Server-driven Outbound Web-application Sandboxing
FP7-ICT-2009-5, Project No. 256964

https://www.websand.eu

Deliverable D1.3
Informal security policies and Legal Considerations

Abstract

This deliverable describes the security requirements and security properties of the patterns of interaction for the Internet of the Future that are described in D1.2. The security requirements are analysed from the point of view of the end-user, the service provider and other parties involved in the interaction patterns.

Deliverable details

Deliverable version: *v1.0*
Classification: *public*
Date of delivery: *30.09.2011*
Due on: *30.09.2011*
Editors: *Jan Wolff, Siemens AG*
Total pages: *27*
List of Contributors:
Bastian Braun, Jorge Cuellar, Martin Johns, Peng Liu, Michael Kirchner, Joachim Possega, Jan Stijohann, Walter Tighzert, Monica Verma, Jan Wolff

Project details

Start date: *October 01, 2010*
Duration: *36 months*
Project Coordinator: *Martin Johns*
Partners: SAP, Siemens, CHALMERS, KUL, UNI PASSAU

Executive Summary

This document is the first version of deliverable D1.3, due in project month M12, which contains the state of work on security requirements of the interaction patterns, but does not yet contain legal considerations. These will be included in the final version of this deliverable in project month M24. In the description of work, the deliverable D1.3 is described on page 37.

The main part of this deliverable is Section 2, which contains the first version of the description of the security requirements and security properties. These are derived from the interaction patterns, which are described in deliverable D1.2, from the point of the application and the end-user.

In the current version, only an effort of 3 out of the total of 22 person months, that have been planned for the second version in M24, has been invested into the deliverable.

Contents

1 Introduction

This deliverable describes the security requirements and security properties of the patterns of interaction for the Internet of the Future which have been defined in D1.2.

In Section 2, the interaction patterns are analyzed from the point of view of the different parties involved: the end-user, the provider of the web application and possibly other parties. For every interaction pattern in the pattern catalog of D1.2, the pattern is briefly summarized and the security requirements and shortcomings of current implementations are described. For some interaction patterns, this analysis is not yet complete. These section will be added to the second version of this deliverable.

An important part of this deliverable will contain the legal considerations, which will also be included in the second version after the legal partners from Passau has entered the project.

2 Security Requirements and Security Properties

In this section, the interaction patterns are analyzed from the point of view of the difference parties involved: the end-user, the provider of the web application and possibly other parties. For every interaction pattern in the pattern catalog, the pattern is briefly summarized and the security requirements and shortcomings of current implementations are described.

First, an overview of the pattern catalog is given and the structure of the following section that each analyze one of the patterns is described.

2.1 Pattern Catalog

The following table lists the identified patterns from D1.2.

ID	Interaction Pattern
P01	Interaction with a Web Page
P02	Client-side Third Party Content Integration
P03	Client-Side Third Party Library Integration
P04	Third Party User Interface Integration
P05	Resource Reference Sharing
P06	Third Party User Tracking
P07	Geolocation Sharing
P08	Capability-based Authorization Delegation
P09	Distributed Workflow
P10	Out-of-band distributed service

Table 1: The catalog of Interaction Patterns from deliverable D1.2.

For every interaction pattern, first the pattern and its context is reviewed by giving a short description of the pattern and the involved parties. The assumptions and expectations of the interacting parties that form the security requirements are described. The second part of the description of each pattern contains a list of observations regarding current implementation practices followed by a list of shortcomings that are related to the list of security requirements.

2.2 P01 Web Interaction

Pattern Overview A user actively navigates to a web page and interacts with it by consuming content or sharing information with the page.

Actors:

- *User:* The person using a web browser to access a web page. The user is interested in interacting with the web page by retrieving information from or sharing information with the page.
- *Web server:* The web server represents the content provider that delivers the web page.
- *Web page:* The web page is served by the primary web server and rendered by the browser of the user.

Requirements

- *Privacy (User and primary web page):* The fact that a user interacts with the primary web page must not become known to external and uninvolved parties.
- *Privacy (User and primary web page):* A user actively wants to interact with the primary web page loaded from the primary web server. The primary web page may integrate elements of 3rd party web servers (e.g. images, videos, feeds, ads, social media components, etc.). If the user actually does not make use of these external elements, the respective 3rd parties must not be able to track information about the user. If the user only interacts with elemen ts loaded from the primary web server, no user information (e.g. access frequency, IP address, browser information, cookies, referer, etc.) must be sent to the 3rd party.
- *Privacy (User and primary web page):* If the primary web page makes use of tracking mechanisms (e.g. via cookies to recognize users and track behavior over multiple visits), the user must be informed about what data is collected.
- *Confidentiality (User and primary web page):* Information that the user shares with the primary web page must not become known to external and uninvolved parties. This refers to both user-provided data and automatically generated session information (e.g. cookies, one-time tokens, etc.).

- *Integrity (User and primary web page):* A user must have the possibility to check whether the primary web page shown in the browser indeed belongs to the primary web server and accordingly to the party the user wants to communicate with. It must not be possible for 3rd parties to spoof elements of the primary web page (both code and visual contents). A user expects to actually communicate with the intended party in case the following conditions are met:
  - the user establishes a secure (HTTPS) connection,
  - the correct URL is shown in the address bar of the web browser and
  - the server certificate has been successfully validated.
- *Integrity (User and primary web page):* External parties must not be able to affect or alter the communication between the user and the primary web page. This includes both the data a user shares with the primary web page as well as the actions a user performs while using an application.
- *Integrity (User and primary web page):* When a user interacts with a primary web page, the actions conducted by the web site must be within the expectations of the user. For example, the primary web page must not deliver malicious elements (e.g. browser exploits, cross-site scripting code, etc.) not associated with the functions the user actually wants to use. A user must be provided with functionality to restrict which actions the primary web page is allowed to perform.
- *Availability (User and primary web server):* If the primary web server is available and a network connection between the user and the primary web server can be established, it must not be possible for a 3rd party to interrupt the communication between the two endpoints or otherwise render the service unavailable for a user.

Current implementation In connection with the current state of web technology, several of the requirements listed above cannot be fulfilled effectively. The following list highlights the security-relevant issues.

- Current web technology does not effectively hide the fact that a user interacts with a primary web page. If, for example, a user navigates away from a primary web page, the referrer header discloses information about what page the user previously visited. This includes both

the URL path as well as parameters or one-time tokens transported via URL parameters. Furthermore, browsers by default store history information that can be read by 3rd parties using mechanisms like the "CSS history hack".

- Privacy and confidentiality requirements cannot be fulfilled, if 3rd party elements (e.g. images, videos, feeds, ads, social media components, etc.) are integrated into a primary web page. Even though a user only wants to interact with content loaded from the primary web server, the 3rd party elements are loaded automatically, thereby disclosing information like the referrer, access time and frequency, browser information, IP address, etc.

  Furthermore, depending on the browser configuration, 3rd parties can set cookies and thereby track a user over multiple requests and primary web site visits.

- If cross-site request forgery countermeasures have not been implemented, 3rd parties are able to affect the communication channel between a user and a primary web page. By convincing a user to click on a crafted link or to visit a 3rd party web page with hidden elements, interactions with the primary web page can be forged.

- A user does not have any guarantee that elements provided by the primary web page indeed represent the content and functionality the user wants to interact with. If, for example, the primary web page server compromised, browser exploits or cross-site scripting attacks may be performed via the primary web page. Security functions in this area exist (e.g. content security policy), but are not widely adopted. A user does not have the possibility to restrict what the primary web page is actually allowed to do (e.g. read information from the client, store information, etc.).

Identified Security Shortcomings

In connection with the current state of web technology, several of the requirements listed above cannot be fulfilled effectively. The following list highlights the security-relevant issues.

- Current web technology does not effectively hide the fact that a user interacts with a primary web page. If, for example, a user navigates away from a primary web page, the referrer header discloses information about what page the user previously visited. This includes both

the URL path as well as parameters or one-time tokens transported via URL parameters. Furthermore, browsers by default store history information that can be read by 3rd parties using mechanisms like the "CSS history hack".

- Privacy and confidentiality requirements cannot be fulfilled, if 3rd party elements (e.g. images, videos, feeds, ads, social media components, etc.) are integrated into a primary web page. Even though a user only wants to interact with content loaded from the primary web server, the 3rd party elements are loaded automatically, thereby disclosing information like the referrer, access time and frequency, browser information, IP address, etc.

  Furthermore, depending on the browser configuration, 3rd parties can set cookies and thereby track a user over multiple requests and primary web site visits.

- If cross-site request forgery countermeasures have not been implemented, 3rd parties are able to affect the communication channel between a user and a primary web page. By convincing a user to click on a crafted link or to visit a 3rd party web page with hidden elements, interactions with the primary web page can be forged.

- A user does not have any guarantee that elements provided by the primary web page indeed represent the content and functionality the user wants to interact with. If, for example, the primary web page server compromised, browser exploits or cross-site scripting attacks may be performed via the primary web page. Security functions in this area exist (e.g. content security policy), but are not widely adopted. A user does not have the possibility to restrict what the primary web page is actually allowed to do (e.g. read information from the client, store information, etc.).

2.3 P02 Client-side Third Party Content Integration

Pattern Overview A user visits a web page which consists of elements provided by the corresponding primary web server but also contains content from a third party web server. The user may or may not be aware of the existance of third party content.

Actors:

- *User:* The person using a web browser to access the primary web page. The user may not have knowledge about which elements on the primary web page originate from a third party.
- *Primary web page:* The primary web page is served by the primary web server and rendered by the browser of the user. As a part, it contains third party content.
- *Primary web server:* The content provider that delivers the primary web page.
- *3rd party web server:* The 3rd party web server provides content that is integrated into the primary web page.

Security Requirements

Note that this base pattern does not cover any interactions the user may have with the third party via the integrated elements. See the following IAPs which cover this aspect for the resulting more sophisticated security requirements.

- *Privacy (User):* No personal or technical information such as browser configuration must be sent to the 3rd party.
- *Confidentiality (Primary web server and primary web page):* The content of the primary web server and of the primary web page must not become known to the 3rd party web server.
- *Confidentiality (User and primary web page):* Information that the user shares with primary web page elements loaded from the primary web server must not be available to the 3rd party.
- *Integrity (Primary web page):* The 3rd party platform must not be able to introduce elements into the primary web page that are unrelated to the expected content (e.g. privacy-violating content or spoofed content in case of 3rd party compromise). There must be mechanisms to either inspect, restrict or guarantee the correctness the content loaded from the 3rd party web server.
- *Integrity (Primary web page):* The visual elements loaded from the 3rd party web server must not be able to exceed given areas of the primary web page (e.g. size boundaries).

- *Integrity (Primary web page):* From the users perspective, the 3rd party content must not introduce unexpected changes to the primary web page.
- *Availability (Primary web server):* The availability of the primary web server must not be reduced by embedding the 3rd party content into the primary web page.
- *Availability (Primary web page):* In case the 3rd party web server becomes unavailable, the user must still be able to interact with the primary web page loaded from the primary web server in a sensible way.

Example Instantiations

Currently, there exist two common technical realizations. One uses an iframe, while the second one integrates 3rd party JavaScript code directly into the context of the primary web page. The iframe integration variant uses an iframe that loads elements and JavaScript code from the 3rd party web server.

Identified Security Shortcomings

Both instantiations variants currently cannot fulfill all aspects of the security requirements listed in deliverable D1.3.

Due to the frame separation, 3rd party JavaScript code is not executed in the same context as the elements loaded from the primary web server. While this solves certain confidentiality and integrity concerns, other issues remain unaddressed. In the JavaScript (XFBML) integration variant script code is loaded from the 3rd party web server and integrated into the primary web page.

When using this variant, the JavaScript code loaded from the 3rd party has unrestricted access to the primary web page, as it runs in the same context. This gives the 3rd party full read and write access to the DOM of the page.

Generally, the following security requirements are not met in connection with the currently available integration variants and web technologies:

- In case the JavaScript integration variant is used, the 3rd party code has full access to all page contents and can both read and write the DOM. This enables manipulations of the primary web page and allows the 3rd party to access information the user shares with the primary web page.

- Currently, primary web page elements loaded from the 3rd party web server are unrestricted. If, for example, the 3rd party platform gets compromised, the original page can be changed to arbitrary contents.
- In both integration variants it cannot be prevented that the 3rd party introduces privacy-violating content (that e.g. reads user information using the "CSS history hack"). Beside current browser sandbox mechanisms there is no technical restriction on what information the 3rd party can read from the client and transmit back to the 3rd party web server.
- The visual elements loaded from the 3rd party web server technically are not restricted to certain size boundaries or areas of the primary web page. When the JavaScript integration variant is used, arbitrary elements of the primary web page can be changed directly. When the iframe integration variant is used, the embedding web page can be overwritten using "frame busting" techniques.

2.4 P03 Client-Side Third Party Library Integration

Pattern Overview A user visits a web page wich leverages functionality provided by a third party in form of a code library. The functionality is rather basic and tighly integrated with the website, i.e. the user is not able to distinguish between primary web page and third party functionality, or, may not be aware of the integration of third party fucntionality at all.

Actors

- *User:* The person using a web browser to access the primary web page. The user may not have knowledge about which elements on the primary web page originate from a third party.
- *Primary web page:* The primary web page is served by the primary web server and rendered by the browser of the user. Not visible to the user, this page leverages a third party library.
- *Primary web server:* The content provider that delivers the primary web page.
- *3rd party web server:* The 3rd party web server provides the library that is used in the primary web page.

- *3rd party JavaScript (JS) Library:* A set of functions and data. The primary web page just includes a reference to the library, it is loaded dynamically when the browser processes the primary web page.

Security Requirements

- *Privacy (User):* No personal or technical information such as browser configuration must be accessible by the 3rd-party JavaScript (JS) library.
- *Privacy (User and primary web page):* No user's private information, while interacting with the web page, such as mouse click, keyboard strokes, etc. should be accessible by the 3rd-party JS library.
- *Confidentiality (Primary web server and primary web page):* The content of the primary web server and of the primary web page must not become known to the 3rd party JS library.
- *Confidentiality (User and primary web page):* Information that the user shares with primary web page elements loaded from the primary web server must not be available to the 3rd party JS library.
- *Availability (Primary web server):* The availability of the primary web server must not be reduced by embedding the 3rd party JS library into the primary web page.
- *Availability (Primary web page):* In case the 3rd party JS library becomes unavailable, the user must still be able to interact with the primary web page loaded from the primary web server in a sensible way.

Example Instantiations

Several JavaScript libraries such as jQuery and Dojo exist that contain premade software constructs and functionality. A third-party provider offers its own standard API that is embedded into the primary website and acts as a holder for the library. The properties, functionality and features of the third-party library is controlled and governed by the provider itself. The library may be customized by the provider to suit the requirements of the specific user

Identified Security Shortcomings

These libraries form a unique challenge because their functionality can not be easily isolated from the rest of a webpage; i.e. the potentially malicious or infected library code shares the same set of permissions as the integrator code.

2.5 P04 Third Party User Interface Integration

Pattern Overview A user visits a web page wich contains a third party user interface, i.e. the page integrates third party content which is clearly visible to the user and he is able to tell its origin.

Actors

- *User:* The person using a web browser to access the primary web page. The user may not have knowledge about which elements on the primary web page originate from a third party.
- *Primary web page:* The primary web page is served by the primary web server and rendered by the browser of the user. As a part, it contains third party content.
- *Primary web server:* The content provider that delivers the primary web page.
- *3rd party web server:* The 3rd party web server provides a widget content that is integrated into the primary web page.
- *3rd party widget:* The widget (interface and content) provided by the 3rd-party web server integrated into the primary web page.

Security Requirements

- *Privacy (User):* No personal or technical information such as browser configuration must be sent to the 3rd party.
- *Confidentiality (Primary web server and primary web page):* The content of the primary web server and of the primary web page must not become known to the 3rd party widget.
- *Confidentiality (User and primary web page):* Information that the user shares with primary web page elements loaded from the primary web server must not be available to the 3rd party.

- *Integrity (Primary web page):* The 3rd party platform must not be able to introduce elements into the primary web page that are unrelated to the expected content (e.g. privacy-violating content or spoofed content in case of 3rd party compromise). There must be mechanisms to either inspect, restrict or guarantee the correctness the content loaded from the 3rd party web server.
- *Integrity (Primary web page):* The visual elements loaded from the 3rd party web server must not be able to exceed given areas of the primary web page (e.g. size boundaries).
- *Integrity (Primary web page):* From the users perspective, the 3rd party widget and its content must not introduce unexpected changes to the primary web page.
- *Availability (Primary web server):* The availability of the primary web server must not be reduced by embedding the 3rd party content into the primary web page.
- *Availability (Primary web page):* In case the 3rd party web server becomes unavailable, the user must still be able to interact with the primary web page loaded from the primary web server in a sensible way.

Example Instantiations

Many third parties provide standard user interfaces for integrating their content into other web pages. Such widgets serve as read-only streams. Possible belonging content is available in the read-only view. However, the user interface may additionally facilitate certain client-side events/actions, e.g. relocation of the markers on Google Maps i.e. the widget API may provide certain user-customizable features, however in a restricted manner, the primary control still stays with the provider. Known example instantiations are Google Widget, Twitter Widged, Weather Widget.

Identified Security Shortcomings

The security shortcoming of the afore-mentioned example instantiations are still under investigation and will be subject in the next version of this deliverable.

2.6 P05 Resource Reference Sharing

Pattern Overview Resource reference sharing refers to sharing the reference of a resource, for example a photo or text, with other users via integrated third party functionality.

Actors:

- *User:* The end-user that, via a browser, accesses a primary web page and makes use of the resource reference sharing functionality.
- *Primary web server:* The primary web server represents the content provider that delivers the primary web page. The user shares a reference to a resource contained in that web page.
- *Primary web page:* The primary web page is served by the primary web server and rendered by the browser of the user. The primary web page contains third party content elements that implement the resource reference sharing functionality.
- *3rd party web server:* The 3rd party web server provides the content elements wich are necessary for the resource reference sharing functionality. In addition, the reference to be shared is stored on a web page on his web server.

Security Requirements

A complete list of the security requirements of this pattern are still work in progress and will be subject in the next version of this deliverable.

Example Instatiations

With more and more social networks emerging and various providers stepping into the social media market, the number of providers offering the feature to 'share-bookmarks' is multiplying rapidly. Facebook, Google, Twitter, Digg, LinkedIn, and many more provide users the possibility of sharing bookmarks of various resources, they find interesting, across their social media profile. Facebook Like, Google +1, Digg share, Delicious bookmark, LinkedIn share, just to name a few, are available in form of HTML buttons that can be embedded via JavaScript code within the resource website. These social plugins/widgets, simply post/share the bookmark on user's profile at the social media platform.

Identified Security Shortcomings

The third-party social media platforms provide the user the possibility of sharing external bookmarks on their social media account or profile pages through either embedded JavaScript or iFrames.

When using the 'Facebook - I like' functionality while not authenticated at the 3rd party platform, a login dialog is presented. Generally, a user has no guarantee that the login information is actually transmitted to the intended 3rd party. The primary web server has the possibility to spoof the login dialog and thereby steal login credentials.

Due to restrictions in the Same-origin Policy itself with respect to <script> tags and iFrames, a number of security concern arises.

1. Confidentiality: Full access to the DOM via the third-party JavaScript code;

2. Integrity: If user is not logged in to their social media platform account, they receive a login dialog, which could be spoofed;

3. Mutual Authentication: The third-party server simply provides a login dialog for the user to log in;

4. Privacy: By embedding JS on the primary resource webpage from a 3rd Party Website, one provides the third-party access to track what resource the user is viewing, even before the user actually clicks the 'Bookmark' button. The only prerequisite to this is that the user should be logged on to the third-party website. The embedded-JavaScript code can access the information about the resource and the 3rd party can associate/link this information to the currently active user;

Comparison with other techniques like using postMessage, SubSpaces, etc. follows.

2.7 P06 Third Party User Tracking

Pattern Overview Third Party User Tracking refers to the integration of a third-party library to track and collect various information about a user's web activity. In most cases, this happens without the user's knowledge and consent.

Actors:

Description of involved parties:

- *User:* The person using a web browser to access a web page. The user is interested in interacting with the web page by retrieving information from or sharing information with the page.
- *Primary Website:* The primary web server hosts the primary website that is being tracked for collecting and observing users' web activities.
- *Third-party Tracking Server:* The third-party server that provides a tracking code embedded into the primary webpage for tracking purposes.

Security Requirements

- *Privacy (User and primary website):* User's private information such as Geo-coordinates, etc. must not be tracked and made available to the third-party.
- *Availability (User and primary web server):* Usage/Embedding of the third-party tracking code on the primary website must not adversely affect its availability to the user.
- *Availability (User and primary web server):* In case the third-party code becomes unavailable the user should still be able to use the primary website.
- *Confidentiality (User and primary web page):* Sensitive-information that the user shares with the primary web page must not become known to external and uninvolved parties. This refers to both user-provided data and automatically generated session information (e.g. cookies, one-time tokens, etc.).

Example Instatiations

Whether be it Google AdSense or Google Analytics, the third-party provides a piece of JS code that is referenced by the primary website, integrated and executed within the domain of the primary website being tracked or monitored. This 3rd-party *tracking* code has full-access to the website's content, user's browser information, cookies, etc.

Identified Security Shortcomings

In connection with the current state of web technology, several of the requirements listed above cannot be fulfilled effectively. The following list highlights the security-relevant issues.

Generally, the following security requirements are not met in connection with the currently available integration variants and web technologies:

- Since the JavaScript integration is done, the 3rd party code has full access to all page contents and can both read and write the DOM. This enables manipulations of the primary web page and allows the 3rd party to access information the user shares with the primary web page.
- Currently, primary web page elements loaded from the 3rd party web server are unrestricted. If, for example, the 3rd party platform gets compromised, the original page can be changed to arbitrary contents.
- In both integration variants it cannot be prevented that the 3rd party introduces privacy-violating content (that e.g. reads user information using the 'CSS history hack'). Beside current browser sandbox mechanisms there is no technical restriction on what information the 3rd party can read from the client and transmit back to the 3rd party web server.
- The visual elements loaded from the 3rd party web server technically are not restricted to certain size boundaries or areas of the primary web page. When the JavaScript integration variant is used, arbitrary elements of the primary web page can be changed directly.

2.8 P07 Geolocation Sharing

Pattern Overview This pattern encompasses multi-party component interactions that involve sharing of user's private information such as location with third-party components.

Actors:

- *User:* The person using a web browser to access the primary web page. The user may not have knowledge about which elements on the primary web page originate from a third party.
- *Primary web page:* The primary web page is served by the primary web server and rendered by the browser of the user. As a part, it contains third party content.
- *Primary web server:* The content provider that delivers the primary web page.

- *3rd party web server:* The 3rd party web server provides content that is integrated into the primary web page.

Security Requirements

Certain mashup interactions involve and require sensitive user information to be shared across different domains. Although, on one hand, this is an integral part of the working of the mashup functionality, on the other hand, it violates user privacy and raises security concerns.

- *Privacy (User and primary website):* The geolocation of the user or the geotag value of the user resource should not be accessible to the third-party.
- *Privacy (User and primary webserver):* The privacy properties of the geotagged information i.e. the property specifying with whom is the the geotag information shared, should be independent of the privacy properties of the resource itself. For e.g if a user's contact can see a photo's geotags should be independent of whether the contact can see the photo itself.
- *Availability (User and primary web server):* Usage/Embedding of the third-party tracking code on the primary website must not adversely affect its availability to the user.
- *Availability (User and primary web server):* In case the third-party code becomes unavailable the user should still be able to use the primary website.
- *Confidentiality (User and primary webserver):* The information/resource shared between the user and the primary server should not be accessible to the 3rd-party.
- *Confidentiality (User and primary web page):* Sensitive-information that the user shares with the primary web page must not become known to external and uninvolved parties. This refers to both user-provided data and automatically generated session information (e.g. cookies, one-time tokens, etc.).
- *Confidentiality (User):* The request for resources and the geo-information (or geotag metainformation) related to the resources should not be shared with 3rd-party.

Example Instantiations

Whether be it Google AdSense, Google Analytics or any other third-party user tracking service, the third-party provides a piece of JS code that is referenced by the primary website, integrated and executed within the domain of the primary website being tracked or monitored. This 3rd-party *tracking* code has full-access to the website's content, user's browser information, cookies, etc.

Identified Security Shortcomings

In connection with the current state of web technology, several of the requirements listed above cannot be fulfilled effectively. The following list highlights the security-relevant issues.

Generally, the following security requirements are not met in connection with the currently available integration variants and web technologies:

- Since the JavaScript integration is done, the 3rd party code has full access to all page contents and can both read and write the DOM. This enables manipulations of the primary web page and allows the 3rd party to access information the user shares with the primary web page.
- Currently, primary web page elements loaded from the 3rd party web server are unrestricted. If, for example, the 3rd party platform gets compromised, the original page can be changed to arbitrary contents.
- In both integration variants it cannot be prevented that the 3rd party introduces privacy-violating content (that e.g. reads user information using the "CSS history hack"). Beside current browser sandbox mechanisms there is no technical restriction on what information the 3rd party can read from the client and transmit back to the 3rd party web server.
- The visual elements loaded from the 3rd party web server technically are not restricted to certain size boundaries or areas of the primary web page. When the JavaScript integration variant is used, arbitrary elements of the primary web page can be changed directly.

2.9 P08 Capability-based Authorization Delegation

Pattern Overview Certain operations on assets owned by users are critical in nature due to the separation of trust boundaries between the communicating parties, sensitive nature of the assets, side-effect of the operation on

the assets and flow of the sensitive information across different trust domains. In such situation a need arises to delegate the process of authorization. The authorization mechanism relies on a capability-based access model.

Actors

- *User:* The person using a web browser to access the primary web page. The user may not have knowledge about which elements on the primary web page originate from a third party.
- *Primary web page:* The primary web page is served by the primary web server and rendered by the browser of the user. As a part, it contains third party content.
- *Primary web server:* The content provider that delivers the primary web page.
- *3rd party web server:* The 3rd party web server provides content that is integrated into the primary web page.

Security Requirements

- *Privacy (User and primary webserver):* The 3rd-party app should not be able to have access to users' accounts at the primary web server without the needed access rights.
- *Privacy (User and Web Browser):* Multiple third-party apps must be isolated from each other and the system (user's browser).
- *Availability (User and primary web server):* Delegating authorization access rights to the third-party app must not adversely affect the functioning and availability of the primary website and the user's profile.
- *Availability (User and primary web server):* In case the third-party app becomes unavailable the user should still be able to use the primary website.
- *Confidentiality (User and primary webserver):* Any information/content (message, post, etc.) shared between the user and the primary web server (i.e. the user account) must not become known to the third-party application.

- *Confidentiality (User and primary web page):* Sensitive-information that the user shares with the primary web page must not become known to external and uninvolved parties. This refers to both user-provided data and automatically generated session information (e.g. cookies, one-time tokens, logins credentials, etc.).

- *Access Rights (Primary web server and third-party app):* The tokens issued to the third-party app for accessing the user accounts must not be reusable or renewable without proper re-authentication/re-authorization.

Example Instantiations

A description of popular example instantiations is still under developement and will be included in the next version of this deliverable.

Identified Security Shortcomings

The security shortcoming are still under investigation and will be subject in the next version of this deliverable.

2.10 P09 Distributed Workflow

Pattern Overview While performing various web activites, the user consumes different types of online services such as posting comments, buying articles, etc. For many such services, utilizing a 3rd-party service is a vital step in the successful execution of the primary service consumption.

Requirements

- *Privacy (User and primary service provider):* Sensitive Information shared between the User and the Primary Service provider must not be accessible to the third-party service provider.

- *Privacy (User and Web Browser):* The 3rd-party service provider should not be able to track user's web activity in particular which primary website the user is accessing.

- *Privacy (User and third-party service provider):* The third-party service provider should only provide required information, related to the service requested, to the primary server. Any additional sensitive information shared between the user and the third-party service provider (authenticating credentials, etc.) should not be accessible to the primary service provider.

- *Availability (User and primary web server):* Branching off from the main work flow to access the third-party service must not adversely affect the functioning and availability of the primary web service.
- *Integrity (Primary service provider and third-party service provider):* The delegation of control between the primary and the third-party service provider (in either direction) should not lead to unwanted effects on the primary service provider, and hence the user.
- *Integrity (Primary service provider and third-party service provider):* Only legitimate cross-domain traffic between the primary and third-party service providers should lead to delegation of control.
- *Integrity (Primary service provider and third-party service provider):* The later steps of the primary service, intended to be followed after consumption of the third-party service, should be executable before the third-party service has been successfully executed.
- *Integrity (Primary service provider and third-party service provider):* The request (along with its parameters) to the third-party service provider must be verifiable.
- *Access Management (Primary service provider and third-party service provider):* Any access token or sensitive-parameters exchanged between the primary service provider and the third-party service provider should not be reusable. Such tokens must be time-sensitive.

Example Instantiations

Paypal (Payment Management): PayPal is an example of a CaaS (Cashier-as-a-Service) where the merchant implementing the online shopping workflow is responsible of providing the shopping service to the user. This work flow includes payment step which is performed ex- ternally by consuming payment services from the 3rd-party, in this case PayPal. For further description refer to Work Package 2 Deliverable D2.1 Section 2.4 Online Shopping Workflows over Two Domains.

Identified Security Shortcomings

The security shortcoming of the afore-mentioned example instantiation are still under investigation and will be subject in the next version of this deliverable.

2.11 P10 Out-of-band distributed service

Pattern Overview For increased Security, certain sensitive actions require explicit validation of the user. This is provided using a separate communication interface, e.g. a separate desktop application or an app on a mobile phone. The separate application queries for acknowledgement of the sensitive operation displaying the requesting-entity, the request in form of a text message and gives the possibility to authenticate and acknowledge or deny the request.

Security Requirements

- *Privacy (User and primary server):* Sensitive Information shared between any of the authorized users and the primary server must not be accessible to any of the other authorized users.
- *Access (User and Web Browser):* The out-of-band multi-factor credentials should act as one-time passcodes. User should be able to perform actions with a once-consumed secondary credential.
- *Availability (User and primary web server):* In case of unavailability of the out-of-band communication device, there should be an alternative method to allow the user to authenticate/verify himself in such a multi-stage process.
- *Availability (User and primary web server):* An unintended activity by an attacker during any of the out-of-band verification steps should not lead to unavailability of the system to a legitimate user. It should be possible to differentiate between a valid and an invalid request without causing adverse effects to legitimate users.
- *Integrity (Primary server and secondary verifying object):* It shouldn't be possible to capture/set/modify the credentials under way sent to the user via out-of-band mechanism.
- *Integrity (Primary service provider and third-party service provider):* Only legitimate cross-domain traffic between the primary and third-party service providers should lead to delegation of control.

Example Instantiations

Google Authenticator is a mobile application that allows you to gen- erate 2-step verification codes on your smartphone without a network connection.

We recommend users with smartphones to use Google Au- thenticator to generate verification codes instantly to sign in to their Google Apps accounts.

Identified Security Shortcomings

The security shortcoming of the afore-mentioned example instantiation are still under investigation and will be subject in the next version of this deliverable.

3 Conclusion

In this deliverable, the security requirements and security properties of the patterns of interaction for the Internet of the Future, which have been defined in D1.2, are described.

These interaction patterns have been analyzed from the point of view of the different parties involved: the end-user, the provider of the web application and possibly other parties. For every interaction pattern in the pattern catalog, the security requirements and shortcomings of current implementations are described. For some interaction patterns, this analysis is not yet complete and subject of the work in the next twelve month.

An important part of the next, final version of this deliverable will contain the legal considerations.