Secure Integration

Christian Grothoff

Berner Fachhochschule

31.5.2024
Learning Objectives

How can we securely integrate services?

Terminology

How to register a URI scheme?

Example: RFC 8905

Example: LSD 0006
Integration: Problem Statement

Isolation is a key paradigm in security:
- processes (address spaces!)
- users (quotas, access rights)
- departments (accounting, controlling, revision)
- organizations (auditors)
Integration: Problem Statement

Isolation is a key paradigm in security:

- processes (address spaces!)
- users (quotas, access rights)
- departments (accounting, controlling, revision)
- organizations (auditors)

How to ensure good user experience across application boundaries?
Solution domains

- Fax

Inter-process communication (UNIX Domain Sockets, Shared Memory, Networking)

Intents (Android-only!)

Deep links
Solution domains

- Fax
- Inter-process communication (UNIX Domain Sockets, Shared Memory, Networking)
- Intents (Android-only!)
- Deep links
Addressing

**URI**  Uniform Resource Identifier
**URL**  Uniform Resource Locator — object identification tied to location
**URN**  Uniform Resource Name — namespace independent of location
Structure of a URI

URI = scheme “:” hierarchical-part [ “?” query ] [ “#” fragment ]

- **scheme** Defines the method of identification
- **hierarchical part** Hierarchical access path of the URI
- **query** Search function
- **fragment** access to a part of the document
URI/URN examples

foo://example.com:8042/over/there?name=ferret#nose

scheme  authority  path  query  fragment

urn:example:animal:ferret:nose
The Internet Assigned Numbers Authority (IANA)

- Responsible for unique assignment of parameters and numbers in Internet protocols
- Operates the DNS root zone
- Performs the administrative work on behalf of ICANN and IETF
- Web site: https://iana.org/
- Used to be just Jon Postel
<table>
<thead>
<tr>
<th>Scheme</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ftp</td>
<td>File Transfer Protocol</td>
<td>[1]</td>
</tr>
<tr>
<td>http</td>
<td>Hyper Text Transfer Protocol</td>
<td>[2]</td>
</tr>
<tr>
<td>https</td>
<td>HTTP Secure</td>
<td>[7]</td>
</tr>
<tr>
<td>mailto</td>
<td>Electronic mail address</td>
<td>[4]</td>
</tr>
<tr>
<td>file</td>
<td>Host-specific file names</td>
<td>[1]</td>
</tr>
<tr>
<td>pop</td>
<td>Post Office Protocol v3</td>
<td>[3]</td>
</tr>
<tr>
<td>urn</td>
<td>Uniform Resource Names</td>
<td>[6]</td>
</tr>
</tbody>
</table>


[1] [http://www.iana.org/assignments/urn-namespaces](http://www.iana.org/assignments/urn-namespaces)
URI examples

http://prof.hti.bfh.ch/index.php?id=1403&L=2#howto
http://[2001:620:500:ff80::80]/owncloud

ftp://user:geheim@ftp.bfh.ch/

file:///C:/WINDOWS/system32/drivers/etc/services
mailto:firstname.lastname@bfh.ch

urn:ietf:rfc:3986
Security Considerations

- Link hijacking: malicious or competing apps can register for your scheme
- Data interception: links with sensitive data may be transmitted by users over insecure channels
- Access control bypass: ensure checking access when handling links
- Insecure parameter handling: strings are the source of all eval
How to register a URI scheme? [9]

There are *permanent* and *provisional* registrations. Always start with *provisional*, but largely follow *permanent* guidelines:

1. Write and publish citable specification (ideally, RFC-style) explaining the use-case, syntax, semantics and security considerations
2. Follow syntactic requirements and ensure name is not taken
3. Send a registration request to uri-review@ietf.org and possibly other relevant lists for discussion.
4. Respond to comments, address in specification where reasonable (wait a few weeks for discussion to conclude).
5. Submit updated registration request to iana@iana.org with pointer to the discussion.

You can always “upgrade” to *permanent* status later!
RFC 8905: payto: Uniform Identifiers for Payments and Accounts

Like mailto:, but for bank accounts instead of email accounts!

`payto://<PAYMENT-METHOD>/<ACCOUNT-NR>`

?subject=InvoiceNr42
&amount=EUR:12.50

Default action: Open app to review and confirm payment.
Benefits of payto://

- Standardized way to represent financial resources (bank account, bitcoin wallet) and payments to them
- Useful on the client-side on the Web and for FinTech backend applications
- Payment methods (such as IBAN, ACH, Bitcoin) are registered with GANA

\[2\text{https://gana.gnunet.org/}\]
Security Considerations for payto://

- Interactive applications handling the 'payto' URI scheme MUST NOT initiate any financial transactions without confirmation from the user and MUST take measures to prevent clickjacking.
- Unless a 'payto' URI is received over a trusted, authenticated channel, a user might not be able to identify the target of a payment. A payment target type SHOULD NOT use human-readable names in combination with unicode in the target account specification.
- The authentication/authorization mechanisms used to process a payment encoded in a 'payto' URI are handled by the application and are not in scope of this document.
- Payment target types SHOULD NOT include personally identifying information about the sender of a payment that is not essential to conduct a payment.
Syntax:

taler-URI = ("taler://" / "TALER://" / "taler+http://" / "TALER+HTTP://")

action path-abempty [ "?" opts ]

action = ALPHA *( ALPHA / DIGIT / "-" / "." )

opts = opt *( "&" opt )

opt = opt-name "=" opt-value

opt-name = ALPHA *( ALPHA / DIGIT / "-" / "." / ":" )

opt-value = *pchar

Example:

taler://pay-push/exchange.taler.grothoff.org/D83MG3W7WKVH3C9...
taler:// actions

withdraw  bank-initiated withdrawal
  pay   merchant-initiated payment
  refund merchant-initiated refund
  pay-push P2P payment
  pay-pull P2P invoice
  pay-template merchant offline payment
  restore restore from backup
withdraw-exchange wallet-initiated withdrawal
Hypertext Transfer Protocol – HTTP/1.1.  
Obsoleted by RFCs 7230, 7231, 7232, 7233, 7234, 7235, updated by RFCs 2817, 5785, 6266, 6585.

R. Gellens.  
POP URL Scheme.  
P. Hoffman, L. Masinter, and J. Zawinski.
The mailto URL scheme.
Obsoleted by RFC 6068.

A. Melnikov and C. Newman.
IMAP URL Scheme.
Updated by RFC 5593.
R. Moats.
URN Syntax.
Obsoleted by RFC 8141.

E. Rescorla.
HTTP Over TLS.
RFC 2818 (Informational), May 2000.
Updated by RFCs 5785, 7230.

Acknowledgements

Co-funded by the European Union (Project 101135475).

Co-funded by SERI (HEU-Projekt 101135475-TALER).

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union.

Neither the European Union nor the granting authority can be held responsible for them.