TAS3 Threat Analysis (Draft 02)

Editor: Sampo Kellomäki (sampo@symlabs.com), EIfEL

January 20, 2010

Abstract

TAS^3 is chartered with one overarching business requirement: "secure". For us to meet this requirement, we need to understand the threats. We discuss here various business, IT, operational, protocol, and crypto threats so that we can meet these challenges in our architecture.

Contributors
Michele Bezzi, SAP
Jeroen Hoppenbrouwers, Synergetics
Gang Zhao, STARLab, VUB

Contents

1 Introduction

This document should be read along side of TAS3 architecture documents. It is a requirement that *secure* architecture addresses all known threats. Therefore this document effectively specifies security related requirements (other more business oriented requirements are discussed in other documents) for the architecture.

Once architecture documents stabilize, the intent is to reference here the specific parts that solve these threats.

Our goal is to answer all threats specified in [NIST-SP800-30]. For programmers [Meier09] provides a good check list and [Meier08] a short list for those in a hurry. Further threats in [**?**] are addressed as well.

2 Threat Analysis

This section addresses Reqs. *D1.2-2.7-Safe* and *D1.2-2.8-Avail*. Also Req. *D1.2-2.9-Correct* is touched from secure programming perspective.

2.1 Other work

- [SAML2security]
- [IDWSFSecPriv]
- [NIST-SP800-30]
- [Meier09] provides a check list and [Meier08] a short list

2.2 Business threats

T21-IDTheft Identity Theft

T22-Illegal Illegal transactions

2.3 Trust model threats

T31-OverPrivil Over-privileged process and service accounts.

T32-UnTTP Untrustworthy Third Party.

2.4 Architectural threats

T41-BPMflaw Business process modelling flaws. How to audit or validate the model?

T42-BPMdel Accidental deletion of process steps such as authorization or validation

T43-Berserk Dynamically adaptive business process gone wild

2.5 Trust misconfiguration threats

T51-TNins Insertion of illegitimate members in trust network.

T52-Mole Moles in trust network. A previously trusted member turns into rougue. How to detect? How to rectify?

T55-PolluteRep Pollution of reputation of other party

T56-Augment Illicit augmentation of reputation by party itself

T57-Deface Defacing Front End web site can cause damage to reputation of the Front End. Defacing attacks can happen through same means as phising (T111-Phish) and also through break-ins, either through network, or physical.

T58-WrongCAs Insertion of wrong CAs into a trust network

T59-WrongAAs Wrongly assigning source of authority to an attribute authority

T510-WrongLOA Wrong assertion of LOA value by an IDP

2.6 Protocol misconfiguration threats

T61-Replay Replay attack. See CR211-Uniq.

T62-UnEncLink Unencrypted links e.g. SSLnull cipher suites.

T63-UnAnLink Unauthenticated links

2.7 Authorization misconfiguration threats

T71-WrongGrant Returning grant instead of deny leading to unauthorised access

T72-WrongDeny Returning deny instead of grant leaving to DOS.

T73-WrongObs Returning wrong obligations

T74-MissingObs Missing obligations from authz decisions

T75-OKAC Attribution of wrong access rights to an otherwise trusted member

2.8 Ontology threats

T81-SameButDiff Same term meaning different things to two parties leads to illicit access, due to wrong rule inadvertently matching.

2.9 Exposure threats

T91-Eavesdrop Exposure of data due to network sniffing or eavesdropping. Counter measure: encrypt data and manage keys right.

T91-DBLeak Exposure of data due to database files or transaction logs. Counter measure: encrypt data and manage keys right.

T92-TamperNet Modification of data in transit. Counter measuers: signing or verification of a hash over the data.

T93-TamperDB Modification of data in database.

T94-ExptLeak Error condition or exception reveals too much data or system details (usually to aid debugging). Exception output that appears in user interface or over the network is especially damning. Leakage to logs is also a threat.

T95-CoreLeak Error condition or exception causes a core dump that reveals too much data or system details (usually to aid debugging).

2.10 Privacy threats

T101-LeakBackup Eavesdropping on backups (see CR213-Backup)

T102-Correlation Database correlation by colluding entities (solution: do not leak correlation handles, i.e. use pseudonyms - see Architecture, Core Security Architecture, Access Credentials, Pull Model)

T103-TAIdP IdP collects traffic analysis (and then sells or illicitly use it). Some counter measures:

- TN wide data retention policy, audit this: add compliance requirement
- Pure play IdP operator vs. mixed functions
- Centralized IdP well managed may be a good idea

T104-TADI Disco collects traffic analysis (and then sells or illicitly use it)

T105-TA3rd Traffic Analysis by Third Party

T106-CorrAudit Correlation handles of audit trail will also become correlation handles.

T107-LogTokLeak If WSC parties keeps log of User's pseudonym along with encrypted form of User's identifier at WSP, then WSC and WSP can correlate and collude using the encrypted form. However this threat is acute only between directly interacting parties. In a chain of web services calls longer than 3, the nonneiboughring parties are not in position to collude using this attack.

Current solution is to forbid logging the tokens, see CR53-DontLogTok.

T108-PhishPII Tricking user to reveal PII through phising attack that poses a real looking web page to solicit PII. See also access version of the threat: T111-Phish.

T109-SocEngPII Social Engineering, talking users to revealing PII. See also access version of the threat: T112-SocEng.

T1010-SnoopPII Network eavesdropping to record PII.

T1011-KbdLogPII Keyboard logger or other malware to record credentials.

T1012-MalwarePII PII theft, e.g. copy private contact book, using malware.

T1013-PIITheft Physical theft of PII.

2.11 Authentication threats

T111-Phish Tricking user to reveal his authentication credentials through phising attack that poses a real looking web page to solicit user's access credentials. This could be created through

a. DNS manipulation
b. Cross site scripting
c. Inappropriate insertion of content in legitimate site
d. Containment of legitimate site in illegitimate frame

See also PII version of threat: T108-PhishPII.

T112-SocEng Social Engineering, talking users to revealing access credentials.

See also PII version of threat: T109-SocEngPII.

T113-SnoopCred Network eavesdropping to record credentials.

T114-KbdLog Keyboard logger or other malware to record credentials.

T115-Malware Credential theft, e.g. copy private key, using malware.

T116-Theft Physical credential theft.

T117-Dict Dictionary attack on password

T118-Brute Brute force attacks of simply trying out all credentials.

T119-Cookie Cookie replay attack. Use previously recorded cookie in context where authentication did not happen. Also arises if expired session cookie is allowed as a factor in authentication, resulting stronger factor not being demanded.

T1110-Lure Luring users to do stupid things like

a. Visit web sites that phish or contain malware
b. Install malware and troians
c. Voluntarily give out credentials or PII

2.12 Impersonation threats

T121-Gift Users giving away their credentials to others

T122-Loss Users losing their credentials

T123-Careless Users not protecting their credentials properly e.g. posting passwords on post-it stickers

T124-Delegation Authentication delegation models in which the delegate assumes the user's original ID instead of using their own (e.g. as in Shibboleth see `https://spaces.internet2.edu/display/ShibuPortal/Solution+Proposal`)

2.13 Repudiation threats

T131-Repud Repudiation of events by any party (system entity or user).

2.14 Unauditability threats

T141-AltSig Alteration of signed message (see CR27-Sig and CR28-Vfy)

T142-Tamper Alteration of audit trail (see CR212-Trail)

T143-LeakLogs Eavesdropping on logs (see CR212-Trail)

T144-NoAud Insufficient auditing in the first place

2.15 Software bug threats

T151-Overflow Buffer overflow attack, to gain injection and execution of code, usually leading to compromise of the machine. Also heap overflow.

T152-Inject SQL Query Injection attack, causing database engine to execute unauthorized database statements. This threat also covers similar attacks on other databases or downstream services like shell scripts (command injection).

T153-NI Access control, signature verification, or other security feature not implemented. Testing only positive cases can easily ignore this type of threat. It is imperative that the test suites also include negative testing.

T154-Input Input MUST at all times be validated to conform to the expected syntax, and input in general should never be trusted unless there is a cryptographical or structural guarantee of trustworthiness. Some particular perils

a. Explicit inputs
b. Environment variables
c. URL and query string
d. CGI form fields
e. Cookies
f. HTTP headers
g. SOAP headers
h. Processing contexts of all sorts passed between software modules
i. Fields of certificates (from untrusted source, but you would not know until you manipulated the certificate to find out if it is trusted).
j. Metadata
k. Messages from event bus
l. Data coming from database (even if database is supposed to be trusted, it may have been compromised, thus checking for proper format will help to detect the breach and contain it).
m. Deserialization of any data. This is particularly acute whan using eval to deserialize JSON data.

Perl(1) tainted feature provides a way to track untrusted user input. All untrusted input MUST be scrutinized for attack vectors, such as directory climbing (".." or " " = home directory) as a path component. Where relative path is expected, an absolute path - one starting by "/" - MUST NOT be allowed. All *shell(1)* or SQL escape characters (see also T152-Inject) are of highest suspicion until proven to be secure.

When validating input, preferred strategy is to test if input is good and anything that does not pass is bad. The alternative strategy of looking for known bad things (like ".." in path) is much weaker and prone to errors.

T155-Mem Code MUST NOT

a. Dereference a Null Pointer
b. Use Freed Memory

c. Doubly Free Memory

T156-Fmt Format string mismatch. In *printf(3)* format string it is easy to get the format specifiers and actual arguments mixed, resulting inappropriate format specifier being used for a given piece of data.

T157-Trunc Truncation of value. Sometimes truncated value can have different meaning.

T158-Signed Signed conversion of a value that may not have been identified.

T159-CliValid Reliance on client side validation is no good for server as an alternate client will not perform the validation. Server MUST at all times perform all security critical validations. Client side validation has value in (i) improving user experience and feedback, (ii) reducing network traffic by not submitting obviously invalid inputs, (iii) protecting client side processing like AJAX applications. Such applications MUST be suspicious of what is sent to them by server, in particular if they use JSON and plan to use *eval()* for processing it.

T1510-XSiteScript Cross Site Scripting.

T1511-Stack Stack overflow. Similar to T151-Overflow, but specifically applied to the argument and call stack of most compiled programs.

T1512-HTML Using non-validated input in the HTML output stream. This could lead into insertion of JavaScript on the generated page.

T1513-PtrSub Improper Pointer Subtraction

T1514-EqEq Assignment (=) instead of comparison (==).

2.16 Cloud threats

T161-Panopticon The cloud operator may gain excessive knowledge about the operation of the network as a whole, or even individual users, if too many Trust Netowork components and Service Providers are colocated with the same cloud operator.

T162-Corr The Cloud operator may be able to see data of several Service Providers and correlate this data.

T163-CorrAud The Cloud operator may be able to see all of the audit data and correlate it.

2.17 Service Availability Threats

T161-DoS Denial of Service by massive network load.

T162-DNS Poisoning DNS or taking DNS down will prevent legitimate players from communicating with each other.

T163-Expt Using program flaw or exception to trigger excessive resource consumption (spinning process, writing all logs full, and leaking memory) or to cause a service to crash.

T164-GC Feeding service request pattern (e.g. monotonically increasing input size so that previously freed memory is never enough to satisfy the new request) that causes fragmentation of memory, excessive or ineffective garbage collects, or memory leaks.

3 Conclusion

4 Future work

- SPOFs and fat targets: discovery, idmapper, IdP

References

[TAS3BIZ] Sampo Kellomäki, ed.: "TAS3 Business Model", TAS3 Consortium, 2009. Document: draft-sampo-tas3-biz-model-2009-v03.pdf

[TAS3THREAT] Sampo Kellomäki, ed.: "TAS3 Threat Analysis", TAS3 Consortium, 2009. Document: tas3-threats-vXX.pdf

[TAS3ARCH] Sampo Kellomäki, ed.: "TAS3 Architecture", TAS3 Consortium, 2009. Document: tas3-arch-vXX.pdf

[TAS3PROTO] Sampo Kellomäki, ed.: "TAS3 Protocols and Concrete Architecture", TAS3 Consortium, 2009. Document: tas3-proto-vXX.pdf

[TAS3COMPLIANCE] Sampo Kellomäki, ed.: "TAS3 Compliance Requirements", TAS3 Consortium, 2009. Document: tas3-compliance-vXX.pdf

[TAS3GLOS] Quentin Reul (VUB), ed.: "TAS3 Gloassary", TAS3 Consortium, 2009. Document: tas3-glossary-vXX.pdf

[IAF] Russ Cutler, ed.: "Identity Assurance Framework", Liberty Alliance, 2007. File: liberty-identity-assurance-framework-v1.0.pdf (from http://projectliberty.org/liberty/resource_center/papers)

[NIST-SP800-30] Gary Stoneburner, Alice Goguen, and Alexis Feringa: "Risk Management Guide for Information Technology Systems", Recommendations of the National Institute of Standards and Technology, NIST, 2002. http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf

[Meier08] J.D. Meier: "Threats, Attacks, Vulnerabilities, and Countermeasures", 30.3.2008. http://shapingsoftware.com/2008/03/30/threats-attacks-vulnerabilities-and-countermeasures/

[Meier09] J.D. Meier: "Security Hot Spots", 9.3.2009. http://shapingsoftware.com/2009/03/09/security-hot-spots/

[SAML2security] "Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0", Oasis Standard, 15.3.2005, saml-sec-consider-2.0-os

[IDWSFSecPriv] "Liberty ID-WSF Security & Privacy Overview", liberty-idwsf-security-privacy-overview-v1.0.pdf from http://projectliberty.org/resource_center/

Revision History

03 30.11.2008 Sampo

- Added some cloud threats

02 12.4.2009 Sampo

- Integrated into deliverable
- Added references to SAML and IDWSF security analyses

01 11.3.2009 Sampo (sampo@symlabs.com)

- First draft out of blue

Document ID `tas3-threats-v02.pdf`

Repository path `repo.tas3.eu:/var/lib/tas3repo/arch/tas3-threats.pd` (1.16)

```
export CVSROOT=:ext:repo.tas3.eu:/var/lib/tas3repo
cvs co arch
cd arch
# modify tas3-*.pd
cvs ci -m 'What changed...'
```

URL path `https://portal.tas3.eu/arch/review/tas3-threats-v02.pdf`

Commenting

- Please comment on the `TAS3WP02@LISTSERV.CC.KULEUVEN.AC.BE` mailing list, or that failing, send your comments to the editor.
- Any footnotes in this document will not appear in final version. They are editorial comments that may help reviewers to put material in context.