Security and Trustworthiness Threats to Composite Services: Taxonomy, Countermeasures, and Research Directions

Per Håkon Meland[1], Muhammad Asim[2], Dhouha Ayed[3], Fabiano Dalpiaz[4], Edith Félix[3], Paolo Giorgini[5], Susana Gonzáles[6], Brett Lempereur[2], John Ronan[7]

[1] SINTEF ICT, per.h.meland@sintef.no
[2] Liverpool John Moores University, {m.asim,b.lempereur}@ljmu.ac.uk
[3] Thales Services, {dhouha.yahed,edith.felix}@thalesgroup.com
[4] Utrecht University, f.dalpiaz@uu.nl
[5] University of Trento, paolo.giorgini@unitn.it
[6] ATOS, susana.gzarzosa@atos.net
[7] Waterford Institute of Technology, jronan@tssg.org

Abstract. This chapter studies not only how traditional threats may affect composite services, but also some of the new challenges that arise from the emerging Future Internet. For instance, while atomic services may, in isolation, comply with privacy requirements, a composition of the same services could lead to violations due to the combined information they manipulate. Furthermore, with volatile services and evolving laws and regulations, a composite service that seemed secure enough at deployment time, may find itself unacceptably compromised some time later. Our main contributions are a taxonomy of threats for composite services in the Future Internet, which organises thirty-two threats within seven categories, and a corresponding taxonomy of thirty-three countermeasures. These results have been devised from analysing service scenarios and their possible abuse with participants from seventeen organisations from industry and academia.

Keywords: Threats, taxonomy, countermeasures, service composition, security, trustworthiness.

1 Introduction

The capability to effectively cope with unexpected changes and threats is desirable for any system. Systems residing on the Internet are no exceptions, as the Internet is a volatile and vulnerable environment that poses difficult challenges for researchers and systems engineers. Representatives from European industry and academia [17] have already stated in their Future Internet vision that *a primary research direction is to make the Internet—and the systems deployed over it—more secure, dependable, reliable, and flexible.*

This chapter investigates both how traditional threats will affect composite services, and some of the new challenges that shall be accounted for in the emerging Future Internet. For instance, while atomic services may in isolation comply with privacy requirements, a composition of the same services could lead to violations due to the use and manipulation of combined information. Furthermore, with volatile services and evolving laws and regulations, a composite service that seemed secure enough at deployment time may become non-compliant.

Our main contribution is a taxonomy of threats—organised within seven categories—and corresponding countermeasures for composite services in the Future Internet. These results have been devised from analysing service scenarios and their possible abuse with participants from seventeen organisations from industry and academia.

This chapter is organised as follows. Section 2 describes the research method that we followed. Section 3 presents our taxonomy of threats and Section 4 suggests possible countermeasures to these threats. Section 5 outlines research directions to tackle the threats as well as the implementation of countermeasures. Section 6 gives and overview of related work and, finally, Section 7 concludes the chapter.

2 Research method

In order to study the threats related to composite services, we have employed a scenario-driven method to identify the most relevant types of threats based on both the knowledge of the present situation as well as what different stakeholders envision concerning the near future.

As service composition is still an emerging field, one cannot simply look at incidents in the past to determine what will be the greatest challenges. The scenario development process involved the seventeen organisations in the Aniketos project, with their different expertise and domain knowledge related to service technology. Together, these organisations cover private and public service providers, Cloud providers, security companies, researchers (institutes and universities) on secure service engineering and end-users.

The organisations were encouraged to focus on their expertise domains during scenario description. However, in order to have a comprehensive catalog of threats, we also allowed scenarios related to other domains, even beyond the project case studies. The scenario development was done iteratively, starting off with rough sketches of the usage and behaviour of service environments. A chosen moderator gathered the scenario descriptions and performed an initial review of their relevance. This was followed by a refinement process, where each scenario was updated by the scenario creators in collaboration with the moderator. Afterwards, there were several iterations where the group as a whole determined necessary steps to remove ambiguousness and gaps.

The rough scenarios consisted of short text with focus on the normal situations and behaviour. They were then refined into a more structured template

consisting of a summary, workflow description, workflow deviations, stakeholders involved, and expected outcome. For each scenario, we tried to identify how the service environment could be exploited for malicious intents through misuse/abuse case scenario descriptions. They were similarly structured with a short description, stakeholders (the attackers), outcome, assets involved, and possible countermeasures/mitigations. Figure 1 illustrates the overall process.

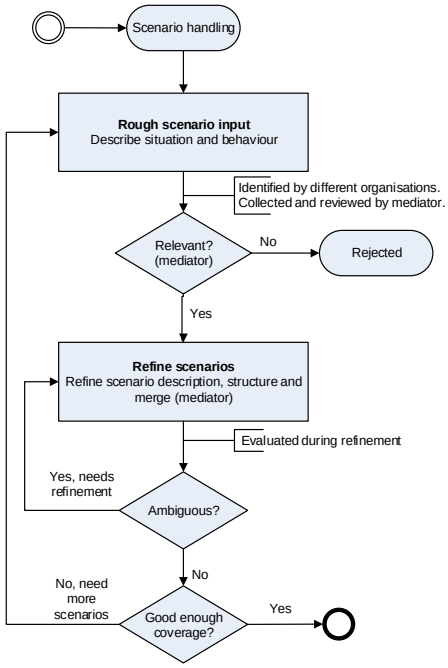

Fig. 1. Scenario elicitation process

The first round of scenario creation resulted in more than fifty scenarios for normal situations, and twenty misuse cases identifying threat events and threat agents. These scenarios have been our main information source for defining threats and countermeasures. A second round of scenario elicitation was performed two years later. Eleven additional scenarios were identified then, and we have been updating the classification itself and the threat description over a period of four years as technology and service uptake have progressed.

We chose to focus on and classify threats that are critical for composite services; thus, the taxonomy should not be regarded as a complete overview for software systems in general. To sort out what is already established threats, we have studied the dictionary of Common Weakness Enumeration (CWE) [2], the dictionary of Common Vulnerabilities and Exposures (CVE), and results from various research project such as Forward [3] (extensive list in section 6).

3 A taxonomy of threats for composite services

Trustworthiness and security in composite services have the same foundations as in traditional information systems. They are grounded by the security needs which shall roughly be aligned with classical needs as addressed by the information security field, for example confidentiality, availability, integrity, trustworthiness, privacy, access control, non-repudiation as defined in [16].

However, composite services present specific vulnerabilities and threats that do not affect traditional information systems. A threat is defined as a potential for violation of security, which exists when there is a potential for accidentally triggering or intentionally exploiting a specific vulnerability. Vulnerabilities are instead security weaknesses or flaws that make a system susceptible to an attack, whereas attacks consist in the exploitation of such vulnerabilities, being the actual materialization of threats [18,19]. Countermeasures are defensive security mechanisms used for mitigating system vulnerabilities. If a vulnerability is detected within a service, this would typically reduce the trustworthiness of the service until it has been repaired or mitigated.

Taking into account the special characteristics of dynamic composite services, we have defined a set of categories and classes of threats following the method presented in Section 2. Categories are more abstract than classes, and help with organisation. Our categories share similarities with the STRIDE[8] categories [10], which are widely used for modelling threats to traditional software systems, but we have specialized them for our domain. A threat class can belong to more than one category, and represents more specific unwanted events. Note that the threat classes have different levels of abstractions as well, and can in some cases partly overlap/subsume each other. The reason for this is that they have been collected from different industry domains. In our work, we had to make a balance between making the threat classes generic enough for wider use, and preserving them in their original form in terms of name and level of detail. For each threat class we have also indicated a threat impact value within the range of low, medium, high. These values must be considered as an indicative starting point, as they are based on how they would affect the scenarios they originate from. For further reuse, these values may need to be re-assessed based on the system under consideration. Section 3.1 explains our categories, while Section 3.2 lists and explains the threat classes. Table 1 summarizes the relationship between categories and threats, along with their impact value.

3.1 Threat categories

[**TC-1**] **Incompatibility:** Service composition can be a highly complicated task, and such complexity tends to cause security issues concerning incompatibility. Functionality should not be the only criteria while considering composition. Other non-functional constraints such as efficiency, redundancy, resource, and

[8] Spoofing, Tampering, Repudiation, Information disclosure, Denial of service and Elevation of privileges

synchronisation issues may all result in a composition failure. The interface of the component services have to be secure and compatible with each other. A composite service might seem to work correctly from a functional point of view, but there may be violations of security requirements. Even though a composite service only consists of components that are considered to be safe and secure individually, their combination might increase overall vulnerability to threats. The incompatibility category is related to *information disclosure*. If we consider for example, that a travel assistance service provider is committed to ensure customer information privacy, a *point of interest* (POI) service provider that uses data related to user preferences, and sells the data to advertisement companies, should not be selected in the composition.

If one service component is insecure this may compromise the overall security of the composite service. It is unlikely that each component will be fully secure, but the objective is to make them secure enough for their purpose.

[**TC-2**] **Constraints:** The services within a composition can be geographically distributed and composed, in turn of other services with different capabilities and constraints, such as security policies, laws or technical and conceptual restrictions. All these new features must be considered in order to avoid the appearance of incompatibilities between the integrated services. Thus, it is necessary to be conscious of the component services to assure a reliable result.

It is possible that each component service has its own security policy. Incompatibility policies may result in a security breach or may lead to different vulnerabilities being exposed. For example, the composite services of the travel assistance service can be deployed in several countries that have incompatible confidentiality laws which can cause an incompatibilities between confidentiality policies of the composition. The constraints category is related to *information disclosure*.

[**TC-3**] **Unaccountability:** Unaccountability is related to *repudiation*. It should be possible to hold service providers of composite services and service components accountable on how data is managed, used and transferred. Responsible parties should be defined in contracts, but this can be difficult in a complex and dynamic environment. With composite services, information exchanged between different services is typically maintained in the form of logged data. This logged information can be used for accountability and chain-of-evidence. Also, the logged data may contain sensitive information (such as user's bank account details), and therefore its integrity and confidentiality have to be protected.

[**TC-4**] **Malicious activity:** Composite services are not spared from malicious activities or *tampering*. Attacks are launched with varying motivations. Examples include financial gain, competitive advantage, and damaging reputation. An attacker can, for instance, first gain access to a single service component before it compromises the overall composition. The component service can be either maliciously coded (e.g., by the service developer) or can offer a vulnera-

bility that the attacker can exploit. Malicious activities always have an intention of damaging the composite service or related assets, and can be performed by insiders or external agents.

[**TC-5**] **Overtrust:** By building trust relationships and establishing trustworthiness, service providers and organisations will improve business value and consumer confidence in the service oriented environments. In a composed services context, a number of trust-related threats arise where several individual services are put together in a composed service. This makes it difficult to have complete control over composite services and thus predict their behavior, and eventually their trustworthiness. For example, the trust level of service components related to the travel assistance service can change over time due to several reasons, such as decreasing reputation, and this compromises the trust level of the whole composition. Overtrust is somewhat related to *elevation of privilege*, *information disclosure*, and *spoofing*.

[**TC-6**] **Usability:** A bad user interface may result in user frustration that can lead them to make errors or compromise their own data. Sometimes it may occur that user interaction with the interfaces or tools in a composite service can increase the likelihood of data being compromised. For example, a lack of user notification or, indeed a large volume of unnecessary notifications could frustrate the end user to such an extent that they inadvertently make bad decisions, compromising their own data. Consequently, it is necessary to have a friendly and easy-to-use user interface. Usability relates with *information disclosure*.

[**TC-7**] **Unavailability**: Unavailability is closely related to *denial of service*, and is especially critical for composite services as the unavailability of any service components can easily make the composite service useless. Unavailability as such is typically a consequence or results of other threats, as a provider may be forced to take down the service if it is not trustworthy enough.

3.2 Threat classes

1. **Incompatible laws:** When services are geographically distributed, legal incompatibilities may arise and pose a security threat (e.g., an adequate level of data confidentiality is not ensured by law in all involved countries). A composite service is perceived as a unique entity by its users. Such a threat occurs because users are typically unaware of the identity of individual providers, of their geographic distribution, and of the laws that apply in the countries where the service component resides. In some business areas, regulations and laws might forbid transferring sensitive data, and may require the consent of the data owner or may result in undesirable legal liabilities for service providers. For example; to exchange a confidential electronic document with the company's vice-president, who is currently on a business trip in country B, the financial manager (who is in country A) assembles a composition based on a secure service S_1 that provides Microsoft Word to Adobe PDF

conversion (to make sure the rendering of the document is preserved regardless of the specific document reader/editor), and a file sharing service S_2 used to share the PDF document with the vice-president. S_1 is deployed in country A, S_2 in country B. The law of country B does not ensure data confidentiality over the Internet, as service providers are obliged to introduce lawful intercept facilities. This generates a confidentiality concern for the composite service. In most countries data confidentiality norms exist. The risk could be low/medium if all parties are within EU. However, the risk could be high if parties involved are more globally (e.g., US and EU).
• *Threat Category: TC-1, TC-2*
• *Impact: High - Incompatible laws may put data confidentiality at risk.*

2. **Incompatible access control models:** Access control of a composition is dependent on the access control capabilities of the individual services. Where different component services use different access control models, the result could be a violation of any of the models. As a simple example, consider a travel assistance service composition where the POI service applies Bell-LaPadula ("no read up, no write down" for confidentiality), while the the route service applies Biba ("no read down, no write up" for integrity), is liable to result in a confused system with both models partially implemented.
• *Threat Category: TC-1, TC-2*
• *Impact: High - The impact of such a attack can be high as confidential and private data may be leaked to unauthorized users or, potentially, attackers. This could be, in fact, worse than having no access control as the users would be completely unaware of the issue and think they are operating in a secure environment.*
3. **Privacy violation via composition:** When some services are composed together, it is possible that although every one of them has its own security policy, the interaction between them or the data shared in the composition can lead to vulnerabilities and privacy violation. In isolation, none of the services in the composition is a threat to privacy; however, when in a composition, privacy is endangered. For example, an organisation relies on services to let employees collect needed data for their job. The administration uses service "Tax" to retrieve the tax number of an employee, given her name and surname, and birth date. The statistics department uses service "Real Properties" to gather anonymous data about real properties of employees. If employees of the statistics department gain access to the "Tax" service and compose it with "Real Properties", they violate the privacy of employees, for they can associate real properties to specific employees.
• *Threat Category TC-1, TC-2, TC-3*
• *Impact: High - Privacy data confidentiality and integrity are very sensitive security issues, and pose as potential show-stopper for compositions.*
4. **Exploitable interaction:** An important characteristic in dynamic service re-composition is the increased, and potentially unplanned, interactions between services. Such interactions are themselves a potential source of vulnerabilities and threats. Problems often arise from existing vulnerabilities. These might exist in individual services, but can be exacerbated or exploited

Threat Categories	Threats Classes	Threat Impact
TC-1: Incompatibility	Incompatible laws	High
	Incompatible access control models	High
	Degraded policy negotiation	High
	Privacy violation via composition	High
	Exploitable interaction	High
	Unwanted recomposition and reconfiguration	High
	Synchronisation threats	High
	Degraded security interface	High
	Insecure interfaces and API's	High
TC-2: Constrains	Incompatible laws	High
	Incompatible access control models	High
	Degrade policy negotiation	High
	Privacy violation via composition	High
	Security guidelines compromised	Medium
	Dissolved redundancy	High
TC-3: Unaccountability	Extracting information from logs	High
	Information and accountability lost	Medium
	Insecure interfaces and API's	High
	Privacy violation via composition	High
	Security guidelines compromised	Medium
	Malicious service provider	High
	Lack of trust between providers	Low
TC-4: Malicious activity	Insufficient automated security evaluation	High
	DDoS attack occurs on service composition	High
	Malicious service provider	High
	Failure to sanitize special element	High
	Embedded malicious code	High
	Protection mechanism failure	High
	Insecure interfaces and API's	High
	Exploitable interaction	High
	Degrade policy negotiation	High
	Extracting information from logs	High
	Manipulation of trust properties	High
TC-5: Overtrust	Manipulation of trust properties	High
	Untrusted outsourcing/delegation	High
	False perception of trust for end user	High
	Reliance on untrusted inputs in a security decision	High
	Inclusion of functionality from untrusted control sphere	High
	Degraded security interface	High
	Degrade policy negotiation	High
	Failure to sanitize special element	High
	Embedded malicious code	High
	Trustworthiness level variability	Medium
TC-6: Usability	Missing end user notification	Medium
	End user gets annoyed by confirmations	Medium
	Lack of usability in secure composition	Medium
	False perception of trust for end use	High
TC-7: Unavailability	Lack of trust between providers	Low
	DDoS attack occurs on service composition	High
	Corrupt load-balancing	Medium
	Recomposition corrupts response time	Medium
	Synchronisation threats	High
	Cascade failure	High

Table 1: The taxonomy of threats to composite services

through dynamic interactions across multiple services. Data validation vulnerabilities are a well-understood and widely-exploited type of vulnerability present in a large number of existing systems. The class encompasses any security threat arising from a failure to validate the syntactic or semantic integrity of data passed between services before the data is used.

• *Threat Category TC-1, TC-4*

• *Impact: High - Confidential and private data may be leaked to unauthorized users or, potentially, attackers.*

5. **Degraded security interface:** Service compositions might be long-lived. However, not all services are invoked together. Some are invoked after previous providers deliver the service. During this time, security service interfaces might change, and this could be a threat for the service composition. For example, consider a service composition that determines the salary of a company's employees. Among the various services, there are two subsequent services: "Analyse timesheets" determines the amount of work, while "Compute gross salary" takes the timesheet data and determines the gross salary. In such composition, the provider of "Compute gross salary" commits to confidentiality and not to further delegate the task. However, service "Analyse timesheets" takes time, for human verification is needed. During this time, the service "Compute gross salary" changed its interface, which does not guarantee non-delegation anymore.

   • *Threat Category TC-1, TC-5*

   • *Impact: High - Changes in the security interfaces might affect the effectiveness of the composition and may not meet the security needs of the user.*

6. **Unwanted recomposition and reconfiguration:** A system adaptation may involve replacing existing services with new ones or re-structuring the services. The resulting composition may introduce some functionality that might not be desirable for the user or the new functionality may not support the existing compositions. This may leads to a number of problems, for instance incompatible compositions which could prevent the correct delivery of composite service; compromise on security requirements and degrade the efficiency of the system.

   • *Threat Category TC-1*

   • *Impact: High - This may leads to a number of problems, i.e., incompatible compositions which could prevent the correct delivery of composite service; compromise on security requirements and degrade system efficiency.*

7. **Insecure interfaces and APIs:** Service providers typically expose a set of software interfaces or APIs that service consumers use to manage and interact with their services. Reliance on a weak set of interfaces can expose an organisation to a variety of security issues related to confidentiality, availability, and password integrity. For example, anonymous access or reusable passwords, clear-text authentication or transmission of content, inflexible access controls or improper authorizations, limited monitoring and logging capabilities, unknown service or API dependencies. Consider a small company that uses a cloud service for daily business management such as online

sale and order management. An insecure interface is exploited by attackers causing financial losses and damage of company's reputation.

• *Threat Category TC-1, TC-3, TC-4*

• *Impact: High - Using services with insecure interfaces and APIs may result in an incompatible composition thereby introducing a threat to the overall security of the composite service.*

8. **Degraded policy negotiation:** Different services may have different policies and often multiple policies cannot be reconciled. This leads to negotiation between service providers. A malicious service provider might use this opportunity to try to affect the security policies of a service to make them weaker in order to attack the service at a later time. A weaker security policy can make a system vulnerable to various attacks.

   • *Threat Category: TC-1, TC-2, TC-4, TC-5*

   • *Impact: High - A weaker security policy can make a system vulnerable to various attacks.*

9. **Security guidelines compromised:** The process of matching security requirements (security guidelines defined by a security specialist) with security capabilities of the services can be a notoriously complex and technical process. In general, developers concentrate more on the functional aspects of a system and may not have extensive experience dealing with security considerations. In some cases, it may be impossible to fulfil all of the required security requirements, such as where security ease-of-use must be balanced against security restrictions. If a developer is unable to create a system that fulfils the requirements, problems are likely to arise. Such problems could take the form of inadequate security, or of failure to deploy a service. The source of the threat comes from lack of security expertise or intractable security requirements.

   • *Threat Category TC-2, TC-3*

   • *Impact:* Medium - *This could lead to all sort of security issues.*

10. **Dissolved redundancy:** Service compositions often involve redundant provision of a certain service. Sometimes, service providers further delegate service provision to third-party providers. If they delegate the service to the same third-party, then the redundancy principle is violated. For example, an air traffic controller needs accurate weather forecasts. According to the flight regulations, he assembles a service composition that includes two providers for rain/snow real-time data. However, both providers outsource the provision to the same third-party. This way, redundancy is not guaranteed any more and the redundancy policy has been violated.

    • *Threat Category TC-2*

    • *Impact: High - Redundant provision of a service is mandatory for critical tasks. When redundancy dissolves, the critical task is at risk (its failure is more likely).*

11. **Information and accountability lost:** In a decentralized system each end point is responsible for collecting and storing information usage events (logs) that may be relevant to current or future assessment of accountability

to some sets of rules/policies. These logs become the major source of assessing policy accountability either in real time or in the future when such an assessment is needed. Therefore, it is important to securely maintain these logs in the system. For example, Alice tries to sign up for a subscription to the newspaper from a foreign country, making use of a SoA comprised of a series of services. For delivery reasons it is not possible to send the newspaper to that country, so one of the services cancels the order. However this is a rare event and the service does not pass the information back to other services with which it is composed. In fact one service sends an email to Alice saying that her subscription was successful. The newspaper has no record of Alice's details.

- *Threat Category TC-3*
- *Impact: Medium – Information and accountability lost may damage the company reputation.*

12. **Extracting information from logs:** Logging information is an essential part of maintaining composite services. These logs capture an extreme amount of data, including sensitive information (e.g., personal information, authentication data, bank details) that must be protected. By adequately securing the logged information, the risk of releasing confidential information to untrusted parties from both inside and outside the organisation can be reduced.
    - *Threat Category TC-3, TC-4*
    - *Impact: High - Confidential and private data may be leaked to unauthorized users or attackers.*
13. **Malicious service provider:** A malicious service provider could ask for unnecessary private or confidential information and store all the gained data in order to assemble and sell a detailed costumer profile. Consider a use case of "travel reservation". In the use case, a user would like to reserve a complete travel package from a composition of loosely-coupled web services. First, the user finds a travel agent service on the web and provides the travel agent with destination and preferred dates. Based on the customer's requirements, the travel agent searches and contracts many airline and hotel services, in order to obtain information on the flights and hotel rooms. The travel agent service then assembles a list of travel alternatives and presents them to the user. The user makes his/her choices and provides the travel agent service with personal information for reservation. The travel agent service then asks for the credit card details and confirms the reservation. During the travel booking process, personal data such as name, date of birth, address, phone no and credit card number are exchanged. However a question arises how to ensure that the requested user's personal data is only used for the stated purpose. A user giving personal identifiable information to an organisation may result in the data being used in ways the user never intended. For example, the credit card details could be passed on to persons intending to commit fraud.
    - *Threat Category TC-3, TC-4*

• *Impact: High - Compromise privacy, confidentiality, integrity and availability depending on the type of information provided.*

14. **Failure to sanitize special element:** Composite services are often involved in receiving inputs from its users. However, a user could inject keystrokes or even code in order to cause an adverse effect on the service behaviour and its integrity. In composite services, a service with such a weakness may put the integrity of the whole composition at risk. Special elements are often important in weaknesses that can be exploited by injection attacks. Therefore, user-controlled input should be properly filtered and intercepted for special elements.
    • *Threat Category TC-4, TC-5*
    • *Impact: High - In composite services, a service with such a weakness may put the integrity of the whole composition at risk. Further, it can be exploited by service injection attacks.*
15. **Embedded malicious code:** A dishonest service developer could insert malicious code within a service to subvert its security. A simple example could be: insert malicious code in a service to send credit card details or any other sensitive information to a particular email address. The malicious code is normally inserted during service implementation. As service developers are often considered trusted, this threat needs consideration.
    • *Threat Category TC-4, TC-5*
    • *Impact: High - It can compromise privacy, confidentiality, integrity and availability depending on the malicious code and the attacker motivation.*
16. **Protection mechanism failure:** Services are often equipped with security mechanisms that provide defence against various attacks. However, a security weakness arises when a service does not use or uses an insufficient protection mechanism. In case of an insufficient protection mechanism, a service could be saved from certain attacks but could not be saved from others. For example, service A is vulnerable to both Distributed Denial of Service (DDoS) attack and service injection attack. However, the security mechanism it uses can only provide defence against the DDoS attack. Thus, service A can do nothing against the service injection attack. A missing security mechanism could expose service A to both types of attack.
    • *Threat Category TC-4*
    • *Impact: High - A service without any security mechanism or uses an insufficient protection mechanism may expose a service to various attacks.*
17. **Insufficient automated security evaluation:** Without a timely evaluation, services with malicious intent or vulnerabilities can cause all sorts of trouble such as leakage of information or financial losses. This can be compared to traditional viruses in software code. A simple example could be an insider of a bank inserting back-door code into a service component before the security evaluation has been performed, in order to get customers' personal information.
    • *Threat Category: TC-4*
    • *Impact: High - It could cause serious privacy and data protection issues for an organisation.*

18. **DDoS attack occurs on service composition:** DDoS attacks are not new for web-based services. Many high-profile companies have been victims of such attacks. A DDoS attack is easy to detect but difficult to prevent. The distributed nature of composite services makes them even more exposed to DDoS, since the attacker can attack any of the service components and inflict damage to the overall service. This broad attack surface is something that makes DDoS attacks even more likely than for isolated systems. A DDoS attack normally targets web services that have public access gateway. By flooding a server with requests, the service can be overwhelmed, thereby preventing valid access to the service.
    - *Threat Category TC-4, TC-7*
    - *Impact: High - This can make a service unavailable.*
19. **Manipulation of trust properties:** The role of reputation systems is to facilitate trust. Remote monitoring of fulfilment of a contract relies on trustworthy collection of real data. However, a dishonest service provider could change or manipulate some trust properties of one of the services that are involved in a composite service. This could happen by compromising the monitoring engine either by manipulating trust properties or submitting a fake report to increase the trust level. For example, suppose Johnny is a service provider with limited ethics. By setting up a large number of false composite services using a payment service he provides, he is able to boost the trust level of this payment service.
    - *Threat Category: TC-4, TC-5*
    - *Impact: High - Malicious users could control service reputation according to their goal. The integrity of the overall composition could be at risk.*
20. **Trustworthiness level variability:** The trustworthiness of single services and service providers often changes over time. This is also true when a service is used within a service composition. Maintaining trustworthiness helps consumer confidence and provides a safe environment for businesses to dynamically interact and carry out transactions. The trustworthiness of one of the component services can be deteriorated during the execution of a composite service. This may lead to a situation where a single service with low trustworthiness becomes a threat for the entire composition. For example, in a travel assistance composition, the reputation of the map service provider goes down; thus, the integrity of the entire travel assistance is threatened. The composite service cannot be trusted because the map service could impose a huge threat to the entire composition. It could have a major security flaw that may let an attacker launch a denial-of-service attack and damage the overall composition or it may put the data confidentiality at risk. It is therefore necessary to continuously monitor the trustworthiness of the services and decide to replace the deteriorated component service with another service with the same functionality as soon as the trustworthiness value falls below a threshold.
    - *Threat Category: TC-5*
    - *Impact: Medium - It can compromise privacy, confidentiality, integrity and availability depending on the type of an attack.*

21. **Untrusted outsourcing/delegation:** To deliver a service, providers might outsource it or delegate specific activities to other service providers. This might be dangerous if the service user does not trust the additional providers. An air traffic controller is relying on a certain composition to obtain accurate weather forecasts. The service provider that delivers rain/snow real-time data delegates this service to another provider that the controller distrusts. Distrust might concern both service delivery (the controller does not rely on that service) or the handling of data (the position of the aeroplanes might be confidential).
    • *Threat Category: TC-5*
    • *Impact: High - Untrusted outsourcing may put data confidential and privacy at risk. Furthermore, the untrusted service may not be adequately secured and may introduce several vulnerabilities. It could be a weak spot for the attackers to attack the overall composition.*
22. **False perception of trust for end user:** An untrustworthy composite service could boost its overall trustworthiness by including highly trustworthy service components. Most of these components do not have an active role in the composition; they are just there to contribute to the calculation of the trustworthiness level of the composition as a whole. This can occur if the mechanism for calculating trustworthiness is simply based on the average trustworthiness of included components. For example, Gary is a customer who is looking looking to buy a product online. The WrongWeb shop uses his preferred and trusted provider, SafePay, so he trusts the WrongWeb shop implicitly (false sense of security). However, when he makes a purchase, the received product is of a terrible quality and worse yet, the WrongWeb shop sells his contact information to spammers. The transaction itself goes without problems.
    • *Threat Category: TC-5, TC-6*
    • *Impact: High - Exploiting the reputation of others can give a service false credibility, enabling a large number of attacks. This credibility can be used to exploit assets from end users, and make trustworthiness/reputation mechanisms less trustworthy.*
23. **Reliance on untrusted inputs in a security decision:** In some protection mechanisms, security decisions such as authentication and authorisation are made based on the values of input such as cookies, environment variables, and hidden form fields. However, an attacker could change these inputs using customized client applications and bypass the protection mechanism. For example, a web-based email list manager may allow attackers to gain admin privileges by setting a login cookie to 'admin'.
    • *Threat Category: TC-5*
    • *Impact: High - This may lead to the exposure or modification of sensitive data or damaging service availability.*
24. **Inclusion of functionality from untrusted control sphere:** Services using or importing executable functionalities (a library or a widget) from an untrusted source could introduce several security issues. The functionality could be malicious in nature, outdated or contain other vulnerabilities.

- *Threat Category: TC-5*
- *Impact: High - This might lead to many different consequences depending on the included functionality, but some examples include injection of malware, damaging service availability or gaining access to sensitive data. In a composite service, malicious functionality could inflict damage to the overall composition. It depends on how often a service imports or uses functionalities from other services that are not evaluated for trustworthiness. Furthermore, the impact increases if there are insufficient protection mechanisms in place to check the functionalities that are borrowed and does not belong in the same domain.*

25. **Missing end user notification:** In a composite service, a recomposition may consist of replacing existing services with new ones. It is possible that the new composition fulfils user requirements but compromises some important properties. By not giving this information to the end user may, it my lead to severe or unintended consequences. For example, Donald is a business man who uses a stock quote service to see the current stock prices for certain important stocks. When Donald sets his preferences about which stock exchange service to use, he only sets the minimum required trustworthiness level and the maximum price of the service. When the initial web service is no longer usable due to the lowering of the services trust level, a free stock exchange service, now at the highest trustworthiness level, is inadvertently recommended to Donald's client. Unfortunately the free service has a 15-minute built-in delay for stock market data. Donald is not notified about this and loses money.
    - *Threat Category TC-6*
    - *Impact: Medium - A service may not deliver as expected.*
26. **End user gets annoyed by confirmations:** This is largely a usability problem, arising from the tension between the need to ensure users to consider the consequences of changes to a system (and their actions) and the desire of the user to focus on functional rather than non-functional aspects of the system. Although most users acknowledge the importance of security, it nonetheless often represents a hindrance to them achieving their intended aims. This is especially true in relation to notifications. The threat is therefore that an overabundance of notifications frustrates the user and makes him choose to fulfil functional desires over security. This can be mitigated to some extent by considerate approaches towards notifications (e.g., providing non-modal notifications, and avoiding repeated notifications), but achieving a suitable balance is a difficult technical problem.
    - *Threat Category TC-6*
    - *Impact: Medium - A user may agree to a reduced security policy unintentionally. This may lead to several security issues, i.e., a threat to data confidentiality.*
27. **Lack of usability in secure composition:** Breadth, depth and flexibility of provided features in a development tool can often lead to compromises in terms of usability. Creating an interface that is both technically rich and easy to use is a difficult proposition. One of the goals of the Future Internet

is to provide flexibility through the use of services, however this often means that complexity management is simply transferred from the end user to the service developer. This is particularly true in the development of generic services, for a developer may have to consider a variety of scenarios, and is therefore unable to make assumptions on how the deployed service will be used. Designing tools and techniques for dealing with this complexity introduces difficult usability challenges. Usability can be measured, but the process of determining the resulting threats is an uncertain process.

• *Threat Category TC-6*

• *Impact: Medium - A service developer may find the development environment too difficult to understand and eventually give up on using it.*

28. **Cascade failures:** In cascade failures, a failure in one system has an impact on the activities of other systems it interacts with. A real-world example of a cascade failure is the electrical blackout that affected much of Italy on 28 September 2003: the shutdown of power stations directly led to the failure of nodes in the internet communication network, which in return caused further breakdown of power stations [8] [9]. In terms of systems-of-systems (e.g., power stations attached to the national grid), the threat applies equally to composed software services and the Future Internet more widely.

    • *Threat Category TC-7*

    • *Impact: High - Cascade failures result in some of these other systems failing, which in turn have a cumulative impact on the remaining systems, and so on. Ironically the situation arises especially where back-ups and fail-safes have been put in place, but with the potential consequence that the cascade failures result in a complete failure of the entire composition of systems.*

29. **Corrupt load-balancing:** If one system fails due to an attack (e.g., denial-of-service), the remaining systems have to handle the load from the failed system. This may result in an additional backlog transferred from the failed system that pushes the remaining services over their capabilities. If one of these systems fails, even more load is transferred the remaining systems, and a further backlog, with the process repeating to cause a cascade of failures. Such cascade of failures can be attributed to the dynamic reassignment of services resulting from an attempt to address an existing failure.

    • *Threat Category TC-7*

    • *Impact: Medium - Dynamic system re-composition may be required to address an existing failure, which may affect the overall system operation.*

30. **Recomposition corrupts response time:** When a composite service reconfigures, its component services are rearranged and/or replaced. However, it is possible that some services of the composition are unable to effectively participate in the process of recomposition due to their availability/response time. For example, consider the case where one of the components of Service X is a storage service. Replacing this storage service would require a time-consuming migration task, since large data volumes are stored there. Unfortunately, the composite service is recomposing too frequently, thereby spending significant time on changing the storage service component.

    • *Threat Category TC-7*

• *Impact: Medium - Access to data could be restricted or may cause delay in accessing critical information.*

31. **Synchronisation threats:** In a composite services environment, services may suffer from synchronisation/timing issues that prevent the correct delivery of composite services. These synchronisation/timing issues might cause deadlock, race conditions and prevent the services to interact with each other. For example, the parallel execution of services means that deadlock might occur between two services if they both reach a state whereby they are waiting for input from the other.
    • *Threat Category TC-1,TC-7*
    • *Impact: High - This can cause severe interaction flaws.*
32. **Lack of trust between providers:** Assembling a service composition is not sufficient to ensure it works. Given their autonomy, service providers might refuse to collaborate when they do not trust each other. This may cause an unreliable composition which may fail to achieve its objectives. For example, a service composition is established to compute income taxes for a company's employees. Within this composition, service "Incomes" returns the income for employees, whereas service "Tax computation" determines the taxes to pay on the basis of the income. However, "Incomes" does not trust "Tax computation", for it does not guarantee an adequate level of confidentiality. Perhaps, "Tax computation" preserves it but has an incompatible trust certificate. Thus, a service that would be an excellent choice for a composition is unavailable due to the fact that it does not trust other candidate services that would participate in the composition.
    • *Threat Category TC-3, TC-7*
    • *Impact: Low - This may cause an unreliable composition which may fail to achieve its objectives.*

4 Countermeasure methods for the threats

From the scenario descriptions, we have devised a set of countermeasure methods for the threats to composite services described in Section 3.

Issues related to incompatible policies and laws can be tackled via design-time verification techniques (*M1*). This requires the interface of both individual and composite services to specify (i) allowed deployment locations, and (ii) the laws/policies that apply. Automated verification checks if the expected exchange of data between services complies with the laws/policies about, e.g., data privacy.

When design-time verification is inapplicable, the information flow has to be monitored and/or enforced at run-time (*M2*). This requires the service infrastructure to monitor data exchange through observable channels. Access control enforcement mechanisms ensure that confidential information is not accessed by unauthorized users. The distinction between data and information is fundamental here: while data exchange can be observed, there is always a risk that information flows in a way that cannot be directly observed.

If the policies of consumers and providers are incompatible, negotiation techniques (*M3*) can help to identify a trade-off that satisfies both parties. Policy federation patterns can be studied in this context.

Identity management systems (*M4*) can prevent (or at least make it more difficult) providers from assuming fake identities. These systems require each service to be bound to a legal entity (a human or an organisation). Trustworthiness/reputation mechanisms will be key for services to successfully operate in a volatile environment. However, these mechanisms have to be robust, both in terms of their computation algorithms (*M5*)—the computed value shall be as realistic as possible—and of their monitoring techniques (*M6*)—resistance to fake reports and attacks to integrity.

Notification mechanisms (*M7*) enable actors to get up-to-date information concerning consumers' and providers' trustworthiness (especially in case of relevant changes, either negative or positive). A possible way to implement notification is via publish/subscribe [11]. In addition to notification, service re-composition algorithms (*M8*) enable responding to decreasing trustworthiness levels. Re-composition should balance quality and stability, i.e. it should not disrupt the current composite service. A particular type of re-composition pattern involves relying on redundant service providers (*M9*). Though more expensive, this avoids the scenario where failure of a component service affects the composite service. Service re-composition shall take into account that services are not controllable agents; rather, their providers are autonomous in choosing when, how, and if to deliver a specific service (*M10*). Thus, while assembling composite services, such autonomy cannot be neglected.

A possible way to prevent composite services from including untrusted services is to provide explicit support to outsourcing (*M11*). This means that service interfaces have to specify whether such operation is allowed to be outsourced as well as providers and services that can/cannot be involved. Such method also requires that, at run-time, actual outsourcing can be observed.

Services can be certified at deployment-time (*M12*) to verify whether a service operates as declared by its interface. Relying on certified services prevents malicious providers from injecting their services in compositions. However, such technique requires access to the source code (or the availability of inspectable binaries). Certification is not sufficient to analyse all possible interactions a service may engage in. Consequently, it should be complemented by runtime interaction monitoring techniques (*M13*) to keep track of actual interactions services participate in. A different yet fundamental approach is to devise secure service development methods (*M14*) that, if followed by developers, prevents or significantly reduces the likelihood of attacks from insiders. Such methods may include pair programming, automated validation techniques, and the establishment of traceability links from requirements to code.

Service interfaces shall be expressive enough to represent fine-grained access control rules about the confidential data a service provides and needs (*M15*). This way, composite service designers can check which data will be disclosed (possibly to whom) and they can verify need-to-know properties, i.e., if data is

disclosed to some actor that does not need it. Another technique is to give service interfaces a contractual validity (*M16*): violations lead to penalties (e.g., negative feedback or economic loss). In service-level agreements, penalties are referred to as credits. Necessary condition to make *M16* applicable is that services are deployed in an environment where penalties can be enforced.

In order to overcome changes in security interfaces, partial planning techniques (*M17*) are a helpful technique. A partial plan is defined beforehand and, while the composite service is in place, and depending on the results of the execution, the plan might be incrementally refined in order to timely include services that are appropriate to deliver the expected outcome. Though sub-optimal, partial planning is more robust to unexpected circumstances than planning from scratch. An alternative approach is to define security interfaces that manifest temporal validity (*M18*). This would allow for composition to be defined having a temporal horizon in mind (the provider commits to the validity of the interface till a certain point in time). Such technique can be combined with partial planning to create robust compositions. A third way to cope with changes is to perform early binding of services before their actual usage (*M19*). Such solution works if providers are committed to deliver the services that have already been bound. Combining *M19* with *M18* allows service providers to avoid indefinite allocation of resources.

To reduce the effect of DDoS attacks, efficient and scalable access control engines are a possible solution (*M20*). Cloud computing techniques might be adopted to physically distribute the infrastructure over multiple computational nodes, still providing a unique logical interface. To help consumers in service selection, security interfaces can incorporate information about scalability (*M21*). For instance, the maximum amount of requests the provider can deal with or a distribution curve showing how performance and response time degrade with an increasing the number of users. Such details may be either informative or have contractual validity. A way to early detect DDoS attacks is to monitor service performance to detect degradations (*M22*). Upon detection, response mechanisms can be applied, e.g., migration/redeployment of existing services on different servers, refusal of all new requests, usage of existing techniques to filter out attackers.

Mechanisms should be put in place so that the functionality of the composite service is not endangered by continuous re-compositions needed to improve security performance (*M23*). This might include using utility functions that balance traditional quality-of-service factors and security properties. Monitoring service interconnections (*M24*) allows for preventing cascade failures. Indeed, a single service is often used by multiple consumers, and the effects of a failure (and also of a response) shall take such factor into account.

In order to ensure a throughput and response time, load balancing techniques (*M25*) can be exploited at service deployment-time. Composition techniques should therefore give priority to services with better resource availability. In order to guarantee redundancy in service provision, services shall be enriched with information that allows for specifying and monitoring redundancy constraints

(*M26*). If a service commits to redundant provision, its interaction with third-party services shall be monitored to verify that redundancy does not dissolve. Timing and synchronisation issues—that may affect timely delivery of a composite service—can be tackled by conducting test cases (*M27*). If the set of test cases is defined systematically, the tests can dramatically reduce the likelihood of incurring in such issues at run-time.

Providers can specify, in service interfaces, information concerning what type of log information will be kept, which policies will be applied, and how such policies will be enforced (*M28*). The inherent limitation of such technique is that it requires information about how specific service providers work, which organisations are typically unwilling to disclose.

The design of composite services should take that into account, and minimize the risk of frequent re-composition requests that might lead to users carelessly pressing a "confirm re-composition" button (*M29*). More generally, interaction design aspects shall be seriously taken into account when designing composite services and composition mechanisms. The results of formal verification techniques can be abstracted using higher-level models (*M30*), so to ensure designers consider such results to improve the composite service. For example, this means interpreting issues at the organisational level or showing which are the risks that affect the interactions between services. In order to improve the way service designers/composers assemble services in a secure and trustworthy way, training sessions can be foreseen and organised (*M31*). These sessions provide designers with a methodological approach and with knowledge about the verification techniques that are performed by design-time tools.

Most security problems are continuously reoccurring and with known solutions/mitigation strategies. However, developers are not always aware of the available mitigation strategies. By providing the relevant information for a composition the developer will receive definitive advice and will have the knowledge to make more informed decisions (*M32*). If a specific composite service is attacked, similar services or services using some of the same components are likely to be threatened. An early warning system (*M33*) would notify these other services in advance so that they are able to prepare themselves (e.g., via recomposition).

Table 2 summarises the countermeasure methods. The "type" column classifies the methods according to their main function: (i) Prevention (P) methods avoid the occurrence of a threat; (ii) Monitoring (M) refers to observing relevant events that might suggest a threat; (iii) Verification (V) collects analysis techniques that check whether some security/trustworthiness property is guaranteed; (iv) Diagnosis (D) means correlating monitoring data to determine if a threat exists and to identify the root cause of such threat; (v) Response (R) methods mitigate the threat impact after it occurs. The "phase" column describes at which stage of the service engineering process the method applies: design-time (Des), deployment-time (Dep), and run-time (Run).

Method	**Type**	**Phase**
M1 : Design-time security verification takes into account policies/law	V	Des
M2 : Monitor information flow and enforce it using access control rules	M, R	Run
M3 : Policy negotiation automatically performed	D, R	Run
M4 : Detect fake services by keeping track of the identity of the provider	M, D	Run
M5 : Robust trustworthiness/reputation computation mechanisms	D	Des, Run
M6 : Robust trustworthiness/reputation monitoring mechanisms	M	Run
M7 : Monitor and notify changes in reputation/trustworthiness	M	Run
M8 : Recompose when trustworthiness and reputation are decreasing	R	Run
M9 : Create (re)compositions that rely on redundant service providers	P, R	Dep, Run
M10 : Consider providers' autonomy while composing services	P	Des, Dep, Run
M11 : Explicit support to outsourcing (sub-contracting)	P, M	Des, Run
M12 : Deployment-time service certification	V, P	Dep
M13 : Run-time interaction monitoring	M, D	Run
M14 : Secure service development method to prevent insiders attacks	P	Des
M15 : Service interfaces specify fine-grained access control	P, M, D	Des, Run
M16 : Contractual service interfaces, violations lead to penalties	M, R	Run
M17 : Partial planning techniques to enable incremental compositions	P, R	Run
M18 : Security contracts manifest temporal validity	P	Des, Run
M19 : Early binding of services before actual invocation	R	Des, Run
M20 : Scalable access control verification engines	P	Run
M21 : Incorporate scalability information in security interfaces	P	Des
M22 : Monitor service performance to early detect DDOS attacks	P, R	Run
M23 : Consider functionality/service to be delivered during adaptation	P	Run
M24 : Predict cascade failures by monitoring service interconnections	P, D	Run
M25 : Load balancing mechanisms while deploying service compositions	P, R	Dep, Run
M26 : Redundancy specification and monitoring	M, D	Des, Run
M27 : Test cases to check synchronisation/timing issues in compositions	V, P	Des
M28 : Protect logs using the same policies that apply to services	M, D	Des, Run
M29 : Avoid pressing "confirm re-composition" due to annoyance	P	Des, Run
M30 : Design tools should abstract the results of formal verification	P	Des
M31 : Training sessions to educate designers of service compositions	P	Des
M32 : Provide information about threat/attack method	P	Des
M33 : Early warning	P	Run

Table 2: Taxonomy of the countermeasure methods

5 Research directions

Our study on threats and countermeasures has helped us identify the following prospective techniques and research directions for designing, building, and operating secure and trustworthy composite services:

- **Trustworthiness/reputation management.** In the scenarios where consumers and service providers are unknown at design-time and where the service composition is performed with providers that do not know each other, trustworthiness and reputation management will be essential. We envisage that the challenge will be to provide mechanisms that: (i) enable consumers and service providers to obtain information about the reliability of others; and (ii) enable to monitor and compute trustworthiness and reputation in a robust way free of bootstrapping and malicious attackers. Different factors to evaluate should be considered, such as opinions by peers, information about compliance, and certifications released by trusted third parties.
- **Expressive security interfaces for services**. Whereas current service providers represent both functional and non-functional properties about their offered service through the specification of service interfaces, this seems to be largely inadequate to represent security and trustworthiness properties for service compositions. The development of new future languages shall allow service providers for a comprehensive specification of the security and trustworthiness properties they guarantee. Some of them could be: (i) fine-grained access control policies that indicate which information can be shared and with whom, as well as specific services that can or cannot be included in the composition; (ii) redundancy guarantees to increase the reliability; (iii) the threats that affect the composite service and the countermeasures that are deployed to address them.
- **Early warning and response.** Currently, when a threat or security issue that affects a service composition is detected, a reconfiguration or recomposition of the services is performed in response. However, this reactive approach is only a mitigation and does not prevent the occurrence of an event. Early warning and response mechanisms, taking advantage of risk assessment techniques to determine when threats are likely to occur, would enable proactive switching to alternative compositions.
- **Certification at deployment time.** Certification techniques (especially if the certificates are issued by trusted third parties) that guarantee the trustworthiness of a new service deployed (and even their providers) will play a fundamental role to be considered in service compositions. We envisage that these certifications might include information about the structure and composition of a service, the development methodology followed at design-time, or a commitment about the responsibility of the certification authority in case of a breach of agreements by the certified service.
- **Service recomposition revisited.** Existing techniques for recomposition of services are based on components as established in traditional software engineering methodology. Other mechanisms based on service-oriented settings shall be developed to work better in the new scenarios that arise from

the Future Internet. Some of the factors that should be taken into account for these new devised techniques are the following: (i) service providers are autonomous (consequently there is no central overall control and the action of composing services will be based on an interaction protocol among the participating service providers); (ii) threats are recomposition triggers (e.g., a recomposition process might be triggered by lower trustworthiness due to the expiry of a certificate); (iii) countermeasures are based on security patterns (service recompositions will typically consist of applying the most adequate pattern); (iv) service interfaces with contractual validity (the provider is committed to guarantee the declared properties in the socio-legal context where the service is deployed); and (v) incremental compositions (e.g. a service composition is only partially assembled at deployment and necessary services are added on the fly based on the availability and quality of service providers).

- **Representing laws, checking and enforcing their compliance.** Currently, there are no techniques to fully capture laws and associate them with the services where they apply to, e.g. to represent data confidentiality restrictions that apply in certain countries. Some of the challenges in the Future Internet will be to find mechanisms to ensure the compliance of a composition of services with respect to specific laws and, thus, the need of devising a representation of laws in a machine-understandable way.
- **Robust identity management systems.** Also related to legal issues, another relevant challenge in the Future Internet will be the development of robust identity management systems. Each entity in this new context shall be characterized by an identity and each service (atomic or composite) shall be unequivocally associated to its service provider, who could be have legal responsibilities about the service offered. This need of ensuring each user is who he says to be, is even more crucial in contexts of single sign-on where a single identity enables accessing to multiple systems.
- **Methodologies and CASE tools.** A large number of security issues could be produced by insider attacks, sometimes without harmful intention but due to lack of knowledge. There are many methodologies and tools that support the development of secure and trustworthy composite services, but they should be able to provide the results in such way that even non-security-expert developers can understand the risks and threats that can affect the composite service under design. Moreover, training sessions should support these methodologies and tools to guarantee their correct use and application.
- **Automated policy negotiation via flexible templates.** Static policies, such as "Use cryptography protocol X version Y" will be insufficient in the open environment of the Future Internet. More flexible and dynamic policies are required that allow interoperability between different service consumers and providers, dynamic negotiation of the policies in service composition (e.g., within the ranges that have been specified) or even include optional priorities, preferences and parameters that help perform a better matching.
- **Testing techniques for composite services.** The importance of testing is key not only for software, but also for composite services. The main difficulty

will be the opening of an environment where services in a composition can be replaced by others at run-time. Very little attention has been paid to this topic so far, which we envisage will be a crucial challenge in the future.

6 Related work

In this section we briefly present some of the main research projects and papers related to our work and based on the identification and taxonomy of threats and vulnerabilities and methods to deal with them.

During 2008 and 2009, the EU/FP7 project FORWARD identified possible new research areas and threats that need to be addressed. The main results of the project were presented in the FORWARD Whitebook [3], that contains not only the identified threats but also detailed and concrete scenarios of how potential malicious agents can take advantage of them. The main research areas identified by FORWARD were grouped into the following categories: networking, hardware and virtualisation, weak devices, complexity, data manipulation, attack infrastructure, human factors, and insufficient security requirements. The threats identified in the EU/FP7 project FORWARD were updated during 2011 in the SysSec project [4]. SysSec was a European project included in the Seventh Framework Programme that proposes to create a European Network of Excellence in the field of Systems Security and one of its goals is managing Threats and Vulnerabilities in the Future Internet. They decided to preserve the division of threats focusing on three main areas: malware and fraud, smart environment, and cyberattacks. Other related European projects in this area are: the Think-Trust project [5] that has produced a list of research challenges complementary to the RISEPTIS (Research and Innovation for Security, Privacy and Trustworthiness in the Information Society) Report (generated by a high-level advisory body in ICT research on security and trust), the WOMBAT (Worldwide Observatory of Malicious Behaviors and Attack Threats) project [6] that aimed at providing new means to understand the existing and emerging threats that are targeting the Internet economy and the net citizen.

Early work, such as the taxonomy from Landwehr et al.[13], Wang and Wang [20], Weber et al. [21] and Im and Baskerville [12] categorize security threats, flaws and vulnerabilities in a very broad sense related to computer programs. Mirkovic and Reiher [14] have published more specific taxonomies for attacks and defences related to DDoS attacks, but this is something we only treat as a class in our taxonomy. Babar et al. [7] have published a taxonomy of threats for the Internet of Things (IoT), which is more hardware-oriented than ours. The threats taxonomy from Mármol and Pérez[15] is, to the best of our knowledge, the most similar work to ours. They focus on threats trust and reputation models for distributed systems, which have been central aspects for our work as well.

Finally, important work is done through CAPEC [1] from the National Cyber Security Division of the U.S. Department of Homeland Security. CAPEC, the Common Attack Pattern Enumeration and Classification, is a public, interna-

tional and community-developed list of common attack patterns along with a comprehensive schema and classification taxonomy.

7 Conclusion

The Future Internet will be an environment in which a diverse range of services are offered by heterogeneous suppliers. In this environment users are likely to unknowingly invoke underlying services in a dynamic and ad hoc manner. The dynamic environment of service composition carries new security threats. Following a method where scenarios were contributed by seventeen European organisations, we have established a taxonomy of threats, consisting of seven high-level categories and thirty-two classes, and a taxonomy of thirty-three countermeasures that cover the entire life cycle of composite services.

The threats taxonomy is a comprehensive overview of specific dangers for composite services, that was devised through a thorough analysis of existing and potential vulnerabilities, and is clearly focused on trustworthiness aspects. The taxonomy is not meant to be exhaustive, as new threats will inevitably appear in the future. Our identified research directions provide recommendations on how to put countermeasure methods into practical use.

References

1. CAPEC, the Common Attack Pattern Enumeration and Classification, http://capec.mitre.org/
2. CWE (Classified Weakness Enumeration), http://cwe.mitre.org/
3. Forward project, http://www.ict-forward.eu/
4. SysSec project, http://www.syssec-project.eu/
5. Think-Trust project, http://www.think-trust.eu/
6. WOMBAT (Worldwide Observatory of Malicious Behaviors and Attack Threats), http://www.wombat-project.eu/
7. Babar, S., Mahalle, P., Stango, A., Prasad, N., Prasad, R.: Proposed security model and threat taxonomy for the internet of things (IoT). In: Meghanathan, N., Boumerdassi, S., Chaki, N., Nagamalai, D. (eds.) Recent Trends in Network Security and Applications, Communications in Computer and Information Science, vol. 89, pp. 420–429. Springer Berlin Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-14478-3_42
8. Berizzi, A.: The Italian 2003 blackout (June 2004)
9. Corsi, S., Sabelli, C.: General blackout in Italy Sunday September 28, 2003, h. 03:28:00 (June 2004)
10. Hernan, S., Lambert, S., Ostwald, T., Shostack, A.: Uncover Security Design Flaws Using The STRIDE Approach, http://msdn.microsoft.com/en-us/magazine/cc163519.aspx
11. Hoffman, K., Zage, D., Nita-Rotaru, C.: A survey of attack and defense techniques for reputation systems. ACM Comput. Surv. 42, 1:1–1:31 (December 2009), http://doi.acm.org/10.1145/1592451.1592452

12. Im, G.P., Baskerville, R.L.: A longitudinal study of information system threat categories: The enduring problem of human error. SIGMIS Database 36(4), 68–79 (Oct 2005), http://doi.acm.org/10.1145/1104004.1104010
13. Landwehr, C.E., Bull, A.R., McDermott, J.P., Choi, W.S.: A taxonomy of computer program security flaws. ACM Comput. Surv. 26(3), 211–254 (Sep 1994), http://doi.acm.org/10.1145/185403.185412
14. Mirkovic, J., Reiher, P.: A taxonomy of ddos attack and ddos defense mechanisms. SIGCOMM Comput. Commun. Rev. 34(2), 39–53 (Apr 2004), http://doi.acm.org/10.1145/997150.997156
15. Mrmol, F.G., Prez, G.M.: Security threats scenarios in trust and reputation models for distributed systems. Computers & Security 28(7), 545 – 556 (2009), http://www.sciencedirect.com/science/article/pii/S0167404809000534
16. NIST, C.: Glossary of Key Information Security Terms, http://csrc.nist.gov/publications/nistir/ir7298-rev1/nistir-7298-revision1.pdf
17. Papadimitriou, D.: Future Internet–The Cross-ETP Vision Document. Available online (2009), http://www.future-internet.eu/fileadmin/documents/reports/Cross-ETPs_FI_Vision_Document_v1_0.pdf
18. Shirey, R.: Internet Security Glossary, Version 2 (RFC4949) (2007), http://www.rfc-base.org/rfc-4949.html
19. Stoneburner, G., Goguen, A., Feringa, A.: Risk management guide for information technology systems recommendations of the national institute of standards and technology. Nist Special Publication 800(30), 55 (2002), http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf
20. Wang, H., Wang, C.: Taxonomy of security considerations and software quality. Commun. ACM 46(6), 75–78 (Jun 2003), http://doi.acm.org/10.1145/777313.777315
21. Weber, S., Karger, P.A., Paradkar, A.: A software flaw taxonomy: Aiming tools at security. SIGSOFT Softw. Eng. Notes 30(4), 1–7 (May 2005), http://doi.acm.org/10.1145/1082983.1083209