within the automotive industry but is also expanding automotive industry boundaries. Competition is global, and digital technologies have provided resources to go after new opportunities. The reason for this lies in the effective delivery of digital services which requires:

- Transition from a product-centric approach to an ecosystem-centric one
- Seamless integration across different industries, leading to cooperation or coexistence of competition and cooperation

Therefore, automakers, in addition to their automotive products, may have to collaborate with various stakeholders to create a connected vehicles ecosystem, as the stakeholders include:

- Device/component or system manufacturers
- Insurance providers
- Service operators
- Telecommunication operators
- And others

Furthermore, products and services, information, and customer expectations can all be reshaped using new capabilities for mobility, interactivity, and information access. Moreover, connected vehicles become interlinked together through smart devices, such as smartphones, tablets, roadside units (RSU), and others. In the near future, it is predicted that innovative vehicle services will be offered, such as:

- Adaptive cruise control
- Autonomous driving
- Crash avoidance systems

These require connected vehicles, so-called vehicle-to-X (V2X) communication features, such as:

- *Vehicle-to-Infrastructure (V2I)*: This is a concept in which vehicles and roadway infrastructure exchange safety and operational data. In this approach, wireless communication occurs between vehicles and infrastructure, such as smart traffic signals, RSUs (see Sect. 6.5.3), and others.
- *Vehicle-to-Mobile (V2M)*: Technology that uniquely integrates wireless and cellular networks to facilitate intelligent transportation systems applications, such as the AGORA versatile framework for the development of intelligent transportation system applications (Salahuddin and Al-Fuqaha 2013).
- *Vehicle-to-Vehicle (V2V)*: Technology allowing vehicles to communicate with each other. V2V is also known as a vehicular ad hoc network (VANET), a variation of the mobile ad hoc network (MANET), and helps drivers to overcome blind spots, avoid accidents, and other serious dangerous situations.

All key components of intelligent transportation systems (ITS), these components do raise a number of issues/questions with regard to:

- How the IoT is affecting connected vehicles and how to detect and defend against malicious data intrusions.
- How functional safety and security are meshing and becoming more intertwined and what it means for future collaborative developments for the automotive manufacturing companies and their Tier 1 suppliers with regard to securing inter domain communication while trustworthiness is needed for cooperation.
- Cyber risks from the view point of a litigator pursuing a class action law suit.
- Key developments in regulations on data privacy and security across the whole life cycle.
- Security solutions in advanced network architectures and encryption methods which includes vulnerability and incident handling.
- Strategies to properly secure automotive telematics and infotainment systems.

A diagram depicting the digital transformation in vehicles and the associated security challenges is shown in Fig. 6.12.

With regard to the aforementioned issues, the automobile industry is also facing emerging challenges in the area of cybersecurity. The members of the Alliance of Automobile Manufacturers (Auto Alliance) and the Association of Global Automakers believe that by proactively and collaboratively addressing potential cybersecurity challenges, the automobile industry can continue to produce safe vehicles that incorporate modern and robust security options. But defending against cyberattacks often requires collaborative engagement between multiple stakeholders. There are benefits to building partnerships across the vehicle ecosystem, including sharing of cyber threat trends and proven techniques with third parties to defend against cyberattacks which require trustworthiness to support confidence about security levels of involved partners.

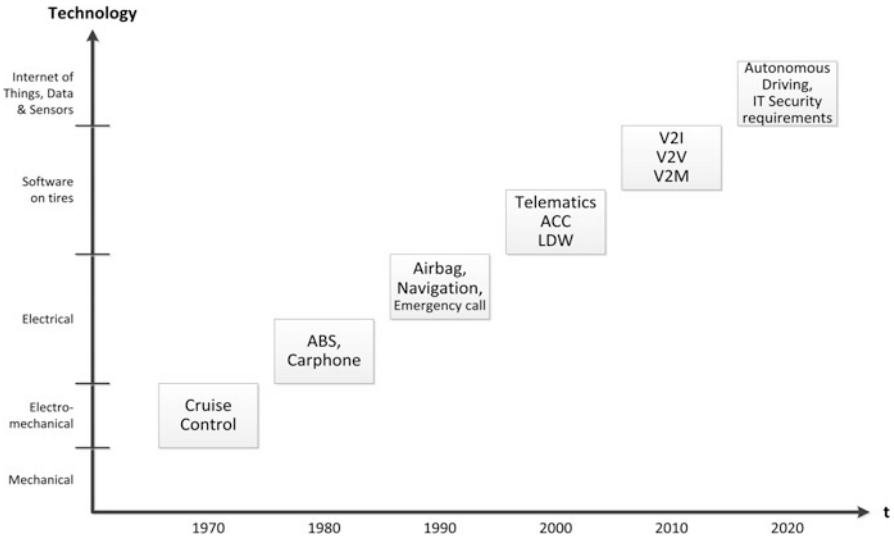

Fig. 6.12 Digital transformations in vehicles and associated security challenges

As written in the "Framework for Automotive Cybersecurity Best Practices" (URL2 2016):

> "*. . .an incident response plan documents processes used to help respond to cybersecurity incidents affecting the motor vehicle ecosystem. A comprehensive response plan that develops increased awareness and capabilities and that establishes communications protocols between automotive manufacturers, suppliers, cybersecurity researchers, and government agencies could assist industry stakeholders in coordinated efforts to address discovered vulnerabilities and enhance product security.*
>
> *The forthcoming best practices aim to address incident response plans that may include processes to activate response teams, notify an internal chain-of-command, and trigger response activities to assess and counter cyber attacks. A comprehensive incident response plan provides strategic flexibility for managing many types of cyber incidents and takes into account internal resources and, where appropriate, external resources likely needed to support incident response measures.*
>
> *The development of protocols for recovering from cybersecurity incidents is also important for ensuring consistent approaches for making available updates to vehicles in a reliable and expeditious manner based on specific circumstances*".

Therefore, security in vehicles in general is a challenge for automakers today because it is a moving target:

- As more smart digital devices connect to each other, the public's dependency on them is creating more movement toward connecting them with mobile vehicles, such as cars, buses, trucks, trains, aircraft, etc.
- Safety and security issues related to automotive objects are becoming more and more relevant in the realm of Internet-connected devices and objects which require long term capable security.

Security evaluation methods are needed for identifying and removing software security vulnerabilities as well as vulnerability analysis and intrusion detection and prevention systems with regard to security governance and risk posture. Connected vehicles may be targets for cyberattacks because:

- Vehicles are frequently parked in unsecured locations.
- Vehicles can be used to inflict serious bodily injuries.
- Vehicles could be targeted for antisocial activities such as terrorism.

Thus, growing needs exist to understand and address technology and policy issues around cybersecurity with regard to embedded wireless connected technology in vehicles. In addition, today's vehicles are becoming more and more equipped with intelligent electronic control modules which support drivers in tremendous ways, ranging from simple functions such as:

- Dashboard modifications
- Navigation
- Streaming of personal music via smartphones, as well as customized media content
- Vehicle adjustments

to

- Semiautomated driving on highways

to

- OEMs which are competing with each other to integrate the most up-to-date features emerging from the consumer electronics industry, as well as connectivity solutions that enable valuable remote services with their inherent security problems.

However, autonomous driving technologies are increasing the demand for continuous connection of the vehicle's ECUs to a variety of cloud services that would help to improve advanced processing and subsequent vehicle maneuvering strategies, accompanied by the possibility of distributing new software updates and other essential content into every ECU or infotainment system. Hence, ubiquitous internal and external connectivity is undoubtedly the gatekeeper for future needs and possibilities within the automotive industry with regard to security design.

All of these advancements are calling for vehicle cybersecurity, a problem which is not trivial with regard to specific requirements, such as speed, real-time constraints, etc., and contradictory expectations. Industrial standards are still under development, such as IEEE P1556: Security and Privacy of Vehicle and Roadside Communications Including Smart Card Communications. Today, communication is typically done over dedicated short-range communications (DSRC) at the 5.9 GHz level based on the IEEE 802.11p protocol. Therefore, one problem of vehicle

Table 6.7 Vulnerable access points

Communication Channel hacking	In-vehicle hacking	Remote hacking
RFID keys: Embedded with RFID tag and a reader in the vehicle. Vehicle can be immobilized if the correct tag is not verified	**CDs and USB connectivity, and physical interface for entertainment devices:** Entertainment Systems and CAN bus connectivity to update ECU firmware interface with systems within the vehicles	**Cellular/telematics connectivity units:** Equipped with connectivity used for various functions. Provides access to internal network and ECU
Keyless entry: Remote keyless entry used to open doors and activated alarms can be blocked by interfering transmitters allowing access to vehicles	**ODB II port:** Provides a regulated access to CAN buses to control key components	**Dedicated short-range communication (DSRC):** Emerging technology proposed standard for cooperative driving. Can potentially transmit malicious inputs to other vehicles causing damage
Tire pressure monitoring system (TPMS): Alerts drivers about tire pressure readings; can be manipulated showing inconsistent readings		**Wi-Fi hotspots:** Make vehicle's OBD II port vulnerable to attacks by connecting wirelessly
Bluetooth: Used as standard supporting hands-free callings. Paired with phones it can be a medium for downloading malware		

cybersecurity lies in the advancements in malicious methods and tools emerging in traditional ICT environments which can be applied to automotive systems with no additional cost or effort and which can be a significant threat to safety. Elements such as automotive-specific vehicle communication buses do not offer robust protection against advanced attack vectors. Hence, in Table 6.7, vulnerable access points are summarized with regard to the chosen attack method.

With regard to Table 6.7, cyberattacker's methods of attacking vehicular communication can be manifold because a cyberattacker can:

- Attack against liability-related messages by cheating with its own identity, position, speed, etc.
- Be an inside or an outside cyberattacker, whereby the insider has to be prevented from cheating about its own position; and the outsider has to be prevented from spoofing the position on an honest traffic node to secure positioning.
- Disrupt network operation which results in a denial-of-service attack.
- Intrude bogus information against traffic information, such as "a traffic jam is ahead."
- Undefended uncovering of identities of other vehicles.

In cases where vehicles carry a certified identity and public key, such as an electronic license plate (ELP), mutual authentication can be done. Authorities are able to cross-certify a vehicle's position by using verifiable multilateration for vehicle identification, as is used in aviation. Multilateration is a surveillance application that accurately establishes the position of transmissions, matches any identity data that is part of the transmission, and sends it to the air traffic management (ATM) system. Multilateration is considered to be a cooperative surveillance technique, combining a dependence on target-derived data for identification and altitude with ground-based calculation of position (URL3 2016). Thus, using this secure, verifiable multilateration (triangle) positioning technique in the automotive domain (Hubaux et al. 2004) results in the following:

- A vehicle located within the triangle cannot prove to be at another position within the triangle except at its true position.
- A vehicle located outside the triangle formed by the verifiers cannot prove to be at any position within the triangle.
- An outside adversary cannot spoof the position of a vehicle such that it seems that the vehicle is at a position different from its real position within the triangle.
- An outside adversary cannot spoof the position of a vehicle such that it seems to be located at a position within the triangle, if the vehicle is out of the triangle.

6.1.9 Attack Value Chain

The latest and greatest advances in technology have created greater efficiency and effective for all kinds of industries. However, the pace of data breaches and intrusions into secure industrial systems, such as computers and communication networks, is accelerating at an alarming rate. The present risks and potential new avenues of compromise and increasing sophistication of intruders are making computers and communication networks more vulnerable. To manage these risks, automakers must enhance and standardize their security procedures including vendors, partners, and even customers looking for potential weaknesses in the attack value chain to secure their own as well as suppliers products and services, as shown in Fig. 6.13. Viruses or malware carried in a smartphone or in an infotainment system can easily invade automotive electronics. Therefore, cybersecurity in vehicles has to maintain zero compromise on security, preventing costly and massive cyber-attack-caused recalls.

Moreover, an unsecure implementation of communication protocols within an infotainment system (see Fig. 6.13) can lead to a remote access tunnel for cyberattackers who are then able to remotely deactivate critical safety elements, such as steering and braking systems, during driving. These types of attacks can be triggered from any place outside the system at any time. Other vulnerable vehicles can be identified with simple ICT methods with regard to an unsecure configuration of a mobile network provider. Therefore, efficient security features within systems and/or components are required which protect against critical threats assumed to

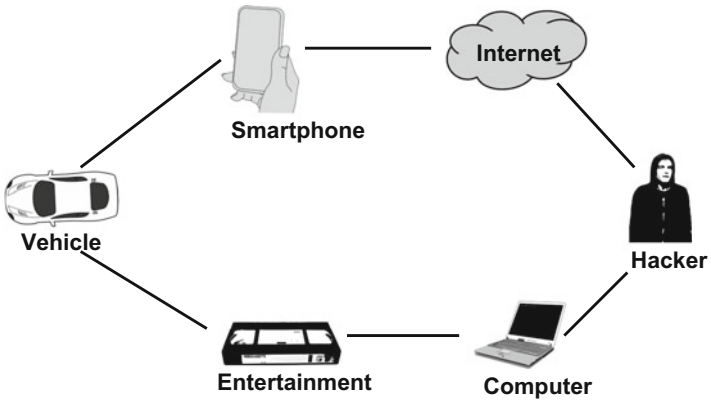

Fig. 6.13 Attack value chain in vehicles

happen. Furthermore, important documentation and source code for securing elements of the vehicle's electric backbone could be bought with minimal effort.

Today's vehicle systems are not designed to continuously upgrade software and hardware to integrate or update security elements. While this could prevent some attack vectors, it could require more processing power than what is available for sustainable testing and validation efforts. Mission-critical ECU components, such as system-on-a-chip (SoC) designs, have appropriate cybersecurity extensions already in place, albeit at a significantly higher cost. It can be assumed that newer software features would increase the cybersecurity level in the automotive domain, such as virtualized or vehicle bus message encryption, which relies on certain hardware-based elements that are not yet integrated into today's vehicles.

Original equipment manufacturers (OEMs), Tier 1 suppliers, and other contributors of complex vehicle cyber-physical systems are facing the same issues with successful and efficient integration of cybersecurity elements. At first glance, the implementation of cybersecurity elements within the automotive industry is seen purely as an additional cost which doesn't innately entice more customers to buy a specific car. Unfortunately, this results in poor integration of security functions over the long term, which will add further costs, both directly and through costly fleet-wide recall events. Thus, cybersecurity is certainly an opportunity for OEMs and Tier 1 suppliers with strong bottom-line implications. In this regard the design and manufacturing of vehicular components and systems as well as vehicles itself require to follow a new design and manufacturing paradigm, which can be stated as security by design, as it has been introduced by the German Industry 4.0 Platform (URL1 2018). The risk of paying penalties or recall costs for security-related issues may soon be as real as costs related to vehicle safety. The primary subcategories of threats to vehicles can be summarized as follows (Bittersohl and Thoppil 2015):

- *Compromised Privacy*: Interception or readout of privacy-related user data that is directly connected with personal details of drivers, which can be stored within cloud services, etc., such as:
  - Billing details
  - Destination targets
  - Driving behavior
- *Dysfunctional Sensor Processing*: Disturbing sensor input for further processing of vehicle maneuvers through modification of bus communication systems or unauthorized software modification directly to ECUs
- *Man-in-the-Middle Attack (MITM)*: Interception of internal and/or external vehicle communication in order to obtain information from ECU-to-ECU communications or other mission-critical software elements
- *Side-Channel Attack*: Utilizing weaknesses in hardware, software, and communication protocols in a system connected to a cyberattack target in order to open an unprotected channel
- *Spoofing*: Faking the presence of communication partners and information that is used to control advanced sensor systems, thereby activating maintenance functions within vehicles and creating new possibilities to modify a vehicle system's configuration

Recent reported events have shown that security breaches into vehicles are sophisticated cyberattacks combining several attack vectors, as shown in Table 6.8. Therefore, the objective is to find the weakest links of the integrated cyber-physical systems. Infotainment systems are often identified as the ideal target as they are based on highly complex and modern operating systems. The huge volume of software code implemented for features such as navigation, radio, video/audio, and external content makes the effort of maintaining secure code more and more complicated if not unmanageable, especially due to the high amount of individual internal or external third-party partners working on such projects. In addition, the integration of vehicle communication protocols and layers into the infotainment system further increases the threat of a cyber attacker gaining access to the more mission-critical elements of the vehicle, which can ultimately result in denial of service (Bittersohl and Thoppil 2015).

Without a doubt, it took significant effort to identify the vulnerabilities essential to achieve far-reaching access to vehicles. Henceforth, research will be based on a combination of different attack categories with the objective of discovering the weakest link within each subdomain. Prior to now, OEMs had no other choice but to recall vulnerable vehicles as the functionality of a remote over-the-air (OTA) update was not yet implemented.

6.1.10 Holistic Cybersecurity Solutions

Despite efforts to protect vehicle cyber-physical systems against cyberattacks, the attacks are growing in number and sophistication. This indicates that a change in the

Table 6.8 Attack value chains

Attack vector value chain	Critical element of attack vector part	Attack category	Vehicle attack targets
Critical communication systems not protected for external access (e.g., Wi-Fi, 4G).	Direct access to critical vehicle communication	Man-in-the-middle	Vehicle bus communication
	Elements with consumer electronic devices		Traffic control unit
Easy access to operating system images and decompiling of software components	Publicly available software images with/without encryption	Side-channel	Infotainment
Modified operating system image transferred to infotainment system without security integrity check for unauthorized modification	Unauthorized software modifications possible.	Side-channel	Comfort systems
			Traffic control unit
Readout of cellular network configuration and identification of potential targets	Extraction of critical infrastructure data	Spoofing	Vehicle Wi-Fi
		Compromised privacy	V2X (DSRC)
			Smartphone
			Connected services
Modifying CAN chip software through reflash order to send unauthorized messages to other critical ECUs in vehicles	Unprotected external developer/ diagnostic tools	Side-channel	
	No message authentication		
Utilizing publicly available diagnostic tools to reverse engineer CAN messages and unlock ECU encryptions.	Effortless decompilation of message protocols.	Vehicle communication bus manipulation	CAN
	No device authentication within vehicle bus system		FlexRay

defense strategy is required as well as acceptance of the fact that there is no panacea to overcome the ever-growing plethora of cybersecurity problems. Thus, a holistic security approach can be used which suggests system administrators look at the full picture and make a thorough analysis of the security threats to the whole system instead of securing it part by part, using a multilayer approach (Shiva et al. 2010):

- *First Layer*: Core hardware and software components. Envision each of these components as being wrapped with a self-checking module, called self-checking hardware/software components
- *Second Layer*: Traditional network security infrastructure built using techniques such as cryptographic algorithms

- *Third Layer*: Secure applications designed with built-in or built-on security approaches utilizing self-checking concepts and components
- *Fourth Layer*: Game theoretic decision module which has the responsibility of choosing the best security strategy for all three inner layers

In the past, research efforts have focused on the second and third layers. Thus, a traditional intrusion detection system (IDS) can be considered as residing in the top layer, which can be made more effective by use of game theory (see Sect. 6.1.5).

Growing distribution of common software, such as AUTomotive Open Source ARchitecture (AUTOSAR) or GENIVI, provides a basis for achieving a holistic cybersecurity approach with backend services. However, the final decision regarding which cybersecurity feature should be integrated depends almost entirely on the OEM. Costs and supplier readiness are main decision drivers for each technology as well as OEM organizational readiness with regard to developing and adhering to cybersecurity policies and guidelines.

6.1.10.1 AUTOSAR

AUTOSAR is a worldwide development partnership of automotive interests founded in 2003 to create and establish open, standardized software architecture for automotive ECUs, excluding infotainment (see Sects. 4.6 and 4.7).

Development goals include scalability to different vehicle and platform variants, transferability of software, consideration of availability and safety requirements, collaboration between various partners, sustainable utilization of natural resources, maintainability throughout the whole product life cycle, and process managing the entire life cycle of a product from inception, through engineering design and manufacturing, to service and disposal of manufactured products.

AUTOSAR is driven by the advent of innovative vehicle applications, contemporary automotive electrical/electronic (E/E) architecture that has reached a level of complexity requiring a technological breakthrough in order to manage it satisfactorily and fulfill the heightened passenger and legal requirements. This need is important for vehicle manufacturers and their leading Tier 1 suppliers who are faced with often conflicting requirements from:

- *Driver Assistance and Dynamic Drive Aspects*: Key items include detection and suppression of critical dynamic vehicle states and navigation in high-density traffic surroundings.
- *Legal Enforcement*: Key items include environmental aspects and safety requirements.
- *Passenger Convenience and Service Requirements*: Comfort and entertainment functional domains.

Leading OEMs and Tier 1 suppliers, having recognized this industry-wide challenge, decided to work together to meet the challenge. Their common objective is to create a development base for industry collaboration on basic functions while providing a platform which continues to encourage competition on innovative

functions. To this end, a development partnership called AUTOSAR was formed, including all vehicle domains with the goals of (URL4 2016):

- Collaboration between various partners
- Definition of an open architecture
- Development of highly dependable systems
- Scalability to different vehicle and platform variants
- Standardization of basic software functionality of automotive ECUs
- Support of different functional domains
- Support of applicable automotive international standards and state-of-the-art technologies
- Transferability of software

The AUTOSAR standard serves as a platform upon which future vehicle applications will be embedded and also serves to minimize the current barriers between functional domains. It will, therefore, be possible to map functions and functional networks to different control nodes in the system, almost independently from the associated hardware.

The technical goals of AUTOSAR:

- Modularity of automotive software elements to enable tailoring of software according to the individual requirements of ECUs and their tasks
- Reusability of functions to help improve product quality and reliability and to reinforce corporate brand image across product lines
- Scalability of function to ensure the adaptability of common software modules to different vehicle platforms and prohibit proliferation of software with similar functionality
- Transferability of functions to optimize the use of resources available throughout a vehicle's electronic architecture

This will help to provide a common software infrastructure for automotive systems of all vehicle domains based on standardized interfaces for the different layers, as shown in Fig. 6.14. This common infrastructure will be comprised of the following elements:

- *Electronic Control Unit (ECU)*: The physical hardware.
- *Runtime Environment (RTE)*: All communication between software components and basic software, including the operating systems (OS) and communication services, is carried out through the RTE layer.
- *Main Software (MSW)*: A combination of:
  - *Basic Software*: Builds on RTE to provide some general utilities which provide the overall functionality of the AUTOSAR infrastructure (software components and RTE on an ECU). Basic software is essential for running the functional part of the software; however, it does not fulfill any functional job itself. The software components do that.

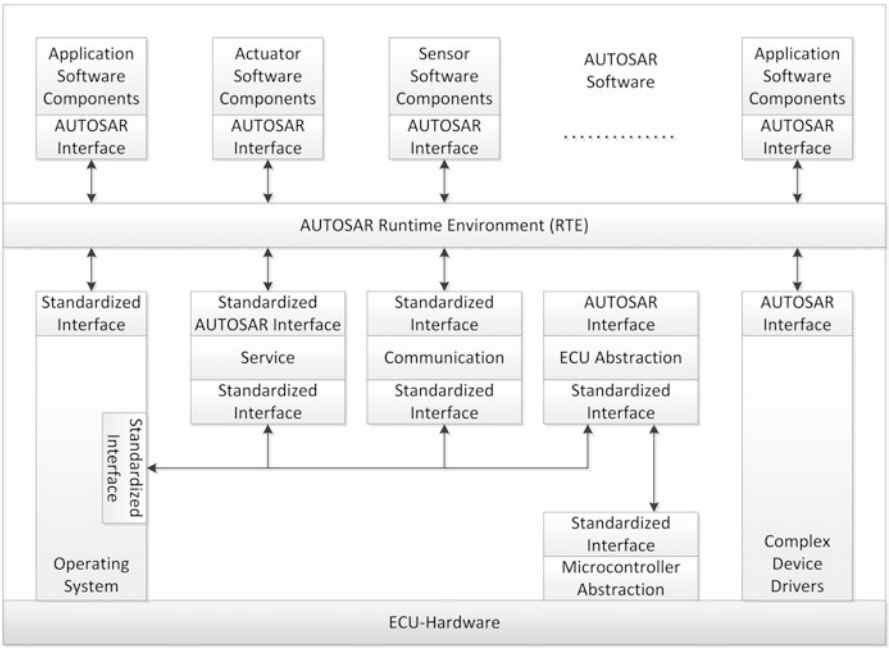

Fig. 6.14 AUTOSAR ECU software architecture (source: www.autosar.org)

  - *Software Components*: Base of any software assembly is the implementation of parts of the functionality of the automotive application. Software components are the fundamental building blocks of AUTOSAR systems. Types of software components are:
    Application software components
    Actuators/sensor software components
- *Complementary Software (CSW)*: Manufacturer- and model-specific software.

Hence, standardization of functional interfaces across automakers and suppliers and standardization of the interfaces between the different software layers is seen as a basis for achieving the technical goals of AUTOSAR. AUTOSAR provides a standard description format for the interfaces as well as other aspects needed for the integration of the AUTOSAR software components.

The constantly growing complexity of software also increases the specific needs for the network infrastructure in a vehicle. Therefore, in addition to the standard CAN bus, other bus systems have been embedded in vehicles. The use of these bus systems is a challenge for automakers and their suppliers as they seek to protect their systems against cyberattacks. The main focus here is on cyberattacks at the protocol level which may result in the following, which always result in paralyzing the ECUs.

- Denial-of-service attacks
- Falsification of sender addresses using Internet Protocol (IP) spoofing
- Redirection of network traffic using Address Resolution Protocol (ARP) spoofing

IP spoofing (see Table 6.6) is a technique used to gain unauthorized access to computers, whereby the cyberattacker sends messages to a computer with a forged IP address indicating that the message is coming from a trusted host.

Within the Internet Control Message Protocol for IPv6 (ICMP-IPv6), functionalities like ARP are directly integrated. Thus, information about how to assign an IPv6 address or the way a controller sends its data are transmitted unprotected on the network. Hence, a cyberattack is theoretically simple, with an attacker able to impersonate the router and redirect network traffic to read it or change the content. Firewalls and IDSs can, with regard to the required computing power, only protect against such cyberattacks to a limited extent. This requires protocol extensions (Finke et al. 2015), such as the secure neighbor discovery (SEND) protocol, which is a security extension of the neighbor discovery protocol in IPv6 defined in RFC 3971 and updated by RFC 6494. SEND uses cryptographically generated addresses and other new neighbor discovery protocol options for the ICMP-IPv6 packet types used in neighbor discovery protocol (URL5 2016).

AUTOSAR considers that due to V2X applications, the requirement that vehicles interact with off-board systems will enhance the integration of non-AUTOSAR systems; and support of cloud interactions will be the next challenge that AUTOSAR has to face. In such an open access environment to select vehicle systems, a dedicated means of security is required with regard to:

- Architecture
- Cloud interaction
- Onboard communication

This will improve the existing standard, support new technologies, and enhance dynamic security architectures.

6.1.10.2 GENIVI

Compared to AUTOSAR, the nonprofit GENIVI Alliance is committed to driving the broad adoption of specified, open-source, in-vehicle infotainment (IVI) software. Therefore, GENIVI provides automakers with four unique approaches to meeting today's challenges:

1. *Define*: Allows flexible definition of IVI systems that fit customers' latest needs
2. *Partner*: Supports business model evolution and networking across the supply chain
3. *Leverage*: Provides standard, open-source architectures, tools, and software components
4. *Reuse*: Allows reuse of components and redeployment of solutions with no royalty fees

Automakers and their suppliers face at least three significant challenges in developing and delivering IVI functionality to their customers (URL6 2016):

- *Responding to Consumers*: Consumers want IVI functionality that is the same or similar to that found in consumer electronic devices, such as smartphones and tablets. New devices with the latest features are typically launched in the market on an 8- to 18-month cycle versus the 2–5 year cycle for most in-vehicle software. As a result, consumers have introduced a new competitive measure that automakers must use: the time from consumer request to in-vehicle availability.
  - GENIVI's open software approach better aligns consumer electronics and automotive development cycles.
  - GENIVI's individual software components and reusable platform provide automakers and their suppliers with the tools to perform rapid prototyping and to quickly develop and deliver IVI systems that fulfill consumer requests.
- *Complexity and Cost*: Consumer functionality requests push the amount of software in a typical IVI system to over several million lines of code. Hence, automakers have to deal with the increasing complexity and cost of developing, validating, and maintaining software. Many automakers are shifting away from the historical black box approach and are taking more ownership of the design and development process, including maximizing the reuse of legacy code to reduce costs and deploying a software platform on multiple hardware platforms based on the needs of their various models.
  - GENIVI's technical deliverables and open approach promote a wide range of supplier models based on the preferences of the automaker.
  - Automakers can launch a single reusable software platform that with limited integration can run on a wide range of automotive boards, from low- to high-end performance.
- *Customer Ownership*: Automakers are keen to keep their customer relationships sustainable. Large technology companies, such as Apple and Google, have entered the automotive market, introducing demands for user experience, branding, and data usage that limit the automaker-driver relationship. Automakers have their own business model; some prefer a single Tier 1 supplier, while others prefer multiple suppliers taking ownership of certain pieces of the overall system.
  - GENIVI's approach allows automakers to maintain their independence from technology titans pushing their own business models in the automotive industry.
  - GENIVI's flexible architecture and pick-and-mix model give automakers the freedom to include preferred, best-in-class software from multiple suppliers.

GENIVI's technical deliverables consisting of:

- Flexible technical architecture
- Individual software components
- Preintegrated, reusable IVI platform
- Standard interfaces/application programming interfaces (APIs)

which are essential to overcoming the IVI challenges faced by every automaker. Thus, GENIVI technologies are at the forefront of a new generation of IVI solutions. As one of the many GENIVI use cases, BMW has changed from its traditional

approach to IVI software development to where it is today; the first automaker to deliver a complete infotainment product, the so-called entry media and navigation system (EMNS). The EMNS rolled off the assembly line in the fall of 2013 and is now part of the MINI and 1, 3, and 5 BMW series product lines based on the GENIVI Linux platform. Since then, other automakers have selected products with GENIVI solutions making the platform available in four continents around the world. Furthermore, several additional automakers will release GENIVI-equipped systems in their vehicles during the next 2 years.

6.2 IT Security in Automotive Cyber-Physical Systems

The rapid growth of information and communication technology (ICT) has prompted the expansion of networked systems that address real-world applications. This has led to the integration of computing and communication technologies with physical processes that incorporate CPS, which can be represented (see Sect. 6.1) more generally as shown in Fig. 6.15 by:

- Computed result of the physical system states which could advise the controller to select valid commands
- Control commands which are sent to actuators
- Data acquisition from sensors
- Physical data aggregation in the network

Cyber-physical systems capture novel aspects of networked systems, including integration of distributed computing systems with monitoring and control entities in the physical environment with regard to:

- *Actuating*: Executes various forms of actions determined during computing phases, such as correcting the cyber behavior of the CPS or component, changing the physical process, etc.

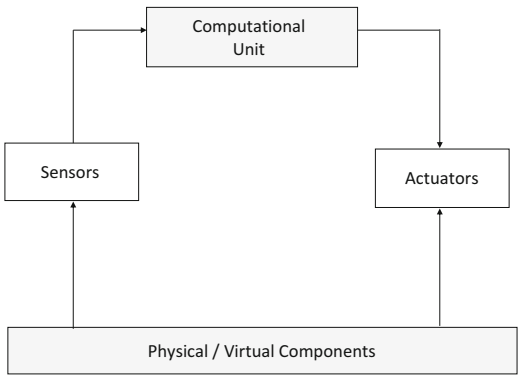

Fig. 6.15 Cyber-physical system architecture

- *Computing*: Reasoning and analyzing data collected during sensing/monitoring to check whether the physical process satisfies predefined constraints. If criteria are not being satisfied, corrective actions are proposed.
- *Networking*: Deals with real-time sensor node data aggregation/diffusion for process analytics. Different applications interact concurrently with networking communication.
- *Sensing*: Fundamental capability of a cyber-physical system giving feedback on any past actions which were taken by the cyber-physical system nodes, ensuring correct operation in the future.

Technological advances of CPSs have a tremendous impact on security vulnerability. Therefore, security is a relatively new realm of research. Like any other new field, most of the effort seems to be focused on mapping security solutions from existing domains onto CPS application needs. However, these solutions are usually not very well suited for CPSs because traditional security solutions were not designed for interoperation among heterogeneous applications. Thus the challenge is how to make sure that CPSs are secure while interacting with another system because major types of cyberattacks to CPSs intrude:

- Actuator devices
- Computing devices
- Networking devices
- Sensing/monitoring devices

These attacks are accomplished through (Wang et al. 2010):

- *Compromised Key Attacks*: A key is a secret code which is necessary to interpret secure information. Once a cyberattacker obtains a key, the key is considered to be compromised (Chalkias et al. 2009).
- *Denial-of-Service Attack*: A cyber criminal network attack that prevents legitimate traffic or requests for network resources from being processed or responded to by the system (Pelechrinis et al. 2011). This type of attack usually transmits a huge amount of data to the network making it too busy handling the data to provide normal services.
- *Eavesdropping*: A cyberattack where an adversary can intercept any information communicated by the system.
  - *Passive Attack*: Cyberattacker does not interfere with the workings of the system; it simply observes the system's operation (Kao and Marculescu 2006).
- *Man-in-the-Middle Attack*: False messages are sent to the operator, taking the form of a false negative or a false positive (Saltzman and Sharabani 2009).
  - *False negative*: A test result indicates that a condition failed when it was actually successful, i.e., erroneously no effect has been assumed.
  - *False positive*: A false alarm indicating that a given condition has been fulfilled when it actually has not been fulfilled, i.e., erroneously assuming a positive effect.

Prior work focused on:

- Actuating
- Computing
- Monitoring
- Networking
- Sensing

It focused on reliability and resilience in protecting CPSs against:

- Random independent or benign faults and failures of their cyber and/or physical components (Akella et al. 2010; Johnson 2010).
- Failure to adequately address integrity, confidentiality, and denial-of-service threats (Cárdenas et al. 2008; Cárdenas et al. 2011; Eisenhauer et al. 2006; Fleury et al. 2009; Mo and Sinopoli 2009).

However, conventional computer and network security approaches do not address, in a unified way, how systems outlive malicious cyberattacks which correlate with survivability or how they recover after a cyberattack, which refers to recoverability (Fleury et al. 2009). Thus, securing CPSs goes beyond securing the individual system components separately. Highly skilled cyberattackers use multivector attacks that exploit weaknesses of separate physical and cyber components of the attacked system, none of which may pose a serious threat for the corresponding component. The combined effect, however, may result in a catastrophic event if the attack vectors are dependent.

One of these multivector attacks was the Stuxnet attack (Falliere et al. 2011), which targeted the functions of industrial nuclear centrifuges used in Iran's nuclear program. In the Stuxnet attack, a worm that used zero-day exploits spread to machines using Microsoft® Windows® via local area networks (LANs) or universal serial bus (USB) sticks, carrying a malware payload that infected and reprogrammed programmable logic controllers. It is believed that Stuxnet possessed a broader panoply of cyber weapons.

Thus, there are many ongoing efforts to ensure the security of CPSs which are primarily based on extending mechanisms already used to protect separate cyber and physical components. However, there is no formal security model for cyber-physical systems that addresses security in a unified framework that deals with:

- Hardware threats
- Network threats
- Physical threats
- Software threats

There is a huge number of publications in the literature highlighting the difficulties of securing physical systems, with regard to timing attacks in particular (Fleury et al. 2009; Lamport 2005; Tang and McMillian 2008), noninterferences

(Gamage and McMillin 2009), and execution monitoring (Hamlen et al. 2006; Lamport 1997, 1998). Thus, to secure CPSs, it is important to understand cyberattacks and what can be done to prevent them from becoming successful. Hence, cybersecurity measures which address the risks expected to be present in a CPS or subsystem can be designed and implemented in such a way that access and operation to legitimate activities is not impeded, particularly during times of emergency or restoration activity.

The Institute of Electrical and Electronic Engineers (IEEE) has developed a cybersecurity standard that presents a balance of the above features, introduced as IEEE 802.11 Wireless Network Standard, which is one of the most attractive and fastest-growing networks. The IEEE 802.11 WLAN standard is extended by IEEE 802.11i, a Standard for Wireless Local Area Networks (WLANs), providing improved encryption for networks that use the popular 802.11a, 802.11b, which includes Wi-Fi, as well as 802.11g standards. The 802.11i standard requires new encryption key protocols, known as Temporal Key Integrity Protocol (TKIP) and Advanced Encryption Standard (AES). However, AES requires a dedicated chip, which means hardware upgrades for most existing WiFi networks. Other features of 802.11i are key caching for access, which facilitates fast reconnection to the server for users who have temporarily gone offline, and preauthentication, which allows fast roaming. The 802.11i standard was officially ratified by IEEE in June of 2004 and thereby became part of the 802.11 family of wireless network specifications. Since introducing WiFi, a variety of keys have been deployed:

- *Wired Equivalent Privacy (WEP)*: The first form of authentication used with Wi-Fi. Unfortunately, it was easy to crack, and other systems are now more widely used.
- *Wi-Fi Protected Access (WPA)*: A software/firmware improvement over WEP. The first version, it is also known as WPA1 or WPAv1.
- *Wi-Fi Protected Access II (WPA2)*: Next update to WPAv1, it provides significant improvement in the level of security.

Cybersecurity in the automotive industry refers to securing the manifold automotive ECUs. ECU is a generic term for any embedded system that controls two or more of the electrical systems or subsystems in a vehicle, connected through a CAN bus as shown in Fig. 6.16.

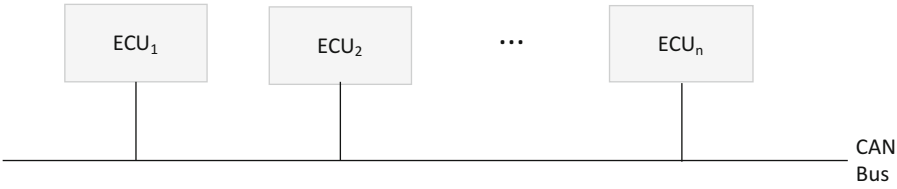

Fig. 6.16 ECUs connected to the CAN bus

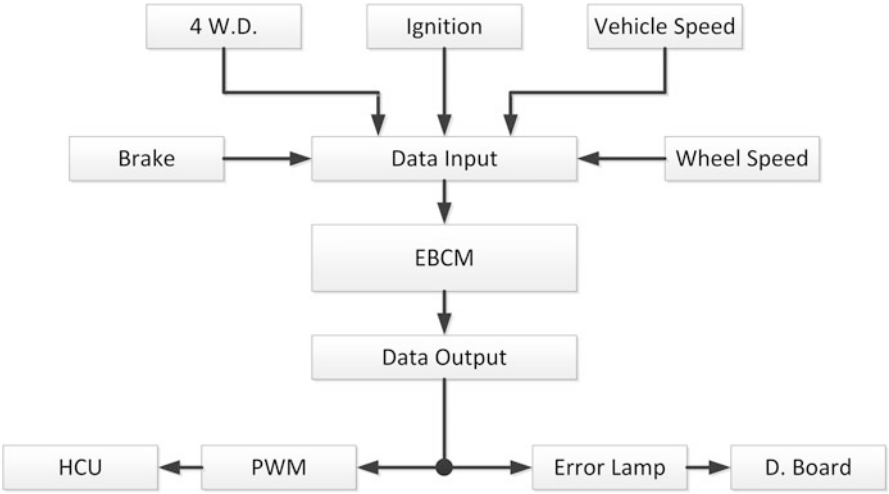

Fig. 6.17 EBCM block structure

The manifold ECU types used in vehicles includes:

- *Body Control Unit (BCU)*: Monitors and controls various electronic accessories in a vehicle's body.
- *Brake Control Unit or Electronic Brake Control Module (EBCM)*: Controls a vehicle's wheels to enhance braking ability on wet, slippery, or icy road surfaces. EBCM regulates the braking systems on the basis of five inputs, as shown in Fig. 6.17 (ni-com 2009).
  1. *Brake*: Input that monitors the status of the brake pedal, i.e., deflection or assertion. Information is acquired in a digital or analog format.
  2. *4W.D.*: Input that monitors the status in digital format as to whether the vehicle is in the 4-wheel-drive mode.
  3. *Ignition*: Input that registers if the ignition key is in place and if the engine is running or not.
  4. *Vehicle Speed*: Input that informs about the speed of the vehicle.
  5. *Wheel Speed*: Application representing a set of four input signals that conveys the information concerning the speed of each wheel, essential to derive all necessary information for the control algorithm.
  6. *HCU*: Hydraulic control unit is a unit in the antilock brake system that controls/regulates hydraulic pressure during an ABS stop.
  7. *PWM*: Pulse width modulation is used in applications such as switching mode voltage regulators, positional motor controls, fuel injector drivers, ignition drivers, and ABS control
  8. *Error Lamp*: Typically the first indicator that the EBCM is damaged and that the ABS system light will illuminate an error on the dashboard.

- *Central Control Unit (CCU)*: Scalable and modular control unit with embedded software; contains mostly internal diagnostics which enhance troubleshooting with distinct messages for fault conditions for the CCU.
- *Door Control Unit (DCU)*: Controls and monitors various electronic accessories in a vehicle's door. Since most of the vehicles have more than one door, DCUs are generally present in each door separately. The DCU associated with the driver's door has some additional functionality which is the result of complex functions, such as locking, driver door switch pad, child lock switches, etc. In most cases, a DCU acts as a master and others act as slaves in communication protocols. Features controlled by DCU are:
  - Automatic window movements
  - Child lock safety feature
  - Global open-close functionality
  - Manual window movements
  - Mirror adjustment
  - Mirror folding
- *Engine Control Module (ECM)*: Controls a series of actuators on an engine to ensure optimal engine performance by reading values from a multitude of sensors within the engine, interpreting the data using multidimensional performance maps (called lookup tables) and adjusting the engine actuators accordingly.
- *Powertrain Control Module (PCM)*: Consisting of the ECM and the transmission control unit, it commonly controls more than 100 factors in a vehicle. Inputs to the PCM come from many sensors, of different types, that are spread around the vehicle. Most of them are oriented toward engine management and performance.
- *Speed Control Unit (SCU)*: Controls the speed of a vehicle. An SCU is a servomechanism that takes over the throttle of the vehicle to maintain a steady speed as set by the driver.
- *Suspension Control Unit (SPCU)*: Responsible for keeping the steering knuckle in place. The steering knuckle connects the wheels to the suspension system, and it also contains the wheel hub or spindle.
- *Telematic Control Unit (TelCU)*: Controls tracking of the vehicle. The TelCU consists of a Global Positioning System (GPS) unit, which keeps track of the latitude and longitude values of the vehicle, an external interface for mobile communication (Global System for Mobile communications (GSM), Global Positioning System (GPS), long-term evolution (LTE) portable radio standard 4G, Wi-Fi), which provides the tracked values to a centralized geographical information system (GIS) database server, an electronic processing unit, a micro-controller to processes the information that also acts on the interface between the GPS, a mobile communication unit, and some amount of memory for saving GPS values in case of mobile-free zones or to intelligently store information about the vehicle's sensor data.
- *Transmission Control Unit (TMCU)*: Controls electronic automatic transmissions. A TMCU generally uses sensors from the vehicle as well as data provided by the ECU to calculate how and when to change gears in the vehicle for optimum performance, fuel economy, and shift quality. In some applications, the TMCU and the ECU are combined into a single unit as a Powertrain Control Module (PCM).

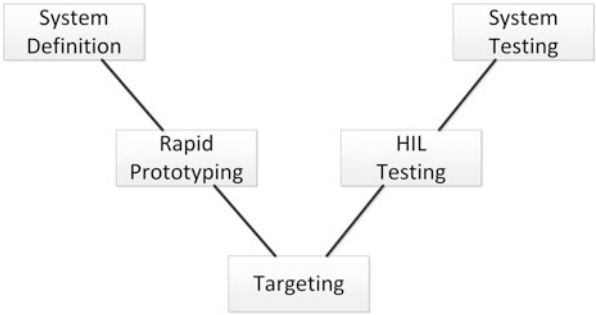

Fig. 6.18 V-diagram used in the ECU design cycle

Managing the increasing complexity and number of ECUs in a vehicle has become a key challenge for automakers and OEMs as modern vehicles have up to 100 ECUs. Moreover, embedded software in ECUs continues to increase in line count, complexity, and sophistication and has reached more than seven million lines of software code today, requiring specific concepts, methods, techniques, and tools for testing security in automotive software in ECUs. Once the code containing the control algorithm is downloaded to the ECU, performance testing of the ECU can be done under extreme conditions, which cannot be achieved in the real world, by performing hardware-in-the-loop (HIL) simulation. In this step, the actual ECU is tested by simulating an engine using the created engine model. In HIL, the software model of the engine is downloaded to real-time hardware; and the appropriate input/output (I/O) interfaces are provided. These I/Os are then connected to the ECU under test. Then various engine conditions can be simulated; and the ECU can be tested to its limits, above and beyond any real-life capabilities of a real engine. The documentation can be done by using any word processing or spreadsheet application. The design process follows the V-model, shown in Fig. 6.18 (ni-com 2009).

6.2.1 Vehicle Network Technologies and Cybersecurity

The software-intensive automotive ECUs control two or more of the electrical systems or subsystems in an automotive vehicle using a bus system for communication. The main forces driving the development of vehicle network technologies have been the advances made in ECU components, governmental regulations imposed, and consumer requests. Among the best known and most common are the CAN, the local interconnect network (LIN) designed for controlling the vehicle ECUs, and the media-oriented systems transport (MOST) designed for all kinds of vehicle multimedia applications, such as audio, video, navigation, and communication systems. Thus, developing new vehicle models increases the number of microcontrollers used which results in an ever-increasing number of nodes within the vehicle network and in turn increase the vulnerability. This makes, cybersecurity of vehicle network technologies is an important factor in the prevention of cyberattacks in vehicles

because ECUs in a vehicle typically receive their input from sensors that send data which is used for computation. Various actors are used to enforce the action determined by the ECUs. The ECUs need to exchange data among themselves during normal operation of the vehicle. For example, the ECM will inform the TMCU of the engine speed; and the TMCU will inform other ECUs when a gear shift occurs. This exchange of data needs to be done quickly, reliably, and securely over the vehicle network. Thus, attacks by adversaries can use the CAN bus to disrupt vehicle control systems in several areas, such as (Kao and Marculescu 2006):

- *Airbag Control System*: Adversaries emulate the behavior of a fully functional airbag control system, including a successful startup check. This code could be included in the network if the airbag system was broken or had been removed or has been electronically deactivated by the attacker.
- *Central Gateway*: Adversaries attack a gateway ECU by implementing basic filtering functions with regard to the internal vehicle communication, forcing a degree of separation between internal and external networks. An implementation flaw of the gateway ECU could be identified and exploited inducing the gateway ECU to pass on arbitrary internal CAN messages to the outside.
- *Warning Light*: Under regular operation, a light flashes in the event of unauthorized opening of a door. Adversaries attack by turning the light off and ensuring it stays off just by sending CAN commands to the comfort subnetwork.
- *Window Lift*: An adversarial attack was conducted in a simulation environment using CAN. In this test, only a few lines of malicious code were added to an arbitrary ECU in the simulated comfort CAN subnetwork. This code deploys when a predefined condition is met; in this case study, it deployed when the vehicle's speed rose over 200 km/h (≈124mph). Then, a window opened and would not close until the end of the window lift attack. Similar results were demonstrated in a corresponding physical environment.

In each of these cases, apart from the central gateway attack, adversarial attackers required physical access to the internal CAN network and the ability to insert malicious code into ECUs. In the case of the mentioned central gateway attack, the adversary required the ability to insert malicious code through the OBD interface. These attacks can be analyzed using the US Computer Emergency Readiness Team (CERT) taxonomy (Cebula and Young 2010; Cichonsky et al. 2012) and prevented using the set of short-term countermeasures suggested. These include intrusion detection and facilitating post-incident analysis through proactive forensics support. The OBD connector offers direct access to all CAN buses through a physical port within the vehicle cabin. The interface and messages are standardized which means there is a plethora of cheap, easily available scan tools for the OBD port. Scan tools available are:

- Full-featured versions with built-in software, user interfaces, etc.
- Dumb tools that must interface with another computing platform, such as a phone or a conventional personal computer (PC).

At the Black Hat Asia Security Conference 2015 in Singapore, a programmable device called CANtact was introduced which represents a physical connection between a vehicle's OBD port and a computer's USB port which runs on open-source software. A Python library makes it easy to interact with CAN networks (Akella et al. 2010). CAN frames can be easily encoded as Python objects and sent, received, logged, and inspected. CAN-based standardized diagnosis protocols are supported, such as OBD-II and Unified Diagnostic Services (UDS) ISO 14429, among others. UDS allows the reading and writing of arbitrary memory into a vehicle, making hacking of vehicles much easier as it only requires physical access to the OBD.

With regard to the CAN bus, its protocol contains no direct support for secure communications. Retrofitting the protocol with security mechanisms poses several challenges given the limited data rates available and potential for bus utilization to increase significantly. In (Lin and Sangiovanni-Vincentelli 2012), a security mechanism is described which keeps the bus utilization as low as possible. Through experimental evaluation, it has been shown that the security mechanism can achieve high security levels while keeping communication overhead, e.g., bus load and message latency, at reasonable levels. In another paper (Lin et al. 2013), an integrated mixed integer linear programming formulation was proposed to address safety and security requirements during the explanation of the mapping from the functional model to the CAN-based architecture platform. The mapping design space includes the allocation of tasks to ECUs, the packaging of signals into messages, the sharing of message authentication codes (MACs) among multiple receiving ECUs, and priority classifications of tasks and messages. The security constraints are set to prevent direct and indirect cyberattacks on the MACs. The safety constraints are defined on the end-to-end latency deadlines for safety-critical paths.

In a master's thesis (Bruton 2014), securing CAN bus communication has been investigated by analyzing software-based cryptographic methods that focus on message authentication where the challenges of dealing with a small packet frame is considerable. The scope of the thesis was to investigate the effects using cryptographic approaches for both encryption to provide message content confidentially, and authentication, to improve security in CAN bus communications without incurring unacceptable delays in communications and without the need for additional hardware resources. With regard to hard real-time constraints of the CAN bus, symmetric encryption techniques are chosen, such as AES which are based on a design principle known as a substitution-permutation network, where a combination of both substitution and permutation is applied to the message which is fast in both software and hardware. Authentication can be achieved on the CAN bus using hash functions for message authentication codes, assuming the hash function employed therein is fast.

In general, security for networked ECUs is an important issue for maintaining the integrity and privacy of data, while also improving network resiliency to cyber physical attacks, which is mostly based on security threats, such as manipulating the system at the information system level and within its surroundings, and others. To ensure security for vehicle CPSs for these types of security threats, several security objectives need to be achieved, as shown in Table 6.9.

Table 6.9 Security objectives and their impacts

Security Objective	Impact
Authenticity	Important proof for securing distributed CPSs and preventing users and devices from impersonating another system or component. Ensures that data, transactions, and communications of a CPS are genuine. Requires that the CPSs can validate that they are who they claim to be and thus avoid intrusion by the means of cyberattacks. This prevents unauthorized access to the sensor nodes or communication network while imposing and enforcing proper restrictions on what authenticated systems and components are permitted to do
Availability	Refers to the ability of always being accessible and usable while a lack of accessibility may cause a denial of service which may result in irreparable damages or malfunction of the system or components around it
Confidentiality	Refers to the capacity of a CPS to prevent the disclosure of information to unauthorized individuals or systems as part of a cyberattack. A CPS must prevent cyberattacks from interfering with the state of the CPS by eavesdropping on the communication channels between the sensor nodes and the controller, as well as between the controller and the actuator nodes
Integrity	Refers to data or resources that cannot be modified without authorization. Integrity is violated if a cyberattacker accidently or with malicious intent modifies or deletes important data such that the receiving CPS or actuator node receives false data and follows this data believing it to be true
Reliability	Fundamental requirement of a CPS, i.e., a system featuring a tight combination of, and coordination between, the CPS's computational and physical elements. A CPS is designed to process large amounts of data, employ software as a system component, run online continuously, and retain an operator-in-the-loop (OITL) because of human judgement and accountability requirements for safety-critical systems. Based on data-centric runtime monitoring, reliability of a CPS can be automatically evaluated with regard to data detection of abnormal input and output through data quality analysis. As a result, alerts can be sent to the OITL, who can then take actions and make changes to the system based on these alerts in order to achieve minimal system downtime and higher system reliability (Falliere et al. 2011)
Robustness	System property describing the degree to which a system operates correctly in the presence of a disturbance, such as unforeseen or erroneous inputs (Eisenhauer et al. 2006). The notion of robustness was inspired by notions of input-output stability as developed in control theory (Lamport 2005). Moreover, it has been shown that the proposed notion of robustness has to meet two intuitive goals: (1) bounded disturbances lead to bounded deviations from nominal behavior, and (2) the effect of a sporadic disturbance disappears in many finite steps. The proposed notion of robustness for a CPS can be verified in pseudo-polynomial time. The synthesis problem, consisting of designing a controller to enforce robustness, can also be solved in pseudo-polynomial time (Lamport 2005)
Trustworthiness	Estimating the feasible impact of a cyberattack requires evaluation of the system's dependency on its cyber infrastructure and its ability to tolerate potential failure. Further exploration of the cyber-physical relationships within the system and specific or possible attack vectors is necessary to determine the adequacy of cybersecurity efforts (Tang and McMillin 2008)

6.2.2 Cyberattack Taxonomy

Cyberattacks are more difficult to detect and prevent in ECUs and cyber-physical systems compared to cyberattacks on the Internet (Yuzhe et al. 2013). To evade detection, cyberattacks may apply multiple stages to gain access to a vehicle mission-critical system. Moreover, cyberattacks over the years have become both increasingly numerous and sophisticated. This calls for their analysis and categorization, assistance in combating new cyberattacks, and improvement of computer and network security, which necessitates cyberattack taxonomy.

The term taxonomy, in general, is derived from the Greek *taxis*, meaning arrangement or division, and *nomos*, meaning law. In this regard, it is the science of classification according to a predetermined system, with the resulting catalog used to provide a conceptual framework for discussion, analysis, or information retrieval. In theory, the development of a good taxonomy takes into account the importance of separating elements of a group (taxon) into subgroups (taxa) that are mutually exclusive, unambiguous, and, taken together, include all possibilities. Furthermore, as a good practice, the taxonomy should be simple, easy to remember, and easy to use, as mentioned in (URL7 2016). As reported in Kjaerland (2005), taxonomy of cyber-based intrusions can be proposed as it relates to computer crime profiling and highlighting cyberattackers and the attacked systems. Thus, cyberattacks were analyzed using facet theory, which offers a set of principles for guiding research design, has a companion set of multivariate statistical procedures to analyze data, and establishes a framework within which to construct theories (Brown 1985). The analysis included multidimensional scaling with *R*, a programming language and software environment for statistical computing and graphics, which provided functions for both classical and nonmetric multidimensional scaling, with the method of operation, target, source, and impact. Each facet contained a number of elements with an exhaustive description. Hence, taxonomy is proposed to consist of at least four dimensions, providing a holistic taxonomy to deal with the inherent problems in computer and network cyberattacks, as shown in Table 6.10. Within each dimension, various levels of information are provided showing the characteristics and consequences of cyberattacks.

From Table 6.10, it can be deduced that taxonomy should fulfill the following requirements, listed in Table 6.11 (Hansman and Hunt 2005).

As mentioned in Hansman and Hunt (2005), work is needed to improve the classification of blended attacks, which is a limitation within their taxonomy. Another limitation is the lack of vulnerability information which prohibits capturing information to aid in protecting a system from attacks.

An attack-centric taxonomy called Validation Exposure Randomness Deallocation Improper Conditions Taxonomy (VERDICT) has been proposed in (Lough 2001), which focuses on four major causes of security errors:

Table 6.10 Classification, characteristics, and consequences of cyber criminal attacks

Dimension	Cyberattack description
1st	Classifies the cyberattack into a cyberattack class based on the attack vector and the main behavior of the cyberattack. If there is no attack vector, the cyber criminal attack is classified into the closest category
2nd	Classifies the cyberattack targets. Targets can be classified down to very specific targets or a class of targets
3rd	Covers vulnerabilities and exploits, if they exist, used by the cyberattack. They do not have a structured classification due to the infinite number of possible vulnerabilities and exploits
4th	Takes into account the possibility for a cyberattack to have a payload or effect beyond itself. In many cases, a cyberattack will clearly be a certain kind of cyberattack; but yet it will have a payload or cause an effect that is different

Table 6.11 Requirements to develop a pragmatic taxonomy

Requirement	Aims to take into account
Accepted	Taxonomy is structured such that it can become generally approved
Comprehensible	Taxonomy is understood by those who are in the security field, as well as those who only have an interest in it
Completeness	Taxonomy is complete/exhaustive. It should account for all possible attacks and provide categories accordingly While it is hard to prove a taxonomy is complete or exhaustive, it can be justified through the successful categorization of actual attacks
Determinism	Classification procedure is clearly defined
Mutually Exclusive	Taxonomy categorizes each attack into, at most, one category
Repeatable	Classifications are repeatable
Terminology	Complies with established security terminology
Terms	Should be well defined. There should be no confusion as to what a term means
Unambiguous	Each category of the taxonomy must be clearly defined such that there is no ambiguity with respect to an attack's classification
Useful	Taxonomy is used in the security industry and by incident response teams, in particular

- *Improper Deallocation*: Improper destruction of information, or residuals of data, which also includes dumpster diving.
- *Improper Exposure*: Involves improper exposure of information that could be used directly or indirectly for the exploitation of vulnerability.
- *Improper Randomness*: Deals with the fundamentals of cryptography and the improper usage of randomness.
- *Improper Validation*: Refers to improperly validating unconstrained data, which also includes physical security.

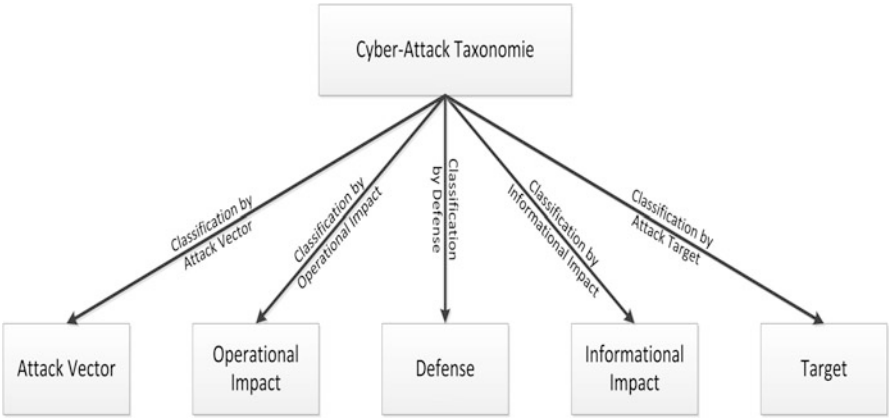

Fig. 6.19 Structure of the cyber attack taxonomy AVOIDIT

In Hansman and Hunt (2005), it is mentioned that the taxonomy described in Lough (2001) lacks pertinent information that would be beneficial for knowledge bodies, such as a CERT, to classify day-to-day attacks and ensure advisories, a taxonomy that can be used as a tool to assist in the identification of all applicable operational cybersecurity risks. Furthermore, the taxonomy described in Lough (2001) lacks classification based on the type of attack, such as Trojan, virus, worm, and others.

The cyberattack taxonomy Attack Vector, Operational Impact, Defense, Information Impact, and Target (AVOIDIT), introduced in (Guttmann and Roback 1995), provides, through application, a knowledge repository used by a defender to classify vulnerabilities that a cyberattacker can use, as shown in Fig. 6.19. AVOIDIT provides details on each cyberattack classification and how a variety of cyberattacks are represented in each category.

The general scheme shown in Fig. 6.19 is expanded in Fig. 6.20, which provides details on each attack classification and how a variety of attacks are represented in each category (Simmons et al. 2014). AVOIDIT could be extended to include new categories within each classification and will provide a cyber criminal attack defender with the appropriate information to make a clear decision in defending against cyberattacks. Advanced approaches to defending against attacks will become available and provide an extensible taxonomy for capturing new defenses. In future work, building a game-theoretic defense strategy, the applicability of AVOIDIT in determining the action space of a cyberattacker will be investigated (Shiva et al. 2010).

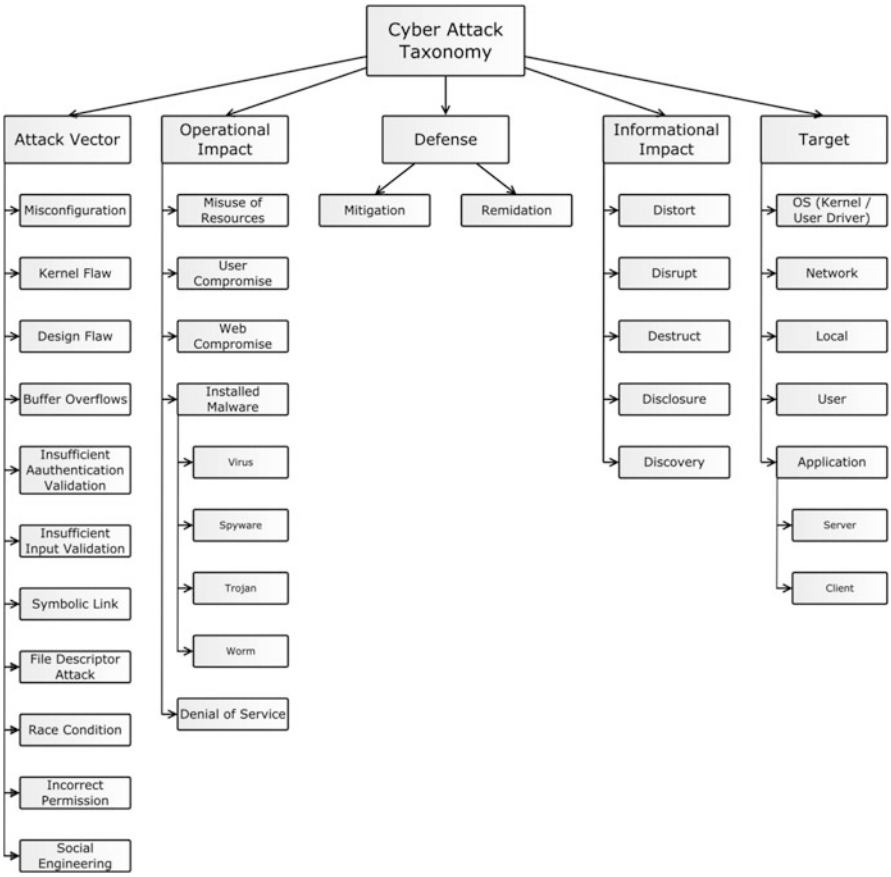

Fig. 6.20 Architecture of the AVOIDIT cyber attack taxonomy

6.3 Hacking and Automotive Attack Surfaces and Vulnerabilities

6.3.1 Hacking

Hacking is a very real vehicle security risk as evidenced by the increasing number of cyberattacks on systems and data. Hacking means that adversaries target trusted security controls as a means of facilitating later cyberattacks whereby hacking adversaries (hackers) may threaten information security on multiple levels simultaneously. Hackers may attack (Shimeall and Spring 2014):

- *Data* used in essential business processes, including compromise, imitation, or redirection of data sources, using websites that closely imitate institutions to obtain authentication information used in later frauds.
- *Individual hosts*, exploiting weaknesses in the operating system or in the application software
- *Users*, either as malicious insiders or as malicious outsiders
- *Networks*, via remote access methods or by exploiting the trust within networks to propagate from an initial intrusion point of compromise

Hackers employ the following strategies:

- *Direct Physical Access*: In this, the simplest hacking attack strategy, the hacker strikes against the target from an intrusion point, without intermediate or third-party hosts involved except for normal traffic routing. This strategy is applied in cyberattacks where the intrusion point is of little value to the hacker or the probability of backtracking is very low.
- *Progressive Access*: The hacking adversary uses a series of intermediate hosts between the intrusion point and the target, each of which is compromised using the same set of exploits.
- *Mass Hacking*: The hacker compromises a group of third-party hosts and uses all of them at once against the targeted host.
- *Misdirection Access*: Generates traffic to confuse or distract the defenders in dealing with their direct cyberattack (Shimeall and Spring 2014).

6.3.2 Automotive Attack Surfaces and Vulnerabilities

As described in Sect. 6.4.2, the number of electronic components in modern vehicles has increased rapidly and continuously during recent years. This has resulted in millions of lines of code executing on several heterogeneous embedded computers with huge connectivity provided by automotive bus systems such as CAN. On one hand, many sensors and actuators have been developed and embedded in vehicles to make passengers feel safer. On the other hand, more advanced entertainment and navigation systems have made their way into vehicles to make traveling more comfortable. Although this technological progress has generated significant benefits in terms of efficiency and cost, it has also created more opportunities for new attack surfaces which increase the vulnerability of vehicles to cyberattacks.

With regard to the Bluetooth® network protocol used in vehicles, an overview of the security architecture and security modes of the Bluetooth® protocol, as well as the vulnerabilities that Bluetooth® networks face, is reported in (Johnson 2010). Three of the most crucial categories correspond to the confidentiality, integrity, and availability (CIA) triad, a model designed to guide policies for information security with regard to threats of:

- Denial of service
- Disclosure of unauthorized information
- Integrity of information

The triad is sometimes referred to as the availability, integrity, and confidentiality (AIC) triad to avoid confusion with the initialism for the US Central Intelligence Agency (CIA).

In the context of the CIA triad:

- *Confidentiality* is a set of rules that limits access to information and is roughly equivalent to privacy.
- *Integrity* is the assurance that the information is trustworthy and accurate.
- *Availability* is a guarantee of reliable access to the information by authorized people.

Availability is best ensured by rigorously maintaining all hardware, performing hardware repairs immediately when needed, and maintaining a correctly functioning operating system environment that is free of software conflicts. However, the powerful directional antennas in Bluetooth-based networks can be used to considerably increase the scanning, eavesdropping, and attack range of almost any kind of Bluetooth® attack.

In Cárdenas et al. (2011), Bluetooth® is considered to be one of the biggest and most viable cyber attack surfaces on modern vehicles, due to the complexity of its protocol and underlying data. Additionally, Bluetooth® has become ubiquitous within the vehicle domain, giving cyberattackers a very reliable intrusion point to test attack scenarios.

In Cárdenas et al. (2008), Bluetooth® capabilities built into test vehicles' telematics units have been investigated. Access to the telematics ECU's UNIX®-like operating system was gained through reverse engineering, and the particular program responsible for handling Bluetooth® functionality was identified. It was verified that the ECU's operating system contained a copy of a popular embedded implementation of the Bluetooth® protocol stack along with a sample hands-free application and a custom-built interface. The interface contained vulnerability that allowed buffer overflow attacks to be mounted by any paired Bluetooth® device and allowed arbitrary code to be executed on the telematic unit. Adversaries use buffer overflows to corrupt the execution stack. By sending carefully crafted input to an application, a cyberattacker can cause the application to execute arbitrary code, possibly taking over the mission-critical cyber-physical system's functionality. Buffer overflow attacks generally rely on two techniques, usually in combination:

- Having the operating system mishandle data types
- Writing data to particular memory addresses

This means that strongly typed programming languages and environments that disallow direct memory access usually prevent buffer overflows from happening. Available techniques to prevent buffer overflows include:

- Code auditing
- Compiler tools, such as StackShield, StackGuard, and Libsafe, etc.
- Nonexecutable stacks which are supported by many operating systems
- Patches with regard to bug reports relating to applications upon which the code is dependent

The US federal government mandated the OBD-II port, under the dashboard, which provides a direct and standard hard-wired communication link to ECUs through which access is allowed to read and reset a vehicle's fault codes. Also, access to information from various units through the diagnostic connector is possible so that all systems can be diagnosed and programmed. User-upgradable subsystems, such as audio players, are attached to these same networks by a variety of short-range wireless devices, such as wireless tire pressure monitoring system (TPMS), as well as Bluetooth® devices and more, which also represent new partial attack surfaces. However, vehicles equipped with driving aid systems, such as an electronic stability program (ESP) or adaptive cruise control (ACC), allow deep interventions in the driving behavior of the vehicle, too. Furthermore, electronic drive-by-wire vehicle control systems fully depend on the underlying automotive data networks. Moreover, vehicle communication networks assure safety against technical interference; but they are mostly unprotected against malicious cyberattacks. This increasing coupling of unsecured automotive components together with new multimedia networks, such as MOST, and the integration of wireless interfaces, such as GSM or Bluetooth, causes various additional security risks in the context of attack surface intrusion points. Summing up, it can be stated that today's vehicles are pervasively computerized with regard to their increasingly sophisticated services and embedded communication features and, hence, potentially much more vulnerable to cyberattacks than in the past. As a result, the attack surface intrusion points must be multifaceted, as shown in Fig. 6.21, including safety-critical components such as brakes, engine, transmission, and others.

Cyberattacks against vehicle safety-critical systems result in physical control of the various components of the vehicle and access to the internal vehicle network. This allows the adversary to inject code into the vehicle networks to directly or indirectly control the desired ECUs. Researchers from the University of California San Diego and the University of Washington were able to execute code remotely on a telematics unit of a vehicle by exploiting vulnerability in the Bluetooth stack on an ECU and by separately compromising a cellular modem.

Vehicle manufacturers also provide some kind of external digital multimedia port, typically a USB port or an iPod/iPhone docking port, allowing users to control their vehicles' media systems using their personal audio players or phones. Consequently, an adversary might deliver malicious code by encoding it onto a CD or a song file along with using social engineering to convince the user to play

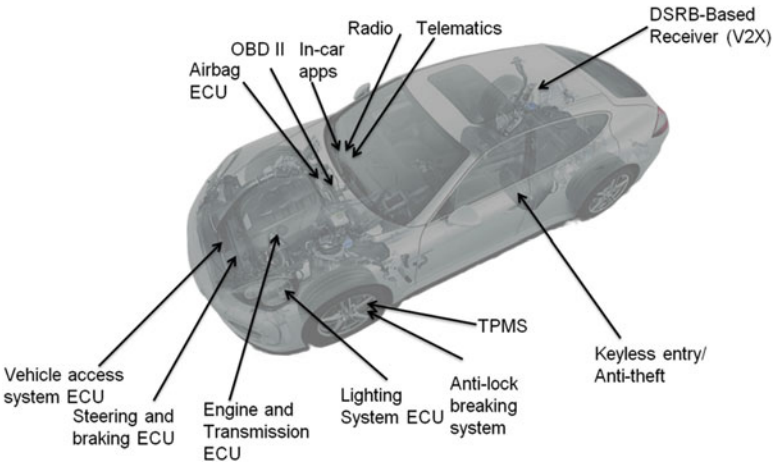

Fig. 6.21 Anatomy of the 15 most hackable and exposed attack surface intrusion points on a next-generation vehicle, modified after (Intel Security 2015)

it. Alternatively, a user's phone or iPod might be compromised out of band and malicious software installed onto it that attacks the vehicle's media system when connected (Checkoway et al. 2011).

As reported in Valasek and Miller (2014), a compromised ECU cannot control the safety features of a vehicle. This ECU's task is typically only related to receiving and processing radio signals. Therefore, a cyber physical attack usually requires a second step which involves injecting malicious code into the internal vehicle network in an attempt to communicate with safety-critical ECUs, such as those responsible for steering, braking, and acceleration. In some vehicles, this may be trivial; but in many designs, the ECU which was compromised remotely will not be able to directly send messages to these safety-critical ECUs. In this case, the cyberattacker will have to somehow get messages bridged from the network of the compromised ECUs to the network where the target ECU resides. This might require tricking the gateway ECU or compromising it outright.

The researchers from the University of California San Diego and the University of Washington (Checkoway et al. 2011) demonstrated a way to compromise the bridge ECU in their vehicle to get from the less privileged CAN network to the one containing the ECU in charge of braking. After the attacker has wirelessly compromised an ECU and acquired the ability to send malicious code to a desired target ECU, the attacker may communicate with safety-critical ECUs, making them behave in some way that compromises vehicle safety. This involves reverse engineering the messages on the network and figuring out the exact format to perform some physical action.

Since each manufacturer, and perhaps each model and even each year, use different data in the messages on the bus, the message reverse engineering process requires a large amount of work and is manufacturer specific. For example, the

messages to lock the brakes on one manufacturer's vehicle likely won't work on a vehicle from a different manufacturer. Furthermore, some ECUs only listen to certain messages and may have safety features built into them, such as not responding to certain messages while the vehicle is in motion (Valasek and Miller 2014). Thus, it is important to know, without a detailed investigation, whether it is possible to affect cyber physical vehicle features though malicious software injection since it essentially relies on the implementation of the ECUs. Therefore, Valasek and Miller (2014) report an approach similar to measuring the remote attack surface. For each vehicle, they list the computer-controlled features. In the Toyota Prius, for example, the collision prevention system is designed to stop the vehicle when certain CAN messages are received. This is a safety feature and can be exploited. So while all vehicles may or may not be vulnerable to safety-critical actions through CAN data injection (Valasek and Miller 2014), it can be assumed that those with advanced computer-controlled features are more susceptible since they are designed to take physical actions based on data received on the internal network.

In the case of telematics services, value-added automatic features, such as those listed below, are provided over a long-range wireless link.

- Crash response
- Remote diagnostics
- Stolen vehicle recovery

For this purpose, telematic systems integrate internal automotive subsystems with a remote command center via a wide-area cellular network connection.

Some service providers have taken this concept even further by proposing a car-as-a-platform (CaaP) model for third-party development and applications related to in-car connected platforms, offering a selection of features in connected vehicles (cars) with a special focus on entertainment apps and safety-management features. Entertainment is one of the most popular features available for the connected car. Entertainment features include integrations with apps, such as Pandora®, Yelp®, Facebook®, and others. Hughes Telematics has described plans for developing an app store for automotive applications (Mollmann 2009), while Ford recently announced that it will open its SYNC® telematics system as a platform for third-party applications (Goodwin 2009). SYNC®, an automaker-installed integrated in-vehicle communications and entertainment system, allows users to:

- Control music
- Make hands-free telephone calls
- Perform other functions with the use of voice commands

The system consists of applications and user interfaces developed by Ford and other third-party developers. With regard to a cellular modem built in to the vehicle, Ford is planning to execute OTA software updates just as Tesla already does. Ford

can already do OTA updates to SYNC3 using Wi-Fi when the vehicle is connected at home. Other telematics systems, such as General Motors' OnStar®, provide value-added features, such as:

- Automatic crash response
- Remote diagnostics
- Stolen vehicle recovery over a long-range wireless link

To do so, these telematics systems integrate internal automotive subsystems with a remote command center via a wide-area cellular connection.

Furthermore, there are many proposed V2V and V2X communications systems (CAMP09 2008; CAMP10 2008; CAMP05 2005; VTTI 2007) that will broaden the attack surface intrusion points further. More possible attack surfaces of connected vehicles are given in Fig. 6.21. Overall, these trends suggest that a wide range of attack vectors will be available by which an adversary might compromise a vehicle's electric/digital components and gain access to internal vehicular networks with unknown consequences. The two kinds of attack vectors by which adversaries might gain access to a vehicle's internal networks are, as previously mentioned, the physical access and the numerous wireless interfaces embedded in today's vehicles. These interfaces accept outside input through which it is possible to remotely compromise key ECUs via externally facing vulnerabilities, remotely control a vehicle over the Internet, and others. With regard to physical access, an adversary can, with even momentary access to the vehicle, insert a malicious component into a vehicle's internal network via the ubiquitous OBD-II port (Kosher et al. 2010).

The next step proposed by some service providers is a connected-car-as-a-digital-platform (CCaaDP) model. The vehicle itself is a connected platform that enables multiple protocols to communicate with each other and connects to the cloud through the user's mobile cellular service and hardware. The current models feature multiple communication systems that connect the following to the drivers display.

- Airbag
- Camera/radar systems
- Driver assist
- Engine
- Safety systems
- Tire pressure

There is also a network that connects the passenger area to the information system along with entertainment control, which will open an additional attack surface. These systems are moving from just wired systems, such as CAN, MOST, and Flexray™, to more standard systems such as Ethernet, a new wired solution which can support low weight, unshielded cable capable of 100 Mbps in full duplex mode for connectivity. This wired system is being sought by many automakers to allow the vehicle to be a platform core, such as a data center, with one in-platform interconnect

system and the edge of the network to be wireless or a USB interface (Chatterjee 2012). This single network configuration simplifies the communication options by creating a single protocol for the data transfer. This allows for industry qualification, such as the standards in the global automotive industry, including TS16949 compliance/ISO 9001 certification, in-car EMC performance, and AEC-Q100, to be addressed in one pass, thus offering the automotive One-Pair Ethernet Alliance Special Interest Group (OPENSIG). OPENSIG is promoting the switch to in-car Ethernet.

Hence, the connected vehicle (car) is driving the automotive semiconductor market. Factory-installed networking connections are increasing due to integration of systems with sensor networks that are not accessible post vehicle assembly. The automaker-installed rate may be as high as 60% in the near future. Costs of these systems have been dramatically reduced, and they are available both in mass market vehicles and luxury applications, which also opens new attack surface intrusion points, resulting in increased vulnerability.

Therefore, the following questions need to be answered as they relate to security in vehicle CPSs:

- Which methods and tools can be used for security testing and evaluation in the automotive industry?
  - Many methods and tools are available for vehicle security testing and evaluation; however, these methods alone are not able to address all security problems that might arise from the implementation phase. Hence, an evaluation methodology is needed, to determine which method(s) could be used for security testing and which could address the security problems arising from the implementation phase in vehicle software (Chalkias et al. 2009).
- How can various methods be combined to systematically perform security testing?
  - Different methods are required for security testing followed by ad hoc approaches. A single method may always lack a foolproof strategy for eliminating all kinds of security problems. The solution is, therefore, to combine the advantages of other methods into the presence of one. Therefore, a systematic approach to combining various methods can be beneficial. By systematic security testing, it may also be possible to prove that a potential vulnerability can be exploited (Chalkias et al. 2009).

Security testing methods available in the automotive domain which prevent intrusion points for cyberattacks are:

- *Functional Security Testing*: Investigates functional correctness and the robustness testing of security functionalities (Chalkias et al. 2009). For example, cryptographic algorithms to be implemented should be checked for their correctness. Implementations of cryptographic algorithms are often tested with official test vectors. Developers mostly rely on specifications and official test vectors during code development. Some doors of opportunity remain for cyberattackers

to exploit potential vulnerabilities that may arise from other sorts of random test vectors. These kinds of security vulnerabilities are missed by functional security testing teams. Adhering to MISRA C/MISRA C++ safety coding standards can reduce the number of such potential vulnerabilities in software. MISRA guidelines have been widely adopted to ensure the quality of safety- and security-critical software in automotive, aerospace, defense, industrial, medical, and rail applications. By following MISRA rules, developers can be assured of using the most stringent software coding guidelines to mitigate liability and risk in software applications on which human lives depend, and to avoid coding practices that can introduce security vulnerabilities (URL8 2016).

- *Fuzzing and Penetration Testing*: Vehicle CPSs with malformed inputs that might be able to uncover unsafe weaknesses and vulnerabilities (Xiao et al. 2008) with regard to possible attacks through external ports and physical devices can be tested. The available code for automotive software is not open source. Reverse engineering is currently used to retrieve binary code, and all types of security tests are performed with the help of third-party debuggers, for example, OllyDbg, a 32-bit assembler level analyzing debugger for Microsoft Windows, and IDA Pro, a Windows, Linux®, or Mac OS® X-hosted multiprocessor disassembler and debugger, and others. Penetration testing investigates possible attacks through external ports and physical devices and is a sophisticated way of testing the whole system by a security tester with his/her knowledge of security testing (Chalkias et al. 2009). It involves testing hardware and software with single a or a combination of various security testing methods such as:
  - Code review
  - Manual inspection
  - Static analysis
- *Vulnerability Scanning*: A test system with a known set of vulnerabilities that could be either unsafe functions or unsafe configurations (Chalkias et al. 2009). The Open Vulnerability Assessment System (OpenVAS) is a framework of several services and tools offering a comprehensive and powerful vulnerability scanning and vulnerability management solution that scans for open ports in automotive IT and software. OpenVAS products are free software. Most components are licensed under the GNU General Public License (GNU GPL). The architecture of OpenVAS is shown in Fig. 6.22 (URL9 2016).

The most essential blocks shown in Fig. 6.22 have the following meanings:

- *OpenVAS Command Line Tool*: Contains the command line tool, Open VAS management protocol, which allows the creation of batch processes to drive the OpenVAS manager. It runs on Windows, Linux, etc. and is a plugin for Nagios®. Nagios can be considered to be an industry standard for monitoring IT infrastructures (www.nagios.org).
- *Greenbone Security Assistant (GSA)*: Is a client for OpenVAS management protocol and OpenVAS administration protocol and serves HTTP and HTTPS.

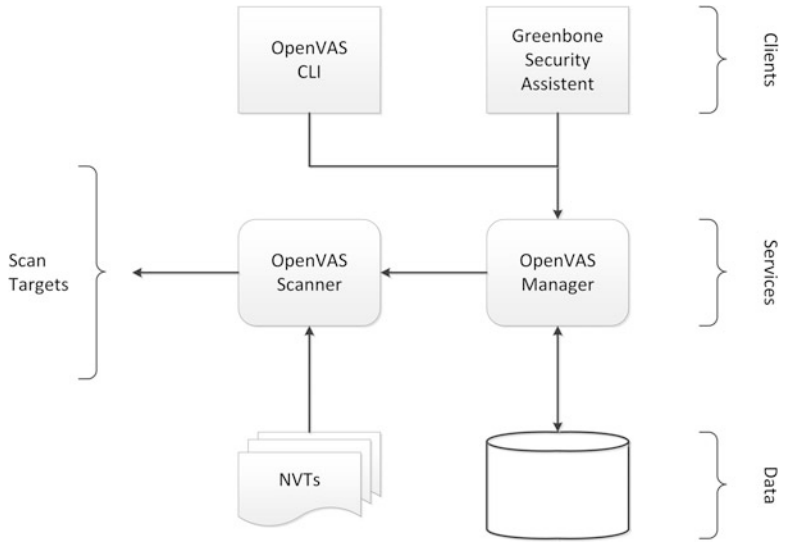

Fig. 6.22 Open Vulnerability Assessment System (OpenVAS) framework, modified after (URL9 2016)

- *OpenVAS Scanner*: Uses the OpenVAS transfer protocol on the server side and the OpenVAS manager on the client side.
- *Network Vulnerability Tests (NVTs)*: Work on the detection of certain product vulnerability evaluations. The actual detection NVTs should result in a Common Platform Enumeration Code (CPEC) code for the product.

Finally, in Fig. 6.23, risks of security attacks are summarized with regard to attack vectors which represent the path or means by which a hacker can gain access to CPSs and communication networks to intrude a malicious outcome. The attacker's goal is to exploit system or component vulnerabilities, including immediate and long-term risks. In general, it can be stated that recent hacking attacks will enable a steep rise in the demand for automotive security solutions that repel malware intrusion. To some extent, firewalls and antivirus software can block attack vectors; but no intrusion prevention method today is totally attack proof given that a defense method that is effective today may not remain so for long. Hackers are constantly updating attack vectors and seeking new ones by looking to gain unauthorized access to vehicle CPSs and communication networks.

Immediate and long-term risk as a function of attack vectors, attack goals, and the vulnerable system are shown in comparison in Fig. 6.23 is:

- *Man-in-the-Middle Attack (MITA)*: Involves an attacker positioning himself between the two nodes A and B which will communicate without the knowledge of each other. Hence, the man-in-the-middle (node C) makes node A believing that he is node B. Thereafter, he makes node B believing that he is node A. In this way node C handles all communication between nodes A and B without revealing this fact, and he can copy, alter, or compromise any messages sent.

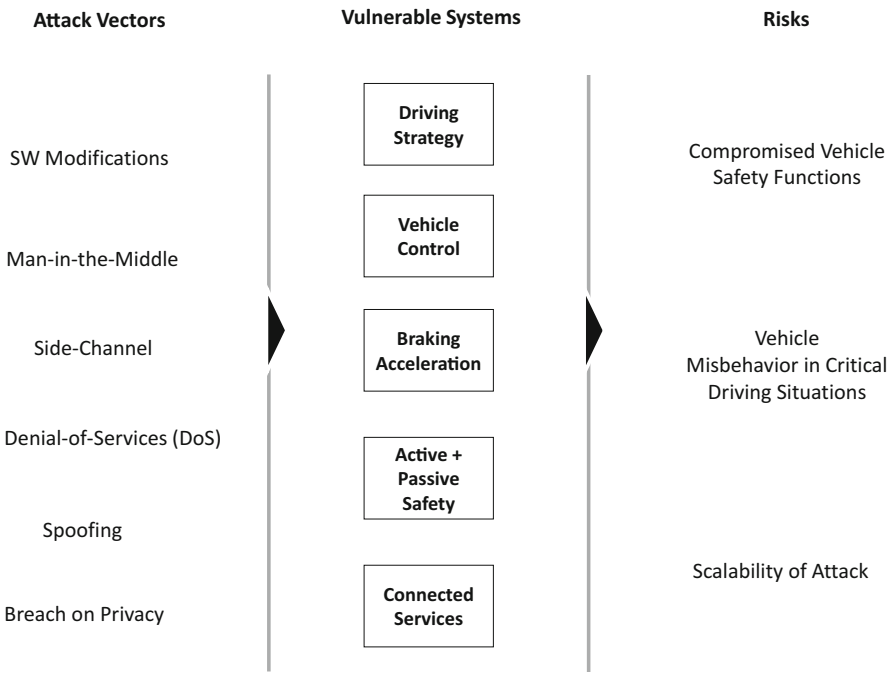

Fig. 6.23 Risks of cybersecurity w.r.t. cyberattacks in vehicles

- *Side Channel Attack (SCA)*: Attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the algorithms. Some side-channel attacks require technical knowledge of the internal operation of the system on which the cryptography is implemented, although others such as differential power analysis (DPA) are effective as black box attacks.
- *Brute Force Attack*: Refers to attempts to obtain logon credentials by guessing usernames and passwords. Some risks exist for services that allow remote access, brute force attackers use password guessing tools and scripts containing default password databases, dictionaries, or rainbow tables that contain commonly used passwords and may try all combinations of a character set. Brute force attacks are typically one-by-one attacks executed by an expert attacker against selected targets (Johnson 2016).
- *Denial-of-Service (DOS) Attack*: Type of attack where the attackers attempt to prevent legitimate users from accessing the service. In a DoS attack, the attacker usually sends excessive messages asking the network or server to authenticate requests that have invalid return addresses. The network or server will not be able to find the return address of the attacker when sending the authentication approval, causing the server to wait before closing the connection. When server closes the connection, the attacker sends more authentication messages with invalid return addresses. Hence, the process of authentication and server wait will begin again, keeping the network or server busy (URL1 2017).

- *Spoofing*: Situation where a cyber attacker (or his program) successfully masquerades as another by falsifying data and hence gaining an illegitimate advantage.
- *Compromised Privacy*: A term used to classified matter, knowledge of which has, in whole or in part, passed to an unauthorized person or persons or which has been subject to risk of such passing.

Cyber attacks have changed. Broad, scattershot attacks designed for mischief have been replaced with advanced persistent threats focused on acquiring valuable data. Modern cyberattacks are often conducted across multiple vectors and stages. They have a plan to get in, signal back from the compromised network, and extract valuable data despite network security measures. Traditional defense-in-depth security measures, such as next-generation firewalls, antivirus, web gateways, and even newer sandbox technologies only look for the first move – the inbound attack. Advanced cyber attacks are designed to evade traditional network security.

6.4 Intrusion Detection and Prevention

6.4.1 Intrusion Detection

Intrusion detection can be defined (Heady et al. 1990) as any set of actions that attempts to compromise the CIA of a resource (see Sect. 6.3.2). Thus, it is a violation of the security constraints of the respective system. But as reported by Kumar and Spafford (1994), any definition of an intrusion is imprecise as security policy requirements do not always translate into a well-defined set of actions because intrusion detection is a methodology by which intrusions are detected. This methodology can be divided into two categories:

- *Anomaly Intrusion Detection*: System activities are observed which periodically generate profiles that capture their behavior, and older data is updated regularly to indicate its anomaly. As input audit records are processed, the observed system periodically generates a value indicative of its abnormality which may happen in a case where there is too much deviation from the regular profiles; and the intrusion detection system reports an intrusion. However, this can lead to false-positive alarms, depending on the conditioning or sensitivity of the intrusion detection system. False positives are events that are reported as malicious but in reality they are not.
  - *Advantage of Anomaly Intrusion Detection*: No predefined rules for detection of intrusions are required; hence new attacks can be detected.
  - *Disadvantages of Anomaly Intrusion Detection*: False positives can arise, leading to inconvenience for the users. Establishment of regular profile usage is required but is often hard to achieve.
- *Misuse Intrusion Detection*: Based on well-defined patterns of input events, assuming that the state transition of the system leads to an intruded state when exercised with the intrusion pattern, weaknesses in the system and application software can be exploited. The objective is to frame the intrusion detection

problem as a pattern-matching problem and to develop efficient algorithms for such matching. But simply specifying an intrusion pattern without the initial state specification is often insufficient to capture an intrusion scenario fully (Shieh and Gligor 1991).

Another classification scheme is based on the intrusion types presented in Denning (1987) and Smaha (1988) and is shown in Table 6.12, which introduces intrusion types, their characteristics, and detection possibilities.

Let $A_1, A_2, \ldots, A_n$ be n measures used to determine if an intrusion is occurring on a system at any given moment, whereby each A_i measures a different aspect of the system with

$$A_i = \begin{cases} 1 & \text{implying that the measure is anomalus} \\ 0 & \text{otherwise} \end{cases}$$

Table 6.12 Intrusion types and their detection

Intrusion type	Characteristics	Detection
Attempted break-in	Breaking into a system might generate an abnormally high rate of password failures with regard to a single account or the system as a whole	Atypical behavior profiles or violations of security constraints
Denial of service	An intruder able to monopolize a resource might have abnormally high activity with regard to the resource, while activity for all other users is abnormally low	Atypical use of system resources (e.g., networks)
Inference by legitimate user	A user attempting to obtain unauthorized data from a database through aggregation and inference might retrieve more records than usual	Atypical behavior profiles using I/O resources
Leakage by legitimate user	A user trying to leak sensitive documents might log into the system at unusual times or route data to remote printers not normally used	Atypical usage of I/O resources
Masquerading or successful break-in	A log into a system through an unauthorized account and password might have a different login time, location, or connection type from that of the account's legitimate user	Atypical behavior profiles or violations of security constraints
	An intruder's behavior may differ considerably from that of the legitimate, e.g., a user using most of his time browsing through directories and executing system status commands whereas the legitimate user might edit, compile, or link programs	
Trojan horse	A program is substituted for a legitimate program	Atypical CPU time or I/O activity
Virus	May cause an increase in the frequency of executable files rewritten or storage used by executable files	Atypical CPU time or I/O activity

Let H be the hypothesis that the system is currently undergoing an intrusion. The reliability and sensitivity of each anomaly measure A_i is determined by

$$p(A_i = 1|H)$$

and

$$p(A_i = 1|/H).$$

The combined belief in H is

$$p(H|A_1, A_2, \ldots, A_n) = p\big(A_1, A_2, \ldots, A_n\big|H\big) \times \frac{p(H)}{p(A_1, A_2, \ldots, A_n)}$$

which requires the joint probability distribution of the set of measures conditioned on H and /H.

In (Lunt et al. 1992), covariance matrices are used to account for the interrelationships between measures. If the measures A_1, A_2, …, A_n are represented by vector A, then the compound anomaly measure is determined by

$$A^T C^{-1} A$$

where C is the covariance matrix representing the dependence between each pair of anomaly measures A_i and A_j.

The foregoing methodology on intrusion detection is now broadened by the issue of intrusion prevention, the process of performing intrusion detection and attempting to stop the possible incidents detected. Therefore, the issue is one of introducing intrusion detection and prevention systems that are primarily focusing on identifying possible incidents, logging information about them, attempting to stop them, reporting them to security administrators, and documenting existing threats. Hence, intrusion detection and prevention have become a necessary issue to the security infrastructure of nearly every mission-critical system. The types of intrusion detection and prevention system (IDPS) techniques can be differentiated by the types of events that they monitor and the ways in which they are deployed, as shown in Table 6.13.

Securing automotive mission-critical components is a very important objective because these components are targeted by cyberattackers who want to gain access to the sensitive information of mission-critical components, system configurations, vulnerabilities, and others. Therefore, specific protective actions are of particular importance, such as encryption and other actions for transmitting data physically or logically over separate network components. This includes verifying that the components are working as desired, monitoring the components for security issues, performing regular vulnerability assessments, responding appropriately to vulnerabilities, and testing and deploying intrusion detection and prevention system updates. Resource constraints should also be taken into consideration by defining specialized sets of requirements for the following:

Table 6.13 Intrusion detection and prevention system types

IDPS Type	Characteristics
Host based	Monitoring characteristics of a single host and events occurring within that host for suspicious activity
Network based	Monitoring network traffic for particular network segments or devices and analyzing network and application protocol activity to identify suspicious activity
Network behavior analysis	Examines network traffic identifying threats that generate unusual traffic flows, such as distributed denial-of-service (DDoS) attacks, certain forms of malware, and policy violations (e.g., client system providing network services to other systems)
Wireless	Monitoring wireless network traffic and analyzing it to identify suspicious activity involving the wireless networking protocols themselves

- *Life Cycle Costs*: Initial and maintenance costs whereby the life cycle concept must be made in the context of achievement of the capability required to meet the operational conditions.
- *Management*: Design and implementation of reliability, interoperability, scalability, and product security requirements, as well as operation and maintenance, including software updates, and training, documentation, and technical support.
- *Performance*: Maximum capacity and performance features of intrusion detection and prevention.
- *Security Capabilities*: Information gathering, logging, detection, and prevention of intrusions.

6.4.2 Intrusion Prevention

Intrusion prevention technologies are differentiated from intrusion detection technologies by the characteristic that intrusion prevention system (IPS) technologies respond to a detected threat by attempting to prevent it from succeeding. Several response techniques are used for intrusion prevention, which can be divided into the following groups (Scarfone and Mell 2007):

- *IPS Stops Intrusion Attack Itself*: Examples of how this could be done are as follows:
  - Block access to target or possibly other likely targets from offending user account, IP address, or other intrusion attacker attribute.
  - Block all access to targeted system, service, application, or other resource.
  - Terminate network connection or user session that is being used for intrusion attack.
- *IPS Changes Security Environment*: IPS could change configuration of other security controls to disrupt an intrusion attack. Common examples are:
  - Cause patches to be applied to a host if IPS detects that the system has vulnerabilities.

  - Reconfigure a network device, e.g., firewall, router, switch, to block access by the intrusion attacker or to the target, and alter a system-based firewall on a target to block incoming attacks.
- *IPS Changes Intrusion Attack's Content*: Some IPS technologies can remove or replace malicious portions of an intrusion attack to make it benign.
  - A simple example is an IPS that removes an infected file attachment from an e-mail and then permits the cleaned email to reach its recipient.
  - A more complex example is an IPS that acts as a proxy and normalizes incoming requests, which means that the proxy repackages the payloads of the requests, discarding header information. This might cause certain intrusion attacks to be discarded as part of the normalization process.

With regard to potential vehicle cyber criminal intrusion attacks, WLAN technology is the most important technology for use with intrusion prevention systems. Most WLANs use the IEEE 802.11 family of WLAN standards. IEEE 802.11 WLANs have two fundamental architectural components, see Fig. 6.27:

- An ACCESS POINT that logically connects STATIONs with a distribution system, which is typically a system's wired infrastructure.
- A STATION, which is a wireless endpoint device.

Some WLANs also use wireless switches which are devices that act as intermediaries between ACCESS POINTS and the distributed systems. The purpose of a switch is to assist in managing the WLAN infrastructure. In WLANs without wireless switches, the ACCESS POINTs connect directly to the distributed systems. The IEEE 802.11 standard also defines the following two WLAN architectures:

- *Ad Hoc Mode*: Peer-to-peer mode that does not use ACCESS POINTs, involving two or more STATIONs communicating directly with one another.
- *Infrastructure Mode*: ACCESS POINTs connect wireless STATIONs to a distributed system, typically a wired network.

Each ACCESS POINT and STATION on a WLAN can be identified by its Media Access Control (MAC) address a unique 48-bit value that is assigned to a wireless network interface card.

Some of the wireless intrusion detection and prevention techniques terminate connections between ill-conditioned or misconfigured STATIONs and an authorized ACCESS POINTs or between an authorized STATION and an ill-conditioned or misconfigured ACCESS POINT. This is typically done by sending messages to the endpoints, telling them to disassociate the current session. The IPS then refuses to permit a new connection to be established. Most IPSs are able to specify the prevention capability configuration for each type of alert. This usually includes enabling or disabling prevention, as well as specifying which type of prevention

capability should be used. Others have a learning or simulation mode that suppresses all prevention actions and instead indicates when a prevention action would have been performed. This allows monitoring and fine-tuning of the configuration of the prevention capabilities before enabling prevention, which reduces the risk of performing prevention actions on benign activity.

Thus, the main task of intrusion prevention is to defend a CPS by detecting an attack and possibly repelling it. Detecting hostile attacks depends on the number and type of appropriate actions, which can be obtained from publicly available data, found in the National Vulnerability Database (NVD), the US government repository of standards based vulnerability management data, or the CVE database, a dictionary of publicly known information security vulnerabilities and exposures. Both of these databases are sponsored by the US Department of Homeland Security/US Office of Cybersecurity and Communications/Computer Emergency Readiness Team and help in understanding the severity of the current security threat landscape (see Sect. 6.1.1). Therefore, intrusion prevention requires well-selected investigations of threats because adversaries are seeking out and exploiting network, device, and application vulnerabilities to attack, causing serious problems for the vehicle attacked. Thus, intrusion detection and prevention strategies are becoming a critical issue for automakers, OEMs, and suppliers.

The main activities of an IDPS are summarized in Fig. 6.24. If a cyberattack is suspected, an alarm list of possible attacks is created, and the component or subsystem the intruder is attempting to attack is locked (Landrum et al. 2014). As can be seen in Fig. 6.24, preprocessing describes processing performed on raw data, transforming this data into a format that is more easily and effectively processed for the purpose of intrusion detection. There are a number of different tools and methods used for preprocessing. One is feature extraction, which pulls out specified data that is significant in some particular context, such as intrusion. The ruleset shown in Fig. 6.24 contains three components:

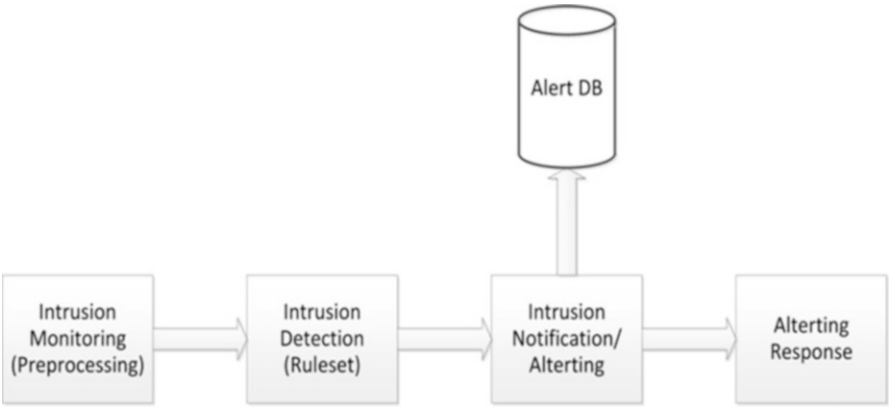

Fig. 6.24 Intrusion detection and prevention systems tasks

- Set of rules
- Database
- Interpreter for the rule

A rule can be defined as an ordered pair of symbol strings. The ruleset has a predetermined, total ordering; and the database is a collection of intrusion-related patterns. The interpreter operates by scanning the ordered pair of pattern strings of each rule until one is found that can be successfully matched against the intrusion-related pattern of the database.

If an intrusion is identified, the notification feature of the intrusion prevention system, shown in Fig. 6.24, starts an alert response as an operational routine encapsulating the identified intrusion scheme. Hence, the intrusion prevention architecture, shown in Fig. 6.24, is a key element in controlling the information flow between attack surfaces and mission-critical systems, as shown in Fig. 6.25.

In addition to the foregoing, the IDPSA architecture scheme, shown in Fig. 6.26, illustrating detecting and preventing unknown vulnerabilities is a task which expands the ruleset-based approach in Fig. 6.24 through an artificial neural network. Executing this approach require again data gathering and pre-processing which means that all incoming data is collected, transformed and normalized to standard entities. Thereafter, feature extraction from this data is required in which feature entities are objects of information that could be used like performance evaluation for number of packets transferred between vehicles, delay in transfer of packets, number of dropped packets and more. Other basic features could be the information in the header of the packets transferred which could include, for example:

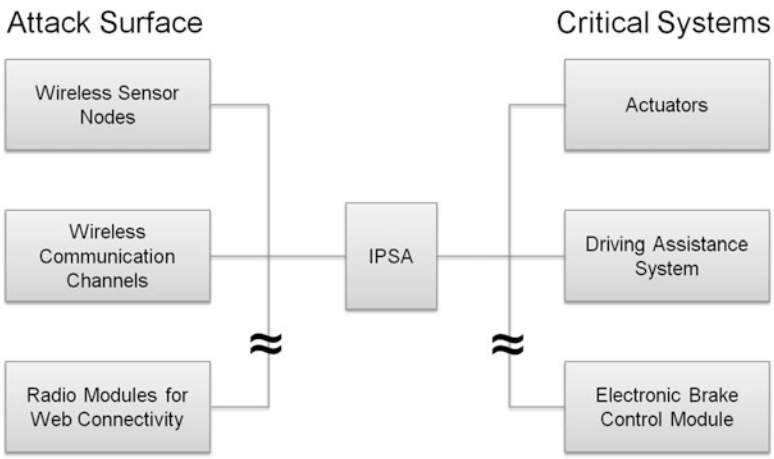

Fig. 6.25 Intrusion prevention system architecture (IPSA) lies in between attack surfaces and mission-critical systems of vehicle cyber-physical systems

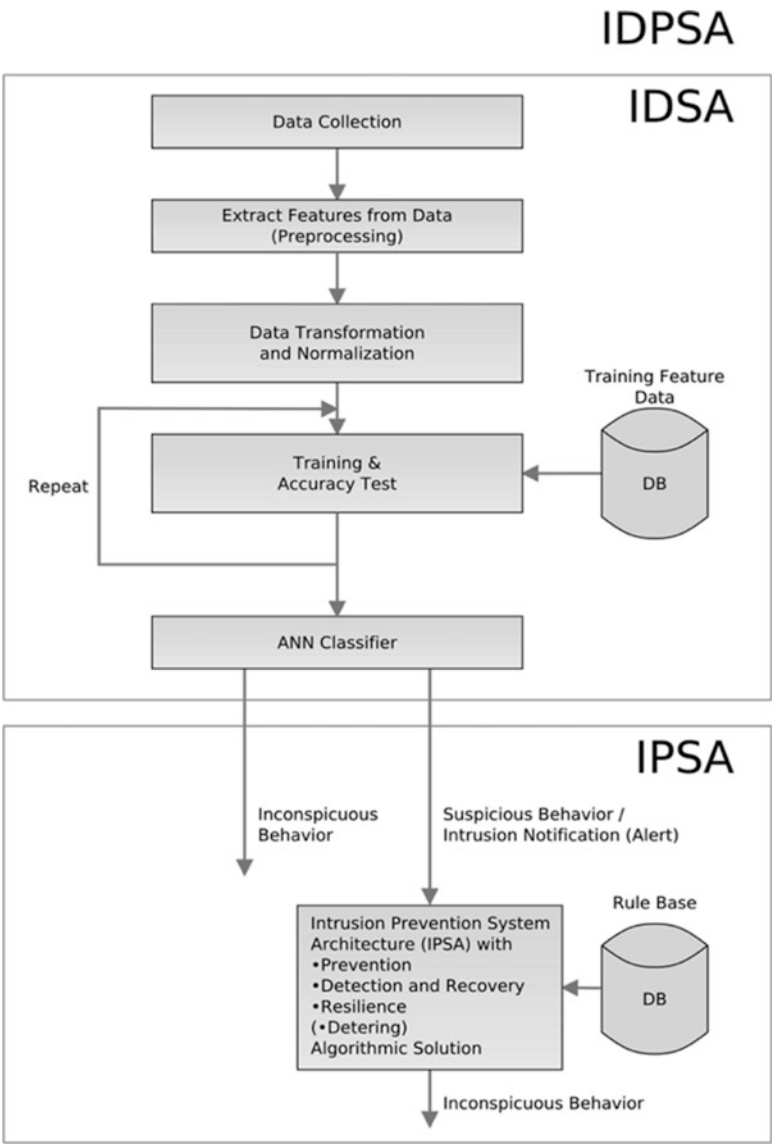

Fig. 6.26 Intrusion detection and prevention system architecture (IDPSA)

- IP address
- Payload size and type
- Port

- Source and destination MAC
- Time to live

The type of artificial neural net is the next important step. In IDSA a feed forward neural network (FFNN) type is used, consisting of an input layer with as many neurons as number of features used for classification, two hidden layer with, for example, less number of neurons and a final output layer. The FFNN requires training based on specified features. The step after training the FFNN is to test it in place with the features assigned to normal and abnormal behavior based on a performance metrics which describe the accuracy of the detection rate and false alarm rate of the IDSA. Accuracy is calculated by ratio of correct classification to the total test data set. Detection rate is the ratio of the number of correct detection to the total number of attacks. In this context an abnormal or anomaly behavior can be received using a statistical based threshold approach.

As reported in (Karim and Proha 2014), numerous static, dynamic, and hybrid solutions are available for analyzing patterns and signatures in program codes and the behavior of program executions in order to identify the presence of malicious agents in the system under test, thereby helping to disable them. In real-time CPSs, which are used for mission-critical tasks, intrusion can be detected through static timing analysis.

In Zimmer et al. (2010), three mechanisms for time-based intrusion detection are described that detect the execution of unauthorized instructions in real-time CPS environments. Such intrusion detection utilizes information obtained by static timing analysis. For real-time CPSs, timing bounds on code sections are available as they are already determined prior to the schedulability analysis. The Zimmer et al. (2010) paper demonstrates how to provide microtimings for multiple granularity levels of application code. Through bound checking of these microtimings, techniques have been developed to detect intrusions (i) in a self-checking manner by the application and (ii) through the operating system scheduler (OSS), which are novel contributions in the real-time CPSs domain.

Another option is testing for stability and resiliency because the complex software systems found in today's vehicles are prone to attacks. Automakers and their OEMs need to fully assess vehicle security to ensure a stable and resilient system. To test for stability and resiliency, several methodologies are used:

- *Functional and Performance Test*: Validates security components under valid traffic and cyberattack conditions.
- *Impairment Test*: Validates performance when communication is impaired; typically used with delayed, dropped, or erroneous packets.
- *Resiliency Test*: Validates operation under degraded or failure conditions, such as sensor failure, actuator failure, etc.
- *Stress Test*: Validates system or components beyond normal operational capacity to observe how the system or components operate.

With regard to tests, another important strategy is the security penetration test (SPT). This test aims to identify weaknesses in IT systems of a defined target environment on the basis of a systematic methodology. When implementing SPTs, the same techniques, tools, and expert knowledge are used, which are also used by real attackers. Hence, experienced penetration testers are required which use automated and manual test procedures to present realistic attack scenarios. In addition to technical analyzes, social-level attacks can also be part of a SPT to test the security awareness of employees of a company with regard to the dissemination of information and the conscious or unconscious use of unauthorized applications. Depending on the targeted object, the vehicle, the following strategies for SPTs can be distinguished as described by TechTarget networking.de:

- *External Penetration Strategy*: External tests deal with attacks on the network. The methods used are carried out from outside the vehicle to be attacked, i.e., through the Internet. This test can be carried out with no or complete knowledge of the vulnerable technical environment. Typically, this penetration test begins with public available information about the vehicle, subsequent network spanning, and others.
- *Internal Penetration Test Strategy*: Internal tests are carried out within the vulnerable technical environment. The penetration test simulates an attack on the internal network. The focus here is to understand what might happen if the network was successfully penetrated or what an authorized user could do to capture specific information resources of the compromised network. One important attack is sniffing which is used to a considerable extent with internal penetrations tests. The sniffer or the computer is directly connected to the network in promiscuous mode, which allows a considerable amount of information to be collected. For sniffing, a variety of free and commercial tools are available, such as Wireshark (the former Ethereal), the Microsoft Message Analyzer (the successor to Netmon), or the Viavi Observer Analyzer.
- *Blind Test Strategy*: In blind tests, one tries to simulate the actions and procedures of a real hacker. As with a real hacker attack, the test team has only limited or no information about the vehicle before performing the penetration test. The penetration test team uses public available data to collect information about the targeted object and perform the penetration tests. These blind tests can provide a lot of information about the targeted object that would otherwise remain unknown – for example, this type of penetration tests can raise problems such as additional Internet access points, directly connected networks, and public available confidential/protected information. However, blind tests are more time-consuming and expensive because the necessary effort of the test team for the target search is higher.
- *Double-Blind Test Strategy*: Double-blind tests are an important test component, since it is possible to check the security monitoring and identification of security incidents as well as the escalation and reaction procedures of the targeted object.
- *Targeted Testing Strategy*: In the case of targeted or systematic tests, sometimes referred to as a lights-turned-on approach, the penetration test team are involved

in the test. The test activities and the information regarding the target and network design are generally known. Targeted penetration testing can be more efficient and cost-effective if the goal of the test is more focused on the technical side or design of the network, rather than on incident response and other workflows of the targeted object. In contrast to blind tests, a systematic test can be carried out in less time and with less effort. The only difference is that this may not provide a complete picture of the targeted object's vulnerabilities and reactivity.

In addition to the aforementioned methods, a large number of distributed computing resources connected by a network representing a so-called "cloud" can be used to deliver essential vehicle applications with regard to connected vehicle needs. Thus, in a connected vehicle, the cloud allows challenges in the vehicle ecosystem to be met, which will increase the value of current business and induce new third parties to take part in the cloud (see Sect. 6.5.4).

Furthermore, vehicle owners will also be able to connect to the vehicle remotely from other devices which, unfortunately, will open the door to new intrusion points for cyberattacks.

6.5 Functional Safety and Security

The growing complexity and networking of today's automotive systems increases the importance of functional safety and security. Safety and security issues have been treated separately for the most part.

Safety systems are set up and operated totally disjoined from other systems, having their own physically separate system and gateways when connecting with others. For functional safety, the absence of reaction is required and has to be proven, usually resulting in a limited read-only access to the safety system.

Trends such as remote access via the Internet require rethinking this separation and setting up concepts for systems that allow common usage safely and securely. This can be achieved by embedding security measures to guarantee the correct execution of functional-safety-relevant operations. This requires that communication systems offer flexible frameworks that on the one hand run the correct utilization of resources needed for safety and, on the other hand, offer respective services, such as access rights or authentication, to other applications.

Using redundancies by integrating safety-critical, security-relevant, and standard operations within a single communication network also allows for cost-efficient solutions. Hence, these trends break up the isolated structure of networking and, therefore, enable new risks and threats concerning safety and security, and set new challenges for the safety and security measures in automotive systems.

6.5.1 Security for Wireless Mobile Networks

As previously mentioned, wireless technologies are bringing significant changes to communication networking and services. Due to their unique features, such as a

shared medium, limited resources, and dynamic topology, wireless ad hoc networks are vulnerable to a variety of potential attacks. However, common security measures employed for wired networks are not enough to protect the nodes of the networks against complex attacks. Therefore, a new line of defense, the intrusion detection approach, has been added. In this section, the wireless mobile networks, along with their security issues, are introduced. The most obvious characteristic of wireless networks is that communication takes place over a wireless channel, usually a radio channel. Such a channel suffers from a number of vulnerabilities:

- *Address Spoofing*: Scenario in which a network node uses the address of another node to exploit privileges granted to the legitimate authorized user of the identity. In WLANs, this can be done by changing the media access code MAC address of a network interface.
- *Eavesdropping*: Placing an antenna at an appropriate location, a cyberattacker can overhear information that the authorized user transmits or receives. Eavesdropping is often used to carry out attacks, notably passive attacks.
- *Location Tracking*: Tracing calls made by a cellular network or using network sensors.
- *Medium Access Control*: Following the rules of a MAC protocol in an attempt to obtain more than a fair share of a WLAN bandwidth.
- *Unauthorized Transmission*: Injecting forged or replayed frames. Attack goal can be to illegitimately join the WLAN.

Passive attacks consist of listening to the communication network and analyzing the captured data without interacting with the network. Such cyber physical attacks can be illustrated by the weakness of wired equivalent privacy WEP, (see Sect. 6.2) a security protocol, specified in the IEEE Wi-Fi standard 802.11b, that is designed to provide a WLAN with a level of security and privacy comparable to what is usually expected of a LAN. WEP seeks to establish protection similar to a wired network's physical security measures by encrypting data transmitted over the WLAN to protect against misdeeds. In case of an unprotected WLAN, the cyberattacker does not need to have physical access to any device to connect to the network. Hence, WEP is intended to transform this simple access into a difficult one by increasing the level of difficulty of attacking WLANs which comes from:

- *Broadcasting Nature of Radio Communications*, because eavesdropping on wireless transmissions is simple. This can be prevented by encrypted messages. There are two main families of encryption techniques: stream ciphers and block ciphers.
- *Connecting to the WLAN*, which does not require physical access to the network access point. Thus any device can try to illegitimately use the services provided by the WLAN, which can be prevented by authentication of the mobile STAs before allowing their connection to the WLAN.

Authentication of an STA is based on a simple challenge-response protocol. Once authenticated, the STA communicates with the access point by means of encrypted

messages. The key used for encryption is the same as the one used for authentication. The encryption algorithm specified by WEP is based on the four-line stream cipherRivest Cipher 4 (RC4). Stream ciphers produce a long pseudorandom byte sequence out of a short secret seed value. This pseudorandom sequence is fused with the clear text message using the *XOR* operation to generate the encrypted message. WEP works in the same way. The sender of a message M initializes the RC4 algorithm with the secret key and connects the pseudorandom sequence K generated by RC4 logically through the *XOR* operation with M. The receiver of the encrypted message $M \oplus K$ uses the same secret key to initialize the RC4 algorithm that produces the same pseudorandom sequence K whereby K is connected through *XOR* operation to the encrypted message to obtain the message:

$$(M \oplus K) \oplus K = M.$$

As mentioned in Butayán and Hubaux (2007), this description is not precise enough. There is more to be taken into account than what WEP does when encrypting messages. It can be seen that if encryption is appropriate, then every message would be encrypted with the same pseudorandom sequence K.

Let's assume that a cyberattacker is eavesdropping on two encrypted messages, $M_1 \oplus K$ and $M_2 \oplus K$. With regard to the XOR operation of these two messages, we receive

$$(M_1 \oplus K) \oplus (M_2 \oplus K) = M_1 \oplus M_2$$

which is equivalent to one message being encrypted with the other, but clear messages are far from being pseudorandom sequences. Thus, $M_1 \oplus M_2$ is a weak encryption; and the cyberattacker is likely to be able to break it using the statistical properties of the clear messages.

To address this problem, WEP appends an initialization vector (IV) to the secret key before initializing the RC4 algorithm, where the IV changes for every message, as described in Butayán and Hubaux (2007). This ensures that the RC4 algorithm produces a different pseudorandom sequence for every message. The receiver should also know that the IV will be able to decrypt the messages received. For this reason, the IV is sent in a clear message together with the encrypted message. Figure 6.27 illustrates the WEP encryption and decryption procedure after Butayán and Hubaux (2007).

From Fig. 6.27, it can also be seen that before encryption, the sender attaches an integrity check value (ICV) to the clear message. The purpose of this value is to enable the receiver to detect any malicious modifications of the message by a cyberattacker. In case of WEP, ICV is a CRC value computed for the clear message. As a CRC value alone cannot enable the detection of malicious modifications, because the attacker can compute the new CRC value for the modified message, the CRC value is also encrypted in WEP. The rationale is that in order to modify the message in an unnoticeable way, the cyberattacker must now encrypt the new CRC

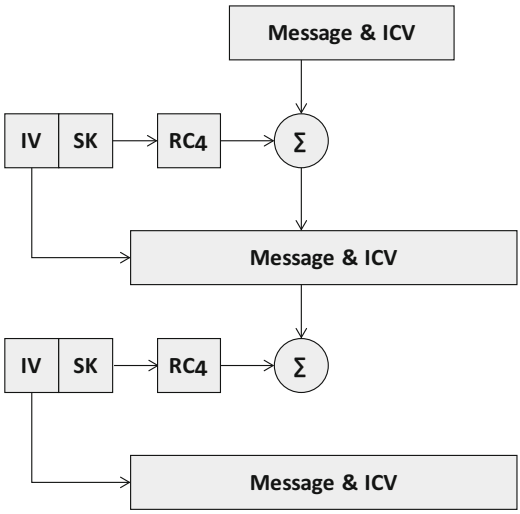

Fig. 6.27 Encryption and decryption in WEP with SK as the security key, modified after (Butayán and Hubaux 2007)

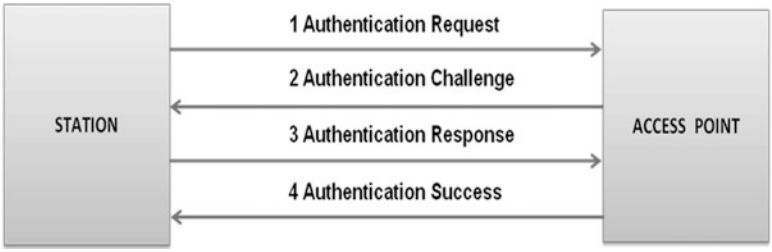

Fig. 6.28 WEP authentication

value but cannot do this without the knowledge of the secret key (Butayán and Hubaux 2007).

WEP also includes a device-level authentication mechanism through which STATION must provide, to the ACCESS POINT, a proof of ownership of the key they share for which four messages are exchanged, as shown in Fig. 6.28.

STATION makes a request. ACCESS POINT shown in Fig. 6.28 sends a challenge, such as a 128-bit random value. STATION sends a response, e.g., a 128-bit random value encrypted with the WEP stream cipher. ACCESS POINT decrypts the response. If the decrypted response matches the original challenge value, then a positive authenticate response is returned to STATION. WEP authentication is one way, i.e., the ACCESS POINT is not authenticated by STATION (Das et al. 2012). After completion of the authentication phase, subsequent traffic is not authenticated. Therefore, the protocol is vulnerable to the authentication spoofing attack. A cyberattacker may obtain the key by using XOR operation for the intercepted challenge value and its response. The key stream may be used by the cyberattacker to create proper responses to new challenges (Housley and Arbaugh 2003).

6.5.2 Security for Sensor Networks

Recent technological advances have made it possible to deploy wireless sensor networks consisting of a large number of functional sensor nodes that communicate over short distances through wireless links (Akyildiz et al. 2002). The desirable features of sensor networks have motivated many researchers to develop protocols and algorithms to support the various applications of sensor networks. A common use of sensor networks in the automotive domain is to sense and monitor cyber-physical systems and/or components. Two access control approaches are in use for wireless sensor networks:

- *Uni-Access Scheme*: Mainly used to access one sensor node at a time. The user can directly access the data on any sensor node in the network without going through the base station, and a sensor node can protect its data so that only authorized users can access it.
- *Multi-Access Scheme*: Applies public key cryptography to achieve an additional feature, which allows a user to access data on many sensor nodes via a single query.

In sensor networks, one can differentiate between two attack forms in (Das et al. 2012):

- *Attacks on Communication*: A cyberattacker can easily perform a denial-of-service (DoS) attack by jamming the wireless channel and disabling the network operation. This attack is easy to intrude and common to the protocols in every sensor network.
- *Attacks on Sensor Nodes and Users*: Once a sensor node is compromised, the cyberattacker has full control of it. The attacker can learn keys and all sensed data stored on the compromised sensor node.

Typical security problems in sensor nodes include:

- *False Node*: An intruder may insert a node into the sensor network that feeds false data or prevents the passage of true data. Such problems are known to occur in distributed network systems as well as ad hoc networks.
- *Legitimate Addition of a Node to an Existing Sensor Network*: If a sensor node needs to be replaced or another sensor node needs to be added to an existing sensor network, securely integrating the new sensor node into the existing sensor network is an issue.
- *Passive Information Gathering*: If communication between sensors or between sensors and base stations is in the clear, then an intruder with an appropriately powerful receiver and antenna can easily pick up the data stream. If the thumbed information is encrypted, then it is important to know which cryptographic approach has been used by the compromised sensor node.

- *Subversion of a Node*: A particular sensor might have captured information stored on it, such as the key, which might be obtained by the intruder. If a sensor node has been compromised, then the issue is how to exclude that sensor node, and that sensor node only, from the sensor network.

Furthermore, sensor network security has some unique features that do not exist in other networks. For example, a sensor node has limited memory space so that the number of keys that can be stored in its memory, as well as the variables for asymmetric cryptographic algorithms, is limited. Moreover, any security solution with a static configuration may not be suitable for ad hoc sensor networks because sensor nodes have mobility, and the sensor network topology may change frequently. Sensor nodes have to continuously detect possible intrusions because their neighbor nodes are not fixed. Similarly, a malicious node with mobility can roam in a sensor network and attack different parts of the network (Xiao 2006).

However, the encryption-decryption techniques devised for traditional wired networks cannot feasibly be applied directly to wireless networks; wireless sensor networks (WSNs) in particular. Utilizing any encryption scheme requires transmission of extra bits, hence extra processing, memory, and power which are important resources for the sensor nodes. Applying security mechanisms, such as encryption, could also increase delay, jitter, and packet loss in WSNs (Saleh and Khatib 2005). Moreover, some critical issues arise when applying encryption schemes to WSNs, such as how the keys are:

- Assigned to a new sensor added to the network
- Generated or disseminated
- Managed
- Renewed, to ensure robust security for the network
- Revoked

As minimal human or no human interaction with the sensor nodes is a fundamental feature of WSNs, how the keys can be modified from time to time is an important issue for encryption because adoption of preloaded keys or embedded keys would not be an efficient solution (Pathan et al. 2006).

A holistic approach reported in Avancha (2005) aims to improve the performance of WSNs with regard to security, longevity, and connectivity under changing environmental conditions. The holistic approach to security is concerned with involving all layers of WSNs to ensure the overall security of a network, as shown in Fig. 6.29.

For such a network, a single security solution for a single layer might not be an efficient solution, where employing a holistic approach could be the best option. The holistic approach has some basic principles, such as security has to be ensured for all layers of the protocol stack. If no physical security for the sensors is ensured, the security measures must be able to exhibit a graceful degradation if some of the sensors in the network are compromised, out of order, or captured by an adversary. Security measures should be developed to work in a decentralized fashion. If security is not considered for all of the security layers, e.g., if a sensor is somehow

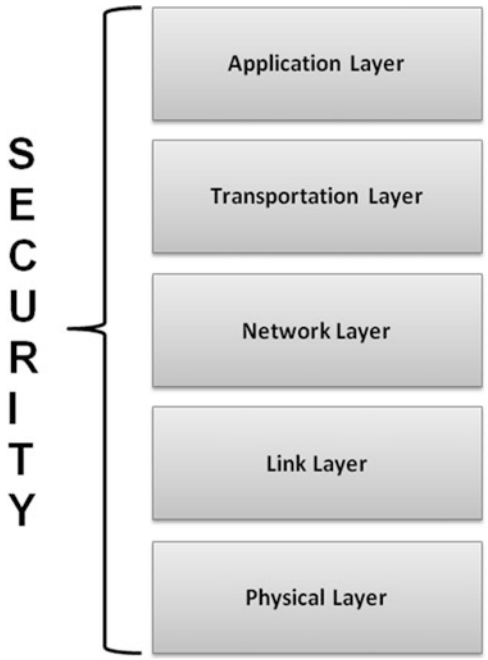

Fig. 6.29 Holistic view of security in wireless sensor networks

captured or jammed in the physical layer, the security for the overall network breaks despite the fact that there are some efficient security mechanisms working in other layers. By building security layers using a holistic approach, protection is established for the overall network (Pathan et al. 2006).

6.5.3 Platform Security

Platform security refers to the security architecture, tools, and processes that ensure the security of an entire computing platform (hardware, software, network, storage, and other components) by using a centralized security architecture or system. Platform security secures all components and layers within a platform. This allows for the elimination of individual security measures and the use of multiple applications/services to secure different layers of an ICT environment. Security at the platform level simplifies the security process for information technology and developers. However, once the security is cracked, the entire platform is vulnerable. Thus, a trusted platform module (TPM) is required, which is a hardware device that is basically a secure cyber-physical controller with added cryptographic functionality. It works with supporting software and firmware to prevent unauthorized access to a platform. The TPM contains a hardware engine capable of performing up to 2048-bit Rivest-Shamir-Adleman (RSA) encryption/decryption. The TPM uses its built-in RSA engine during digital signing and key wrapping operations.

6.5.4 Cloud Computing and Data Security

Cloud computing is a new information technology infrastructure in which both, the application, delivered as a service to users at anytime, anywhere whenever the Internet is available, and the computing resources, hardware and systems software in data centers, may be provided. Services provided by the cloud can be at different levels, described by the X-as-a-Service (XaaS) model, whereby X could be:

- Hardware
- Infrastructure
- Platform
- Software

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. When deciding to use cloud computing, users need to be aware that in addition to the services provided, i.e., ability and system performance, security should be of particular concern. Therefore, cloud-based security services are an important issue when migrating from dedicated hardware solutions to cloud-based security services using the XaaS model. At its core, cloud computing is used to describe data acquisition or distribution through the Internet and wireless networks. In the XaaS model, the application data is generally hosted in the cloud and made available to users via an Internet interface. XaaS users often download a thin client, which gives them access to the application via a web browser. The increase in virtual and cloud networks is boosting demand for cloud-based security because data and applications are now portable and distributed across a wide variety of networks. This means that security applications need to live as software in the cloud, rather than on dedicated hardware devices, thus protecting specific potential intrusion points of the vehicle network.

Today's vehicle users will be able to access applications from a screen in the vehicle, thereby enjoying the same level of digital services that they have in their homes, at work, or on the go via smart devices.

Moreover, cloud computing could also bring additional benefits to the average vehicle in many ways. One of these is actually related to drive dynamics. New vehicles often have electronically adjustable suspensions; with cloud computing, they could be more automated, providing a much better customized drive. The same is possible for electronically disconnecting sway bars and other off-roading features on some Jeeps and multipurpose sport utility vehicles SUVs. Another area is bringing personalized data into the cabin of a vehicle. For drivers, this means that their data from online calendars and contacts, a personal music library, and other data will travel with them and be available right at their fingertips. Thus, data protection in cloud computing is a crucial security issue. Hence, before moving into the cloud, users should clearly identify the data to be protected and classify data based on its security implications. Therefore, the security classification must be specified because different types of data may have different value and hence different security implications for confidentiality, integrity, and availability (CIA).

In cloud computing, data security has become more complicated because users may be confronted with all kinds of cyberattacks with regard to the intrinsic cloud characteristics. This requires an understanding of the potential security threats to identify where the cyberattackers may come from and what kind of cyberattacks they may launch. As reported in (Das et al. 2012), there are two types of cyberattackers:

- *Insiders*: Are users with authorized access privileges inside the cloud organization or at the cloud service provider's site and possibly the cloud service provider itself. They can launch serious attacks by:
  - Obtaining control of the virtual machines
  - Gaining access to sensitive information by logging all communication information of other cloud users, thereby abusing their privileges

  Therefore, cloud users should establish a trusted relationship with cloud service providers. The occasional misbehavior of a cloud service provider may be any, or a combination, of the following:
  1. Colluding with a small number of malicious users for the purpose of harvesting data files and their contents.
  2. Deciding to hide data corruption caused by server hacks or Byzantine failures, thereby maintaining reputations. The Byzantine model (Dolev 1982) assumes a system with n components and an adversary that may compromise up to $k < n$ components. Therefore, the identified vulnerabilities V_j are

$$V_j = f(t_i, q_j)$$

where

$$V_j \subseteq V$$

is a set of

$$j \in [0 : k]$$

faulty components of q_j. The threat transition function $D_f(t_j)$ then ^is

$$D_f(t_i) : V_j \overset{t_j, a}{\rightarrow} V_s$$

where

$$V_j \subseteq V_s$$

which is an adversary that has compromised the components of V_j and was restricted to attacking those states with

$$V_s \supseteq V_j$$

This defines the allowable system transitions that the adversary can exploit. But faulty components cannot be recovered with this model.

- Gaining data information by eavesdropping and monitoring network traffic (Kao and Marculescu 2006).
- Neglecting, keeping, or deliberately deleting rarely accessed data files, thereby saving resources.

For valuable and/or sensitive data or services, cloud users should implement their own security protection mechanisms, such as cryptographic protection.

Outsiders: Cloud computing could be vulnerable to malicious attacks from the Internet. Outsider attackers can launch passive attacks, such as eavesdropping on the network traffic, and active attacks, such as phishing legitimate users credentials, manipulating network traffic, and probing the cloud structure.

In cloud computing, sensitive data pooled in the cloud demands that the cloud data storage and sharing service be responsible for secure, efficient, and reliable distribution of data to a potentially large number of authorized users (Das et al. 2012). One way of providing a secure data access service is through cryptographic methods. The data owner and data user encrypt data before storing it in the cloud, retaining the secret key.

In the literature, related mechanisms can be found in the areas of shared cryptographic file systems and access control of outsourced data (Capitani di Vimercati 2007; Kallahalla et al. 2003; Goh et al. 2003).

Since diverse mobile technologies are available, mobile cloud computing supports and adapts itself to multiple mobile platforms and devices. Thus, mobile-device-centric cloud computing consists of an infrastructure formed by the mobile devices themselves. In this context, security concerns may depend on how the infrastructure is organized to deliver mobile cloud security. A cloud support service specific to mobile devices has been investigated through a dedicated infrastructure and a related model (Satyanarayanan et al. 2009).

Cloud providers, for example, Amazon, Azure, and Google, manage the security and availability of their cloud infrastructure like any other larger enterprise. They monitor and investigate security incidents or events. Cloud service providers (CSP) must therefore distinguish between legitimate penetration tests (see Sect. 6.4.2) by customers and real attacks. If customer tests trigger the wrong countermeasures, connections can be routed into a DDoS black hole or intrusion prevention systems can be activated. This not only costs the CSP time and valuable resources, because of the shared infrastructure of cloud systems, but can also have a negative impact on other customers.

Before testing, it is important to know the limits of cloud penetration testing. This means, for example, that one is aware of its responsibility. This changes depending on what type of system is checked because IaaS, PaaS, or SaaS each have different requirements. An IaaS environment, for example, allows a much more aggressive approach than SaaS, which is mainly due to the fact that IaaS often has numerous users (vehicles) working and/or connected on the system and that a failure would have a massive impact on them which is not the case with SaaS. A concentration test can take a system completely offline. This is not a problem if the OEM or Tier 1 supplier company owns the server completely, but a huge problem when other users are taken offline.

6.5.5 Functional Safety

Functional safety is part of the overall safety of a vehicle system, or a component of it, that depends on the cyber-physical system or its components for operating correctly in response to its inputs, including safe management of likely operator errors, hardware failures, and environmental changes. Functional safety is intrinsically end-to-end in scope, which means that it has to treat the function of a system or subsystem or component as part of the function of the whole system. This means that while functional safety standards focus on electrical, electronic, and programmable systems (E/E/PS), the end-to-end scope, in practice, of functional safety methods has to extend to the non-E/E/PS parts of the system that the E/E/PS actuates, controls, or monitors (URL11 2016).

Functional safety is achieved when every specified safety function is carried out and the level of performance required of each safety function is met. This is normally achieved by a process that includes the following steps as a minimum (URL11 2016):

- *Identify the Required Safety Functions*: This means hazards and safety functions have to be known or identified.
- *Assess the Risk Reduction Required by the Safety Function*: This involves a safety integrity level (SIL), performance level (PL), or other quantification assessment. An SIL applies to an end-to-end safety function of the safety-related system, not just to a component or part of the system.
  - *Automotive Safety Integrity Level (ASIL)* is a risk classification scheme defined by ISO 26262, *Functional Safety for Road Vehicles Standard* which is an adaptation of the SIL used in IEC 61508 for the automotive industry. This classification helps define the safety requirements necessary to be in line with the ISO 26262 standard. ASIL is established by performing a risk analysis of a potential hazard by looking at the severity, exposure, and controllability of the vehicle operating scenario. The safety goal for that hazard in turn carries the ASIL requirements. There are four ASILs identified by the standard: ASIL A is comparable to SIL-1, ASIL B/C is comparable to SIL-2, and ASIL D is comparable with SIL-3. For SIL-4, no comparison exists with ASIL. ASIL D dictates the highest integrity requirements for the product and ASIL A the

lowest. However, ISO 26262 does neither provide normative nor informative mapping of ASIL to SIL. ASIL is a qualitative measurement of risk, while SIL is quantitatively defined as the probability or frequency of dangerous failures, depending on the type of safety function. Thus, in IEC 61508, higher-risk applications require greater robustness to dangerous failures. Hazards that are identified as quality management (QM) do not dictate any ASIL safety requirements (URL12 2016).

- *Ensure Safety Function Performs to the Design Intent*: This includes under conditions of incorrect operator input and failure modes. The design and life cycle are managed by qualified and competent engineers carrying out processes to a recognized functional safety standard. In Europe, that standard is IEC EN 61508 or one of the industry-specific standards derived from IEC EN 61508 or some other standard, such as ISO 13849.
- *Verify the System Meets the Assigned SIL (ASIL, PL, or agPL)*: This can be done by determining mean time between failures (MTBF) and the safe failure fraction (SFF), along with appropriate tests. SFF is the probability of the system failing in a safe state. The critical or dangerous state is identified from a failure mode and effects analysis (FMEA) or failure mode effects and critical analysis (FMECA) of the system under test.
  - *MTBF*: Predicted elapsed time between inherent failures of a system during operation which can be calculated as the arithmetic mean time between failures of a system using the following equation:

$$\text{MTBF} = \frac{\sum(\text{start of downtime} - \text{start of uptime})}{\text{number of failures}}$$

  - *SFF*: Takes into account any inherent tendency to fail toward a safe state. SFF is the sum of the rate of safe failures plus the rate of detected dangerous failures divided by the sum of the rate of safe failures plus the rate of detected and undetected dangerous failures. It is important to realize that the only types of failures to be considered are those which could have some effect on the safety function. SFF can be calculated using the following equation:

$$\text{SFF} = \frac{(\sum \lambda_S + \sum \lambda_{DD})}{(\sum \lambda_S + \sum \lambda_D)}$$

where

$$\begin{aligned}
&\lambda_S : \text{Rate of safe failure} \\
&(\textstyle\sum \lambda_S + \sum \lambda_D) : \text{Overall failure rate} \\
&\lambda_{DD} : \text{Rate of detected dangerous failure} \\
&\lambda_D : \text{Rate of dangerous failure}
\end{aligned}$$

  - *FMEA*: The first step of a system reliability study involves reviewing as many components, assemblies, and subsystems as possible to identify failure modes and their causes and effects. For each component, the failure modes and their

resulting effects on the rest of the system are recorded in a specific FMEA worksheet. FMEA can be a qualitative analysis but may be put on a quantitative basis when mathematical failure rate models are combined with a statistical failure mode ratio database (URL13 2016).
  - *FMECA*: An extended FMEA indicates that a criticality analysis is performed, too.
- *Conduct Functional Safety Audits*: Examine and assess the evidence that the appropriate safety life cycle management techniques were applied consistently and thoroughly in the relevant life cycle stages.

Neither safety nor functional safety can be determined without considering the vehicle cyber-physical system as a whole and the environment with which it interacts. Functional safety is inherently end-to-end in scope.

6.6 Car Hacking Examples

Today's vehicles can be understood as a complex network of ICT systems. As vehicles become increasingly computerized, their attack surfaces also grow and increase. Worldwide security research demonstrates a huge number of vulnerabilities in vehicle electronic systems, showing that automakers have not placed enough emphasis on developing secure vehicular ECUs and communication systems. ECUs receive inputs from sensors and makes adjustments to a series of actuators controlling, e.g., the operation of the engine's physical components. This allows for ignition timing and the fuel/air mixture to be dynamically adjusted in real time, which can save fuel and optimize performance. Prior to the use of ECUs for engine management, these functions were controlled mechanically (Eyal 2007).

To understand the magnitude of the security problems facing today's vehicles, it is first necessary to address the interconnectivity of today's vehicle components which are managed by a vehicle's onboard computer systems. Once hackers have access to personal and other information from vehicle systems, they are able to find myriad new ways to use it. For example, GPS information could be used to track a driver's habits and schedule.

Vehicle hacking is the manipulation of the code in a vehicle's ECU to exploit a vulnerability and gain control of other ECU units in the vehicle. An excellent timeline of recent vehicle hacks is given in (Currie 2015), which we have the author's permission to use.

When the CAN system bus was developed in the mid-1980s, its designers certainly did not envision that the bus would one day be targeted by attackers seeking to take over or otherwise manipulate the function of an automobile (see Sect. 6.4.2). As recently as 10 years ago, hacking vehicles received less media attention and was not a worry to most vehicle users. With regard to the last decade, and particularly the last several years, vehicle hacking has become a real concern. In a 2015 study by Kelley Blue Book®, for which members of the vehicle-buying public were polled, it was found that 78% of study participants believed vehicle

hacking "will be a frequent problem in the next 3 years or less" (PR Newswire 2015). This perception among the general public is mostly a result of several recent high-profile vehicle hacks. The timeline below summarizes some of the more notable vehicle hacks that have recently occurred.

6.6.1 2010: Vehicles Disabled Remotely via Web Application

One of the first widely reported accounts of vehicle hacking occured in 2010 when a disgruntled former employee of an Austin, Texas, car dealership sought revenge against his former employer (Poulsen 2010). This attack did not involve any hacking of the actual vehicles themselves. Nonetheless, the attacker was able to physically disable the vehicles of owners without their knowledge or consent. The former dealership employee used stolen credentials to log into a web application that allowed remote access to functions of customers' vehicles, including the engine immobilizer and the horn (Poulsen 2010). This web application's intended purpose was to let dealership personnel immobilize the vehicles of customer who failed to make their loan payments on time. In fact, it ended up being used to cause mayhem as vehicle owners found themselves locked out of their vehicles with the horns constantly honking (Poulsen 2010).

The web application used by the dealership, in this case, was WebTeckPlus from Pay Technologies, LLC (Payteck 2003). The WebTeckPlus application provides a web portal for dealership employees to interface with PayTeck electronic controllers installed in customers' vehicles. The PayTeck hardware consists of an electronic keypad and controller that is installed inside the customer's vehicle. The controller is wired into the vehicle's engine immobilizer and horn. Each time a customer makes a payment on time, they are given a new code to enter into the electronic keypad. If the correct code is entered, the vehicle will continue to function normally. If payment is late and a code is not entered into the keypad on time, the controller will activate the engine immobilizer, rendering the vehicle useless. The WebTeckPlus application also allows a dealership employee to log in and remotely disable a particular customer's vehicle at will, if necessary. The PayTeck hardware and WebTeckPlus software allow dealerships to save time and money by avoiding having to repossess vehicles.

This hack can best be summarized as an unauthorized intrusion of a web-based application, which is certainly nothing new. The perpetrator faced computer intrusion charges (Poulsen 2010). However, this particular incident highlights the link between a vehicle's critical control systems and the digital transformation of the modern connected world, showing how that link can potentially be exploited by someone with malicious intent.

6.6.2 2010 and 2011 CAESS Experimental Analysis

In 2010, a group of researchers from the Center for Automotive Embedded Systems Security (CAESS) – a joint venture between the University of California, San Diego and the University of Washington – released a research paper entitled *Experimental Security Analysis of a Modern Automobile* (Koscher et al. 2010). The team conducted a range of lab experiments and road tests and found that it was possible to manipulate a vehicle's functions by injecting messages on the CAN bus (Koscher et al. 2010). The researchers successfully demonstrated that a would-be attacker could disable the brakes, selectively brake individual wheels on demand, stop the engine, falsify information on the vehicle's speedometer, and more (Koscher et al. 2010).

Although the CAESS team highlighted serious security flaws in a modern vehicle system, their research was largely met with criticism. At the time, automakers and the media alike claimed that it was neither realistic nor plausible for an attacker to have wired access to a vehicle's CAN bus to be able to carry out this type of attack in the real world (Miller and Valasek 2014, 2015, p. 5ff).

The following year, in 2011, the CAESS team published a new research paper entitled *Comprehensive Experimental Analyses of Automotive Attack Surfaces* (Checkoway et al. 2011). This paper was a response to the media scepticism surrounding the team's previous findings. The team acknowledged that the previous threat model of an attacker having physical access to a vehicle's internal network had *justifiably been viewed as unrealistic* (Checkoway et al. 2011). This time, the researchers sought to analyze the external attack surface of a modern vehicle and determine whether an attack could be carried out remotely.

In analyzing the attack surface of a modern car, the CAESS team created the illustration shown in Fig. 6.30.

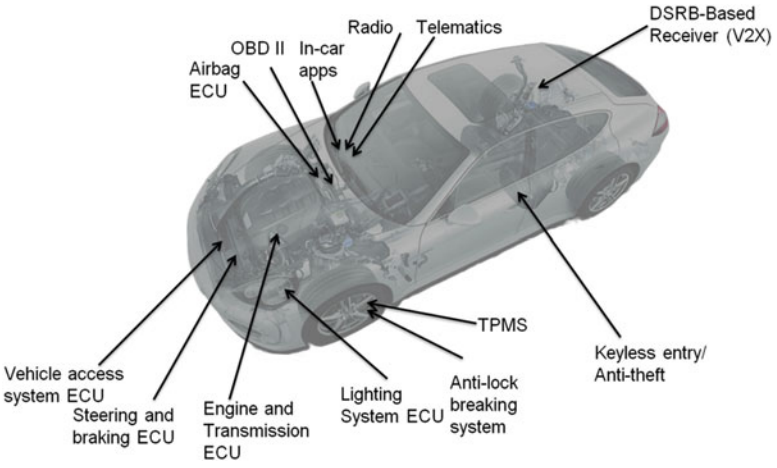

Fig. 6.30 Digital I/O channels on a modern vehicle

The Fig. 6.30 shows the different I/O channels on a modern vehicle, with each one representing a potential entry point for an attacker. The "lightning bolt" symbols represent possible sources of remote wireless access and control. As automakers continue to increase the connectivity of their vehicles, the attack surface only broadens. The vehicle's cellular, Bluetooth, and Wi-Fi systems make particularly attractive entry points for a would-be attacker.

Ultimately, the CAESS team found that it was possible to remotely exploit their test vehicle via a range of different vectors, including the radio's MP3 parser, the vehicle's Bluetooth system, and the cellular connection used for the vehicle's telematics system (Checkoway et al. 2011). From there, CAN messages could be injected on the bus as had been demonstrated in the group's previous findings.

This research was hailed by some as ground breaking because "it showed that vehicles were vulnerable to attacks from across the country, not just locally" (Miller and Valasek 2015, p. 5ff). Despite having answered their critics, the CAESS team's findings failed to garner much media attention or response from the automotive industry. This was due in part to the fact that the researchers did not share how their exploits could be replicated, nor did they reveal the specific vehicle they tested (Miller and Valasek 2015, p. 5ff). While it is understandable that the research team would choose not to release the details of their exploits so as not to aid the "bad guys," this also made the findings a lot easier for automakers and the general public to shrug off.

6.6.3 2013 Miller and Valasek Physical Hack

A more recent high-profile case of vehicle hacking came from researchers Charlie Miller and Chris Valasek. Working with an $80,000 grant from the Defense Advanced Research Projects Agency (DARPA), Miller and Valasek were tasked with finding security vulnerabilities in automobiles and published their findings in 2013 (Greenberg 2013). They conducted a series of real-time demonstrations for journalists and security professionals, before going on to present their findings at the 2013 DEF CON® 21 Hacking Conference in Las Vegas, Nevada. Specifically, Miller and Valasek targeted the systems of a 2010 Ford® Escape and a 2010 Toyota® Prius (Greenberg 2013). They were essentially able to reverse engineer the vehicles' CAN bus communications to demonstrate "everything from annoyances like uncontrollably blasting the horn to serious hazards like slamming on the Prius' brakes at high speeds" (Greenberg 2013). The graphic in Fig. 6.31 lists many of the vehicular functions that Miller and Valasek were able to manipulate on their 2010 Toyota Prius test vehicle.

Some of these capabilities, for example, being able to jerk the steering wheel or slam on the brakes, propelled car hacking from a nuisance to a serious safety concern for automakers.

Miller and Valasek's method involved using a laptop PC running Windows XP hooked into the vehicle's OBD-II port via a series of cables (Miller and Valasek 2015, p. 23). The OBD-II port is traditionally used by mechanics and repair shops to

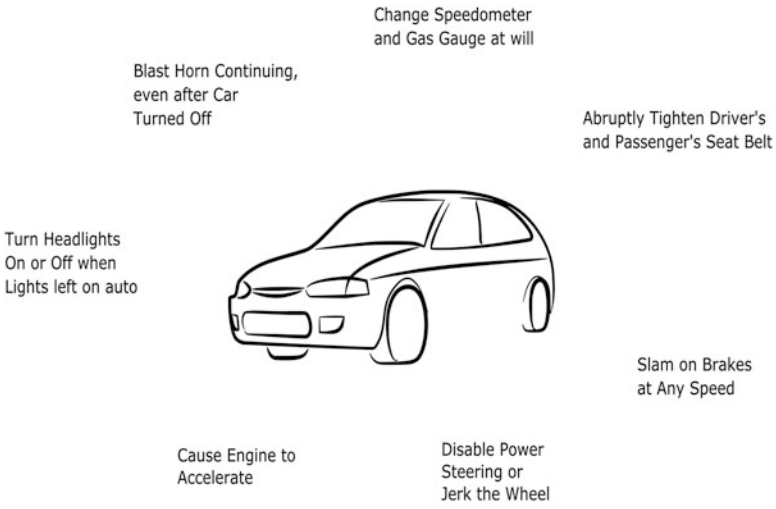

Fig. 6.31 Anatomy of an Automotive Hack (Greenberg 2013)

retrieve fault codes and diagnose problems with a vehicle, but it also represents an attractive point of entry for a vehicle security researcher or an attacker performing reconnaissance (see also Sect. 6.2.2). Miller and Valasek used a proprietary ECOM cable from EControls which was hooked to their laptop via a USB port. They then fashioned a custom ECOM-to-OBD-II connector to allow them to interface with the car's OBD-II port (Miller and Valasek 2015, p. 22). At this point, all Miller and Valasek had to do was listen and observe the CAN messages transiting the CAN bus to begin building a picture of which message corresponded to which vehicular function. The next step was to use the connected laptop to replay captured CAN packets, recording the vehicle's response each time. Finally, they crafted modified CAN packets and were able to manipulate the behavior of the vehicle (Miller and Valasek 2014, p. 26).

It is worth emphasizing again that in Miller and Valasek's 2013 car hacking demonstration, the researchers had physical access to the vehicle's CAN bus. The rationale behind this was that, according to the researchers, it had already been shown by prior scholarly research (Checkoway et al. 2011) that various interfaces, such as Bluetooth or a vehicle's telematics unit, could be hacked to allow for remote code execution (Miller and Valasek 2015, p. 4ff). Considering the challenge of gaining remote access to be trivial, the researchers sought to find out what could be accomplished after access had been gained (Miller and Valasek 2015, p. 4ff).

Following the release of Miller and Valasek's findings in 2013, the general public and big automakers seemingly failed to recognize the triviality of the prerequisite of gaining remote access to a vehicle's systems. Miller and Valasek faced scepticism for having demonstrated security flaws that required an attacker to be physically

located inside the vehicle with a laptop hooked up to the car's data port and, as in the case of the Prius demonstration, with the dashboard completely disassembled for ease of access (Greenberg 2013). Indeed, in response to Miller and Valasek's work, Toyota's safety manager, John Hanson, argued that "[Toyota's] focus, and that of the entire auto industry, is to prevent hacking from a remote wireless device outside of the vehicle" (cited in Greenberg 2013), indicating that Toyota was largely unimpressed by the hacking demonstration. Hanson went on to state, "we believe our systems are robust and secure" (cited in Greenberg 2013).

6.6.4 2015 Miller and Valasek Remote Hack

Charlie Miller and Chris Valasek made headlines again in 2015, this time for successfully demonstrating that an unaltered passenger vehicle – a 2014 Jeep® Cherokee, in this case – could be remotely exploited without the need for any physical access (Miller and Valasek 2015, p.6ff). Unlike their 2013 hack of a Toyota Prius and Ford Escape, this new research mimicked a real-world attack scenario by demonstrating both the ability to gain remote access and the ability to remotely execute code. Unlike the 2013 hack, which was largely met with incredulity by automakers, the 2015 hack prompted Fiat® Chrysler® Automobiles (FCA) to recall some 1.4 million vehicles for a critical security update and forced Sprint® Corporation to enhance the security of its cellular carrier network (Miller and Valasek 2015, p.87).

Miller and Valasek's Jeep hack took advantage of the vehicle's onboard connectivity features, in addition to the familiar lack of security controls on the CAN bus. Access was obtained through vulnerability in Uconnect®, a system that governs the vehicle's infotainment, navigation, built-in apps, and cellular communications (Greenberg 2015a). What made the Uconnect system so attractive to the pair of researchers was that in addition to being a hotbed of connectivity, Uconnect also contains a microcontroller in its head unit which can communicate with other modules on the vehicle's CAN bus (Miller and Valasek 2015, p.20). The hack also took advantage of a weakness in Sprint's cellular network, to which the vehicle's onboard telematics system was connected. The telematics system is used for real-time traffic data, in-car Wi-Fi, and other remote connectivity functions (Miller and Valasek 2015, p.32).

Through port scanning, they found Uconnect's D-Bus port (6667) to be open. D-Bus, also known as Diagnostic Bus, is a messaging system used to communicate between processes (Miller and Valasek 2015, p.28). Under normal conditions, the D-Bus service should not be subject to user input or manipulation, as it is intended for internal systems messages only. Miller and Valasek then found that prior to Sprint's fix, any 3G device on the Sprint network could communicate with the open D-Bus port on any Uconnect-enabled vehicle (Miller and Valasek 2015, p.46). For their attack, Miller and Valasek used a laptop computer tethered to a 3G cellular phone on the Sprint network. The laptop was then able to communicate directly with vehicles running the vulnerable Uconnect system (Miller and Valasek 2015, p.46).

Knowing the IP address of a specific vehicle allowed for a targeted attack; however, they also found that an Internet port scan of port 6667 across IP ranges 21.0.0.0/8 and 25.0.0.0/8 would yield responses from vulnerable Uconnect systems in vehicles nationwide (Miller and Valasek 2015, p.46). The researchers' scans of Internet-facing vulnerable devices turned up a wide range of vehicles across the country, from the Dodge, Ram, Jeep, and Chrysler brands, spanning multiple model years (Miller and Valasek 2015, p.47).

With access to the vehicle's Uconnect system obtained, Miller and Valasek then pivoted to the CAN-connected microcontroller in the Uconnect head unit. They were able to flash the controller with a new firmware version, one that they had reverse engineered to include their malicious code (Miller and Valasek 2015, p.50). With their modified firmware residing on the CAN bus, they were then able to send commands to many different vehicle components and control systems. During a press demonstration, Miller and Valasek showed that they were able to remotely set the air conditioning to its maximum cold setting, turn the radio on at full volume, and cover the windshield with wiper fluid making it difficult for the driver to see. More worryingly, they could also disable the transmission, control the throttle, and disable the brakes (Greenberg 2015a).

This latest automotive hacking demonstration propelled vehicle security into the general public's consciousness in a way that had not been seen previously. Shortly after news of the Jeep Cherokee hack hit the media, a Kelley Blue Book study of the car-buying public found that 72% of respondents were "aware of the recent Jeep Cherokee hacking incident" (PR Newswire 2015). Perhaps more tellingly, 41% of respondents said they would "consider this recent vehicle hacking incident when buying/leasing their next car" (PR Newswire 2015).

For the first time, an automotive hack had the very real potential to cost a large automaker a significant amount of money. Fiat Chrysler Automobiles, facing a reputation hit and possible loss of future customers, made the wise but costly decision to patch any vehicles that were vulnerable to Miller and Valasek's exploit. By some estimates, the amount which this critical security update costed FCA in labor hours alone was in excess of $10 million (Cobb 2015). Miller and Valasek have long stated that their shared goal has been to provide their research to the automotive industry and security community "so that we can learn to build more secure vehicles in the future, so that drivers can trust they are safe from a cyber attack" (Miller and Valasek 2015, p.88). Certainly, hitting an automaker's bottom line is an effective way to accomplish this greater good.

6.7 Exercises

What is meant by the term *digital transformation*?
Give an example of the characteristics of digital transformation.
What is meant by the term *information technology*?
Give an example of the characteristics of information technology.
What is meant by the term *cybersecurity*?

Give an example of the characteristics of cybersecurity.
What is meant by the term *application security*?
Give an example of the characteristics of application security.
What is meant by the term *information security*?
Give an example of the characteristics of information security.
What is meant by the term *network security*?
Give an example of the characteristics of network security.
What is meant by the term *security threats*?
Give an example of the characteristics of security threats.
What is meant by the term *countermeasures* with regard to *cybersecurity*?
Give an example of the characteristics of countermeasures with regard to cybersecurity.
What is meant by the term *likelihood of risk*?
Give an example of the characteristics of likelihood of risk.
What is meant by the term *risk management in cybersecurity*?
Give an example of the characteristics of risk management in cybersecurity.
What is meant by the term *security risk*?
Give an example of the characteristics of security risks.
What is meant by the term *vulnerability*?
Give an example of the characteristics of vulnerability.
What is meant by the term *vulnerable space*?
Give an example of the characteristics of a vulnerable space.
What is meant by the term *vulnerable access points*?
Give an example of the characteristics of vulnerable access points.
What is meant by the term *cyber attack*?
Give an example of the characteristics of cyberattacks.
What is meant by the term *anomaly detection*?
Give an example of the characteristics of anomaly detection.
What is meant by the term *denial of service*?
Give an example of the characteristics of denial of service.
What is meant by the term *artificial intelligence*?
Give an example of the characteristics of artificial intelligence.
What is meant by the term *control theory*?
Give an example of the characteristics of control theory.
What is meant by the term *epidemic theory*?
Give an example of the characteristics of epidemic theory.
What is meant by the term *game theory*?
Give an example of the characteristics of game theory.
What is meant by the term *graph theory*?
Give an example of the characteristics of graph theory.
What is meant by the term *probabilistic dependence graph*?
Give an example of the characteristics of a probabilistic dependence graph.
What is meant by the term *logic bomb*?
Give an example of the characteristics of a logic bomb attack.
What is meant by the term *Trojan horse*?

Give an example of the characteristics of Trojan horses.
What is meant by the term *virus*?
Give an example of the characteristics of viruses.
What is meant by the term *worm*?
Give an example of the characteristics of worms.
What is meant by the term *vehicle-to-infrastructure*?
Give an example of the characteristics of vehicle-to-infrastructure.
What is meant by the term *vehicle-to-mobile*?
Give an example of the characteristics of vehicle-to-mobile.
What is meant by the term *vehicle-to-vehicle*?
Give an example of the characteristics of vehicle-to-vehicle.
What is meant by the term *OEM*?
Give an example of the characteristics of OEMs.
What is meant by the term *remote hacking*?
Give an example of the characteristics of remote hacking.
What is meant by the term *attack value chain*?
Give an example of the characteristics of attack value chains.
What is meant by the term *man-in-the-middle attack*?
Give an example of the characteristics of a man-in-the-middle attack.
What is meant by the term *compromised-key attack?*
Give an example of the characteristics of a compromised-key attack.
What is meant by the term*electronic control unit*?
Give an example of the characteristics of an electronic control unit.
What is meant by the term *CAN*?
Give an example of the characteristics of CAN.
What is meant by the term *cyberattack taxonomy*?
Give an example of the characteristics of a cyberattack taxonomy.
What is meant by the term *attack surface*?
Give an example of the characteristics of attack surfaces.
What is meant by the term *onboard diagnostics*?
Give an example of the characteristics of onboard diagnostics.
What is meant by the term *vulnerability scanning*?
Give an example of the characteristics of vulnerability scanning.
What is meant by the term *intrusion detection*?
Give an example of the characteristics of intrusion detection.
What is meant by the term *intrusion prevention*?
Give an example of the characteristics of intrusion prevention.
What is meant by the term *WLAN security*?
Give an example of the characteristics of WLAN security.
What is meant by the term *sensor node security*?
Give an example of the characteristics of sensor node security.
What is meant by the term *WEP authentication*?
Give an example of the characteristics of WEP authentication.
What is meant by the term *platform security*?
Give an example of the characteristics of platform security.

What is meant by the term *cloud computing*?
Give an example of the characteristics of cloud computing.
What is meant by the term *functional safety*?
Give an example of the characteristics of functional safety.
What is meant by the term *mean time between failure*?
Give an example of the characteristics of mean time between failure.
What is meant by the term *mean address spoofing*?
Give an example of the characteristics of an address spoofing.
What is meant by the term *eavesdropping*?
Give an example of the characteristics of eavesdropping.
What is meant by the term *medium access control*?
Give an example of the characteristics of medium access control.
What is meant by the term *false node*?
Give an example of the characteristics of false nodes.
What is meant by the term *platform security*?
Give an example of the characteristics of platform security.
What is meant by the term *insiders*?
Give an example of the characteristics of insiders.
What is meant by the term *outsiders*?
Give an example of the characteristics of outsiders.
What is meant by the term *Byzantine model*?
Give an example of the characteristics of the Byzantine model.
What is meant by the term *mean time between failure*?
Give an example of the characteristics of mean time between failure.
What is meant by the term *SIL*?
Give an example of the characteristics of SIL.
What is meant by the term A*SIL*?
Give an example of the characteristics of ASIL.
What is meant by the term *car hacking*?
Give an example of the characteristics of car hacking.

References and Further Reading

(Akella et al. 2010) Akella, R., Tang, H., McMillin, B.: Analysis of Information Flow Security in Cyber-Physical Systems. In: Internat. Journal of Critical Infrastructure Protection, Vol. 3, pp. 157–173, 2010

(Akyildiz et al. 2002) Akyildiz, I. E., Su, W., Sankkarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey. In: Comput. News, Vol. 16, No. 4, pp 393–402, 2002

(Avancha 2005) Avancha, S.: A Holistic Approach to Secure Sensor Networks. Ph. D. thesis, 2005

(Barika, et al. 2010) Barika, F., Hadjar, K., El-Kadhi, N.: Artificial neural network for mobile IDS solution, In: Security and Management, pp. 271–277, 2010

(Bitter et al. 2010) Bitter, C., Elizondo, D. A., Watson, T.: Application of Artificial Neural Networks and Related Techniques to Intrusion Detection. In: IEEE World Congress on Computational Intelligence, pp. 949–954, IEEE Press 2010.

(Bittersohl and Thoppill 2015) Bittersohl, C., Thoppil, T. G.: Automotive Cyber Security, P3 Inc., 2015

(Brown 1985) Broqn, J.: An Introduction to the Use of Facet Theory. In: Facet Theory, pp. 17–57, Springer Publ. 1985
(Bruton 2014) Bruton, J. A.: Securing CAN Bus Communication: An Analysis of Cryptographic Approaches. Master Thesis National University of Ireland, Galway, 2014
(Butayán and Hubaux 2007) Butayán, L., Hubaux, J.-P.: Security and Cooperation in Wireless Networks. Cambridge University Press, 2007
(CAMP05 2005) CAMP05 Vehicle Safety Communications Consortium. Vehicle Safety Communications Project Task 3 Final Report 2005. http://www.intellidriveusa.org/documents/vehicle-safety.pdf
(CAMP09 2008) CAMP09 Vehicle Safety Communications Consortium. Vehicle Safety Communications – Applications 1st Annual Report, Sept. 2008. http://www.intellidriveusa.org/documents/09042008-vsc-a-report.pdf
(CAMP10 2008) CAMP10 Vehicle Safety Communications Consortium. Cooperative Intersection Collision Avoidance System Limited to Stop Sign and Traffic Signal Violations Midterm Phase I Report, Oct. 2008. http://www.nhtsa.dot.gov/staticfiles/DOT/NHTSA/NRD/Multimedia/PDFs/Crash%20Avoidance/2008/811048.pdf
(Cárdenas et al. 2008) Cárdenas, A., Amin, S., Sastry, S.: Secure Control - Towards Survivable Cyber-Physical Systems. Proceed. 28th IEEE International Conference on Distributed Computing Systems Workshops, pp. 495–500, 2008
(Cárdenas et al. 2011) Cárdenas, A., Amin, S., Lin, Z., Huang, Y., Huan, C., Sastry, S.: Attacks against Process Control Systems: Risk Assessment, Detection, and Response. Proceed. 6th ACM Symposium on Information, Computer and Communications Security, pp. 355–366, 2011
(Cebula and Young 2010) Cebula, J., Young, L. R.: A Taxonomy of Operational Cyber Security Risks. Software Engineering Institute Technical Note CMU/SEI-2010-TN-028, 2010
(Chakrabarti et al. 2007) Chakrabarti, D., Leskovec, J., Faloutsos, C., Madden, S., Guestin, C., Faloutsos, M.: Information Survival Threshold in Sensor and P2P Networks. In: INFOCOMM, IEEE, pp. 1316–1324, 2007
(Chalkias et al. 2009) Chalkias, K., Baldimtsi, F., Hristu-Varsakelis, D., Etephanides, G.: Two Types of Key-Compromise Impersonation Attacks against One-Pass Key Establishment Protocols. In: Communications in Computer and Information Science, Vol. 23, Part 3, pp. 227–238, 2009
(Chatterjee 2012) Chatterjee, P.: The Connected Car as a Platform. In: EDN Nertwork, December 2012
(Checkoway et al. 2011) Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., Kohno, T.: Comprehensive Experimental Analysis of Automotive Attack Surfaces. http://www.autose.org/pubs/cars-usenixsec2011.pdf
(Cichonsky et al. 2012) Cichonsky, P., Millar, T., Grance, T., Scarfone, K.: Computer Security Incident Handling Guide. National Institute of Standards and Technology (NIST) Special Publication 800-61, Revision 2, 2012
(Cobb 2015) Cobb, S.: Cybersecurity and Manufacturers: What the Costly Chrysler Jeep Hack Reveals. http://www.welivesecurity.com/2015/07/29/cybersecurity-manufacturing-chrysler-jeep-hack/
(Currie 2015) Currie, R.: Developments in Car Hacking. SANS Institute 2015. https://www.sans.org/reading-room/whitepapers/ICS/developments-car-hacking-36607
(Daley and Gani 1999) Daley, D. J., Gani, J.: Epidemic Modelling: An Introduction. Cambridge University Press, 1999
(Das et al. 2012) Das, S. K., Kant, K., Zhang, N.: Handbook on Securing Cyber-Physical Critical Infrastructure. Elsevier Publ. 2012
(De Capitani di Vimercati 2007) De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Over Encryption: Management of Access Control Evolution on Outsourced Data. In: Proc. of VLDB, pp. 123–134. 2007
(Denning 1987) Denning, D. E.: An Intrusion Detection Model. In: IEEE Transactions on Software, Vol: SE-13 Issue: 2, pp. 222–232, 1987

(Dilek et al. 2015) Dilek, S., Caku, H., Aydin, M.: Applications of Artificial Intelligence Techniques to Combating Cyber Crimes - A Review. Internat. J. of Artificial Intelligence and Applications (IJAIA), Vol. 6, No. 11, pp. 21–39, 2015
(Dolev 1982) Dolev, D.: The Byzantine Generals Strike Again. Journal of Algorithms, Vol. 3(1), pp.14–30, 1982
(Eisenhauer et al. 2006) Eisenhauer, J., Donelly, P., Ellis, M., O'Brien, M.: Roadmap to Secure Control Systems in the Energy Sector. Energetics Inc. Columbia, MD, 2006
(Eugster et al. 2004) Eugster, P. T., Guerraoui, R., Kermarrec, A., Massouli, L.: From Epidemics to Distributed Computing. In: IEEE Computer, Vol. 37, pp. 60–76, 2004
(Eyal 2007) Eyal, N.: Vehicle Lab – Engine Control Unit, 2007. http://www.vehicle-lab.net/ecu.html
(Falliere et al. 2011) Falliere, N., O'Murchu, L., Chien, E.: W32. Stuxnet Dossier. Symantec Corporation, 2011
(Finke et al. 2015) Finke, T., Schoop, D., Melcher, H.: Extension of Security AUTOSAR architecture to recognition and Countermeasures in terms of relevant attack scenarios Automotive Ethernet. Thesis Work in German; University of Applied Sciences Esslingen, 2015
(Fleury et al. 2009) Fleury, T., Khurana, H., Welch, V.: Towards Taxonomy of Attacks against Energy Control Systems. Proceed. 2nd Annual IFIP Working Group. Internat. Conference on Critical Infrastructure Protection, pp. 71–85, 2009
(Gamage and McMillian 2009) Gamage, T., McMillin, B.: Enforcing Information Flow Properties using Compensating Events. In: Proceed. 42nd Hawaii Internat. Conference on System Sciences, pp. 1–7, 2009
(Goh et al. 2003) Goh, E., Shacham, H., Modadugu, N., Boneh, D.: SiRiUS: Securing Remote Untrusted Storage. In: Proc. of NDSS, pp. 131–145, 2003
(Goodfellow et al. 2016) Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, 2016. www.deeplearningbook.org
(Goodwin 2009) Goodwin, A.: Ford Unveils Open-Source Developer Platform. 2009. http://reviews.cnet.com/8301-13746_7-10385619-48.html, Oct. 2009
(Gordon and Ford 2006) Gordon, S., Ford, R.: On the Definition of Classification of Cybercrime. Journal in Computer Virology, Vol.2, No. 1, pp. 13–20, 2006
(Greenberg 2013) Greenberg, A.: Hackers Reveal Nasty New Car Attacks-With me Behind the Wheel. http://www.forbes.com/sites/andygreenberg/2013/07/24/hackers-reveal-nasty-new-car-attacks-with-mebehind-the-wheel-video/
(Greenberg 2015) Greenberg, A.: Hackers Remotely Kill a Jeep on the Highway-With me in it. http://www.wired.com/2015/07/hackersremotely-kill-jeep-highway/
(Gupta 2016) Gupta, V.: Control of Cyber-Physical Systems: Recent Results and New Challenges, 2016; http://www.ieeecss-oll.org/sites/default/files/final_gupta_acc.pdf
(Guttmann and Roback 1995) Guttman, B., Roback, E. A.: An Introduction to Computer Security: The NIST Handbook. DIANE Publ. 1995
(Hamlen et al. 2006) Hamlen, K., Morrisett, G., Schneider, F.: Computability classes for enforcement mechanisms. In: ACM Transactions on Programming Languages and Systems, Vol. 28, No. 1, pp. 175–205; 2006
(Hansman and Hunt 2005) Huntsman, S., Hunt, R.: A Taxonomy of Network and Computer Attacks. In: Computers and Security, Vol. 24, Issue 1, pp. 31–43, 2005
(Heady et al. 1990) Heady, R., Luger, G., Maccabe, A., Servilla, M.: The Architecture of a Network Level Intrusion Detection System. Technical Report University of New Mexico, Department of Computer Science, 1990
(Housley and Arbaugh 2003) Housley, R., Arbaugh, W.: Security Problems in 802.11-based Networks. In: Commun. ACM Vol. 46, No. 5, pp. 21–34, 2003
(Hubaux et al. 2004) Hubaux, J. P., Chapkun, S., Luo, J., Raya, M.: The Security and Privacy of Smart Vehicles. In: Journal IEEE Security and Privacy, Vol. 2, No. 3, pp. 49–55, 2004
(Intel Security 2015) Intel Security White Paper Automotive Security Best Practice. 2015; http://www.mcafee.com/de/resources/white-papers/wp-automotive-security.pdf

(IXIA 2014) IXIA Securing the Connected Car, Whitepaper 915–3513-01 Rev. A, 2014: www.ixiacom.com

(Jin et al. 2012) Jin, X., Dan, M., Zhang, N., Yu, W., Fu, X., Das, S. K.: Game Theory for Infrastructure Security: The Power of Intent-Based Adversary Models. In: Das, S. K., Kant, K., Zhang, N.: Handbook on Securing Cyber-Physical Critical Infrastructure, pp. 31–53. Morgan Kaufmann Publ., 2012

(Johnson 2010) Johnson, T.: Fault-Tolerant Distributed Cyber-Physical Systems: Two Case Studies. Master Thesis University of Illinois, ECE Dept., 2010

(Johnson 2016) Johnson, M.: Cyber Crime, Security and Data Intelligence. Routledge Publ. 2016

(Kallahalla et al. 2003) Kallahalla, M., Riedel, E., Waminadham, R., Wang, Q., Fu, K.: Scalable Secure File Sharing on Untrusted Storage. In: Proc. of 2nd USENIX Conference of File and Storage Technologies, pp. 29–42. 2003

(Kao and Marculescu 2006) Kao, J. C., Marculescu, R.: Eavesdropping Minimization via Transmission Power Control in Ad-Hoc Wireless Networks. In: 3rd Annual IEEE Communications Society on Sensor and Ad-Hoc Communications and Networks, pp. 707–714, 2006

(Karim and Proha 2014) Karim, E., Proha, V. V.: Cyber-Physical Systems Security. In: Applied Cyber-Physical Systems, pp. 75–84. Eds.: Shuh, S. S., Tanik, U., J., Carbone, J. N., Rogglu, A.; Springer Publ., 2014

(Kephart and White 1993) Kephardt, J. O., White, S. R.: Measuring and Modeling Computer Virus Prevalence. In: Proceed. IEEE Symposium on Security and Privacy, pp. 2–15, 1993

(Kermack and McKendrick 1927) Kermach, W. O., McKendrick, A.: A Contribution to the Mathematical Theory of Epidemics. Proceed. Royal Society of London, Vol. A, No. 1, pp. 700–721, 1927

(Kjaerland 2005) Kjaerland, M.: A Taxonomy and Comparison of Computer Security Incidents for the Commercial and Government Sectors. In: Computers and Security, Vol. 25, pp. 522–538, 2005.

(Koscher et al. 2010) Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S.: Experimental Security Analysis of a Modern Automobile. In: IEEE Symposium on Security and Privacy, pp. 448–461, 2010

(Kumar and Spafford 1994) Kumar, S., Spafford, E. H.: An Application of Pattern Matching in Intrusion Detection. Computer Science Technical Reports, Paper 1116, Purdue University, 1994

(Landrum et al. 2014) Landrum, R., Pace, S., Hu, F.: Cyber-Physical Systems Security–Smart Grid Example, pp. 135–154. In: Cyber-Physical Systems. Ed.: F. Hu. CRC Press 2014

(Lamport 1997) Lamport, L.: Proving the Correctness of Multiprocessing Programs. In: IEEE Transactions on Software Engineering, Vol. 3(2), pp. 125–143, 1997

(Lamport 1998) Lamport, L.: Proving Possibility Properties. In: Theoretical Computer Science, Vol. 206(1–2), pp. 341–352, 1998

(Lamport 2005) Lamport L.: Real-Time Model Checking is Really Simple. Proceed. 13th Advanced Research Working Conference on Correct Hardware Design and Verification Methods, pp. 162–175, 2005

(Landram et al. 2014) Landram, R., Pace, S., Hu, F.: Cyber-Physical System Security - Smart Grid Example. In: F. Hu: Cyber-Physical Systems - Integrated Computing and Engineering Design. pp. 145–154, CRC Press 2014

(Lin and Sangiovanni-Vincentelli 2012) Lin, C. W., Sangiovanni-Vincentelli, A.: Cyber-Security for the Controller Area Network (CAN) Communication Protocol. In: IEEE Proceed. Internat. Conference on Cyber Security, pp. 1–7, 2012

(Lin et al. 2013) Lin, C. W., Zhu, Q., Phung, C., Sangiovanni-Vincentelli, A.: Security-aware mapping for CAN-based real-time distributed automotive systems. In: IEEE Proceed. Internat. Conference on Cyber Security, pp. 115–121, 2013

(Lough 2001) Lough, G. L.: A Taxonomy of Computer Attacks with Applications to Wireless Networks. Dissertation submitted to the Faculty of the Virginia Polytechnic Institute, 2001

(Lunt et al. 1992) Lunt, T. F., Tamaru, A., Gilham, F., Jagannathan, R., Neumann, P. G., Javitz, H. S., Valdes, A., Garvey, T. D.: A Real-Time Intrusion Detection Expert System (IDES) –

Final technical Report, SRI Computer Science Laboratory, SRI International, Menlo Park, CA, 1992

(Luo et al. 2010) Luo, Y., Szidarovsky, F., Al-Nashif, Y., Hariti, S.: Game Theory Based Network Security. In. Journal of Information Security, pp. 41–44, 2010

(Miller and Valasek 2014) Miller C., Valasek C.: A Survey of Remote Automotive Attack Surfaces. IOActive 2014. Available from: https://www.ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf

(Miller and Valasek 2015) Miller, C., Valasek, C.: Remote Exploitation of an Unaltered Passenger Vehicle. http://illmatics.com/Remote%20Car%20Hacking.pdf

(Mo and Sinopoli 2009) Mo, Y., Sinopoli, B.: Secure Control against Replay Attacks. Proceed. 47th Conf. on Communication, Control, and Computing, pp. 911–918, 2009

(Mollman 2009) Mollmann S.: From Cars to TVs, Apps are Spreading to the Real World. http://edition.cnn.com/2009/TECH/10/08/apps.realworld/

(Möller 2016) Möller, D. P. F.: Guide to Computing Fundamentals in Cyber-Physical Systems – Concepts, Design Methods, and Applications, Springer Publ., 2016

(ni-com 2009) ECU Designing and Testing Using National Instruments Products. White Paper, National Instruments 2009

(Nurse et al. 2014) Nurse, J. R. C., Buckley, O., Legg, P. A., Goldsmith, M., Cresse, S., Wright, G. R., Whitey, M.: Understanding Insider Threat: A Framework for Characterizing Attacks. IEEE Security and Privacy Workshops, pp. 214–222, IEEE 2014

(Patel et al. 2010) Patel, A., Qassim, Q., Shukor, Z., Nogueira, J., Junior, J., Wills, C.: Autonomic Agent-Based Self-Managed Intrusion Detection and Prevention System, In: Proceed. South African Information Security Multi-Conference, pp. 223–234, 2010

(Pathan et al. 2006) Pathan, Al-S. K., Lee, H.-W., Hong, C. S.: Security in Wireless Sensor Networks: Issues and Challenges. In: Proceed. Internat. Confer. Advanced Technology, pp. 1043–1048, 2006

(Payteck 2003) How PayTeck Works. www.payteck.cc/aboutpayteck.html

(Pfleeger et al. 2015) Pfleeger, C. P., Pfleeger, S. L., Margulies, J.: Security in Computing. Prentice Hall 2015

(Pelechrinis et al. 2011) Pelechrinis, K., Iliofotou, M., Krishnanurthy, S. V.: Denial of Service Attacks in Wireless Networks: The Case of Jammers. In: IEEE Communications Surveys and Tutorial, Vol. 13, No. 2, pp. 245–257, 2011

(Poulsen 2010) Poulsen, K.: Hacker disables more than 100 cars remotely. Wired online. March 17th 2010. Available from: www.wired.com/threatlevel/2010/03/hacker-bricks-cars

(PR Newswire 2015) https://www.prnewswire.com/news-releases/nearly-80-percent-of-consumers-think-vehicle-hacking-will-be-frequent-problem-in-near-future-according-to-new-kelley-blue-book-survey-300121740.html

(Salahuddin and Al-Fuqaha 2013) Salahuddin M. A. B., Al-Fuqaha, A.: AGORA: A Versatile Framework for the Development of Intelligent Transportation System Applications. In: Wireless Sensor and Mobile Ad-Hoc Networks: Vehicular and Space Applications, pp. 163–184, Eds.: B. Benhaddou, A. Al-Fuqaha, Springer Publ. 2013

(Saleh and Khatib 2005). Saleh, M., Khatib, I. A.: Throughput Analysis of WEP Security in Ad Hoc Sensor Networks. In: Proc. 2nd International Conference on Innovations in Information Technology, 2005

(Saltzman and Sharabani 2009) Saltzman, R., Sharabani, A.: Active Man in the Middle Attacks – A Security Advisory. Whitepaper IBM Rational Application Security Group. IBM Corporation 2009

(Satyanarayanan et al. 2009) Satyanarayan, M., Bahl, P., Caceres, R., Davies, N.: The Case for VM-based Cloudlets in Mobile Computing. IEEE Pervasive Compt. Vol. 8 No. 4, 14–23, 2009

(Sastry et al. 1994) Sastry, P. S., Phansalpar, V. V., Thathachar, M. A. L: Decentralized Learning of Nash Equilibria in Multi-Person Stochastic Games with Incomplete Information. In: IEEE Transct. On Systems, Man, and Cybernetics, Vol. 24, No. 5, pp. 769–777, 1994

(Scarfone and Mell 2007) Scarfone K., Mell, P.: Guide to Intrusion Detection and Prevention Systems. National Institute of Standards and Technology (NIST) Special Publication 800–94, 2007
(Shieh and Gligor 1991) Shiva, S. W., Gligor, V. D.: A Pattern Oriented Intrusion Model and its Applications. In: Proceed. IEEE Computer Society Symposium on Research in Security and Privacy, pp. 327–342, 1991
(Shimeall and Spring 2014) Shimeall, T., Spring, J.: Introduction to Information Security: A Strategic-Based Approach. Elsevier Publ. 2014
(Shiva et al. 2010) Shiva, S., Roy, S., Dasgupta, D.: Game Theory for Cyber Security. In: CSIIRW Conf. Proceed., ACM Press 2010
(Simmons et al. 2014) Simmons, C., Ellis, C., Shiva, S., Dasgupta, D., Wu, Q.: AVOIDIT : A cyberattack Taxonomy. In: 9th Annual Symposium on Information Assurance (ASIA), pp. 14-1-14-, 2014
(Smaha 1988) Smaha S. E.: Haytack: An Intrusion Detection System. In: Proceed. 4th Aerospace Computer Security Applications Conference, pp. 37–44, 1988
(Tang and McMillin 2008) Tang, H., McMillian, B.: Security Property Violation in CPS through Timing. In: Proceed. 28th Internat. Conference on Distributed Computing Systems Workshops, pp. 519–524, 2008
(Valasek and Miller 2014) Valasek, C., Miller, C.: A Survey of Remote Automotive Attack Surfaces. Technical White Paper, IOActive Inc., 2014
(VTTI 2007) VTTI - Virginia Tech Transportation Institute. Intersection Collision Avoidance - Violation Task 5 Final Report, 2007. http://www.intellidriveusa.org/documents/final-report-04-2007.pdf
(Wang et al. 2010) Wang, E. K., Ye, Y., Xu, X., Yiu, S. M., Hui, L. C. K., Chow, K. P.: Security Issues and Challenges for Cyber Physical Systems. IEEE/ACM Conference on Green Computing and Communications and IEEE/ACM Intern. Conference on Cyber, Physical and Social Computing, pp.733–738, IEEE Publ., 2010
(Xiao 2006) Xiao, Y.: Security in Sensor Networks. Auerbach Publ., 2006
(Xiao et al. 2008) Xiao, K., Ren, S., Kwiat, K.: Retrofitting Cyber-Physical Systems for Survivability through External Coordination. In: Proceed. 41st Internat. Conference on Systems Science, pp. 454–466, 2008
(Yuzhe et al. 2013) Yuzhe, L., Ling, S., Peng, D., Quecedo, E.: Jamming Attack on Cyber-Physical Systems : A Game Theoretic Approach. In: IEEE 3rd Annual Conference on Cyber Technology in Automation, pp. 252–257, 2013
(Zeltser 2015) Zeltser, L.: Antivirus Software uses Several Different Virus Detection Techniques. TechTarget Network, 2015
(Zimmer et al. 2010) Zimmer, C., Bhat, B., Mueller, F., Mohan, S.: Time-Based Intrusion Detection in Cyber-Physical Systems. In: Proceed. 1st ACM/IEEE International Conference on Cyber-Physical Systems, pp. 109–118, 2010

Links

(URL1 2016) www.dhs.gov/science-and-technology/cyber-security-division
(URL2 2016) https://autoalliance.org/connected-cars/cybersecurity/
(URL3 2016) http://www.icao.int/APAC/Documents/edocs/cns/mlat_concept.pdf
(URL4 2016) https://www.autosar.org/fileadmin/user_upload/standards/classic/3-0/AUTOSAR_TechnicalOverview.pdf
(URL5 2016) https://en.wikipedia.org/wiki/Secure_Neighbor_Discovery
(URL6 2016) https://www.genivi.org/challenges
(URL7 2016) http://searchcontentmanagement.techtarget.com/definition/taxonomy
(URL8 2016) https://ldra.com/automotive/
(URL9 2016) http://www.openvas.org/about.html

(URL10 2016) https://www.automotiveisac.com/best-practices/
(URL11 2016) https://en.wikipedia.org/wiki/Functional_safety
(URL12 2016) http://www.exida.com/Resources/Term/Automotive-Safety-Integrity-Level-ASIL
(URL13 2016) https://en.wikipedia.org/wiki/Failure_mode_and_effects_analysis
(URL1 2017) https://www.techopedia.com/definition/24841/denial-of-service-attack-dos
(URL1 2018) https://www.plattform-i40.de/I40/Navigation/DE/Industrie40/Handlungsfelder/Sicherheit/sicherheit.html (in German)