International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1451

(05/2020)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY
Secure applications and services (2) – Security protocols (2)

Risk identification to optimize authentication

Recommendation ITU-T X.1451

International Telecommunication Union

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES (1)	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols (1)	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES (2)	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1319
Smart grid security	X.1330–X.1339
Certified mail	X.1340–X.1349
Internet of things (IoT) security	X.1360–X.1369
Intelligent transportation system (ITS) security	X.1370–X.1389
Distributed ledger technology security	X.1400–X.1429
Distributed ledger technology security	X.1430–X.1449
Security protocols (2)	**X.1450–X.1459**
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589
CLOUD COMPUTING SECURITY	
Overview of cloud computing security	X.1600–X.1601
Cloud computing security design	X.1602–X.1639
Cloud computing security best practices and guidelines	X.1640–X.1659
Cloud computing security implementation	X.1660–X.1679
Other cloud computing security	X.1680–X.1699
QUANTUM COMMUNICATION	
Terminologies	X.1700–X.1701
Quantum random number generator	X.1702–X.1709
Framework of QKDN security	X.1710–X.1711
Security design for QKDN	X.1712–X.1719
Security techniques for QKDN	X.1720–X.1729
DATA SECURITY	
Big Data Security	X.1750–X.1759
5G SECURITY	X.1800–X.1819

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1451

Risk identification to optimize authentication

Summary

Recommendation ITU-T X.1451 specifies a risk identification function in an information and communication technology (ICT) service system as a pre-processor before the authentication function is invoked. It enables the ICT service system to optimize user authentication based on identified risks.

With this specific risk identification function, the ICT service system can make choices on authentication mechanisms adaptively to its users and achieve multiple benefits such as: 1) to improve user experiences; 2) to increase the capacity and reduce the per transaction cost of user authentication; and 3) to reduce the risk of user identity forgery.

For ICT systems, user authentication is a critical security function. Various authentication mechanisms are available, but it may not be clear how to make the best choice from a number of options. Authentication of ICT services should strive to balance multiple functional objectives such as security, user experience, cost and performance.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.1451	2020-05-29	17	11.1002/1000/14252

Keywords

Authentication, risk identification.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics. The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

© ITU 2020

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

Recommendation ITU-T X.1451

Risk identification to optimize authentication

1 Scope

This Recommendation specifies a risk identification function in an information and communication technology (ICT) service system as a pre-processor before the authentication function is invoked. It enables the ICT service system to optimize user authentication based on identified risks.

This Recommendation covers the following topics:

– a reference model of an ICT service system where a risk identification function is introduced in between the core service-handling subsystem and the authentication subsystem;

– a detailed explanation of the component modules of this risk identification function, and how risks are identified, and user authentication is optimized;

– alternative processing designs for the risk identification engine.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

None.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 level of risk [b-ISO Guide 73]: Magnitude of a risk (3.1.4) or combination of risks, expressed in terms of the combination of consequences and their likelihood.

3.1.2 risk evaluation [b-ISO Guide 73]: Process of comparing the results of risk analysis (3.1.5) with risk criteria (3.1.6) to determine whether the risk (3.1.4) and/or its magnitude is acceptable or tolerable.

NOTE – Risk evaluation assists in the decision about risk treatment (3.1.9).

3.1.3 risk identification [b-ISO Guide 73]: Process of finding, recognizing and describing risks (3.1.4).

NOTE 1 – Risk identification involves the identification of risk sources, events (3.1.2), their causes and their potential consequences (3.1.1).

NOTE 2 – Risk identification can involve historical data, theoretical analysis, informed and expert opinions, and stakeholder's needs.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 consequence: Outcome of an event (see clause 3.1.2) affecting objectives.

NOTE 1 – An event can lead to a range of consequences.

NOTE 2 – A consequence can be certain or uncertain and in the context of information security is usually negative.

NOTE 3 – Consequences can be expressed qualitatively or quantitatively.

NOTE 4 – Initial consequences can escalate through knock-on effects.

NOTE 5 – Based on the definition given in [b-ISO Guide 73].

3.2.2 event: Occurrence or change of a particular set of circumstances.

NOTE 1 – An event can be one or more occurrences and can have several causes.

NOTE 2 – An event can consist of something not happening.

NOTE 3 – An event can sometimes be referred to as an "incident" or "accident".

NOTE 4 – Based on the definition given in [b-ISO Guide 73].

3.2.3 risk: Effect of uncertainty on objectives.

NOTE 1 – An effect is a deviation from the expected – positive and/or negative.

NOTE 2 – Objectives can have different aspects (such as financial, health and safety, information security, and environmental goals) and can apply at different levels (such as strategic, organization-wide, project, product and process).

NOTE 3 – Risk is often characterized by reference to potential events and consequences, or a combination of these.

NOTE 4 – Information security Risk is often expressed in terms of a combination of the consequences of an information security event and the associated likelihood of occurrence.

NOTE 5 – Uncertainty is the state, even partial, of deficiency of information related to, understanding or knowledge of, an event, its consequence, or likelihood.

NOTE 6 – Information security risk is associated with the potential that threats will exploit vulnerabilities of an information asset or group of information assets and thereby cause harm to an organization.

NOTE 7 – Based on the definition given in [b-ISO Guide 73].

3.2.4 risk analysis: Process to comprehend the nature of risk and to determine the level of risk.

NOTE 1 – Based on the definition given in [b-ISO Guide 73].

3.2.5 risk criteria: Terms of reference against which the significance of a risk is evaluated.

NOTE 1 – Risk criteria are based on internal and external context, and are regularly reviewed to ensure continued relevance.

NOTE 2 – Risk criteria can be derived from standards, laws and policies.

NOTE 3 – Based on the definition given in [b-ISO Guide 73].

3.2.6 risk treatment: Process to modify risk.

NOTE 1 – Risk treatment can involve:

– avoiding the risk by deciding not to start or continue with the activity that give rise to the risk;
– seeking an opportunity by deciding to start or continue with an activity likely to create or enhance the risk;
– removing the source of the risk;
– changing the nature and magnitude of likelihood;
– changing the consequences;
– sharing the risk with another party or parties; and
– retaining the risk by choice.

NOTE 2 – Risk treatments that deal with negative consequences are sometimes referred to as risk mitigation, risk elimination, risk prevention, risk reduction, risk repression and risk correction.

NOTE 3 – Risk treatment can create new risks or modify existing risks.

NOTE 4 – Based on the definition given in [b-ISO Guide 73].

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart

ICT Information and Communication Technology

ID Identifier

IP Internet Protocol

OTP One-Time Password

PIN Personal Identification Number

SMS Short Message Service

5 Conventions

None.

6 Introduction

For many ICT service systems, user authentication is a critical security function. Various authentication mechanisms are available, but it may not be clear how to make the best choice from a number of options. Authentication of ICT services should strive to balance multiple functional objectives such as security, user experience, cost and performance.

Multi-factor authentication is a successful technology in ICT, but should not be regarded as universal for all cases. Example considerations follow.

– User needs to remember a password, bring a hard token, install a soft certificate, check a mobile phone short message service (SMS), look into a camera, etc. The login and authentication process can be complicated and not user friendly.

– For an ICT service system with both a very large number of users and a high access concurrence, simply applying multi-factor authentication to every single user will impose an extremely large burden on both its authentication subsystem and communication network. This will further increase delay in response and impair user experience, as well as imposing a high authentication cost per transaction.

– Although multi-factor authentication ensures fairly good security for the identification of ICT users, it is still not sufficient for applications related directly to monetary resources, such as digital financial services. Forgery or cloning of authentication credentials, Trojans, phishing websites and ever-changing attack techniques of adversaries are great risks to such services and cannot be solved by multi-factor authentication alone.

Considering these challenges, this Recommendation introduces a risk identification function to optimize user authentication.

Traditionally, an authentication subsystem is closely coupled with the core service-handling subsystem and is invoked according to fixed rules when service requests are received. This Recommendation recommends that the core service-handling subsystem and the authentication subsystem be decoupled and a risk identification function (called a "risk identification subsystem") inserted between them. This helps to make authentication-related risk decisions before the authentication subsystem is invoked.

With this specific risk identification function, the ICT service system can adapt choices of authentication mechanisms to its users and achieve the following benefits:

1) to improve user experience;

2) to increase the capacity and reduce the per transaction cost of authentication; and

3) to reduce the risk of user identity forgery.

7 Reference model of an ICT service system with risk identification

A reference model of an ICT service system with risk identification to optimize user authentication is shown as Figure 1.

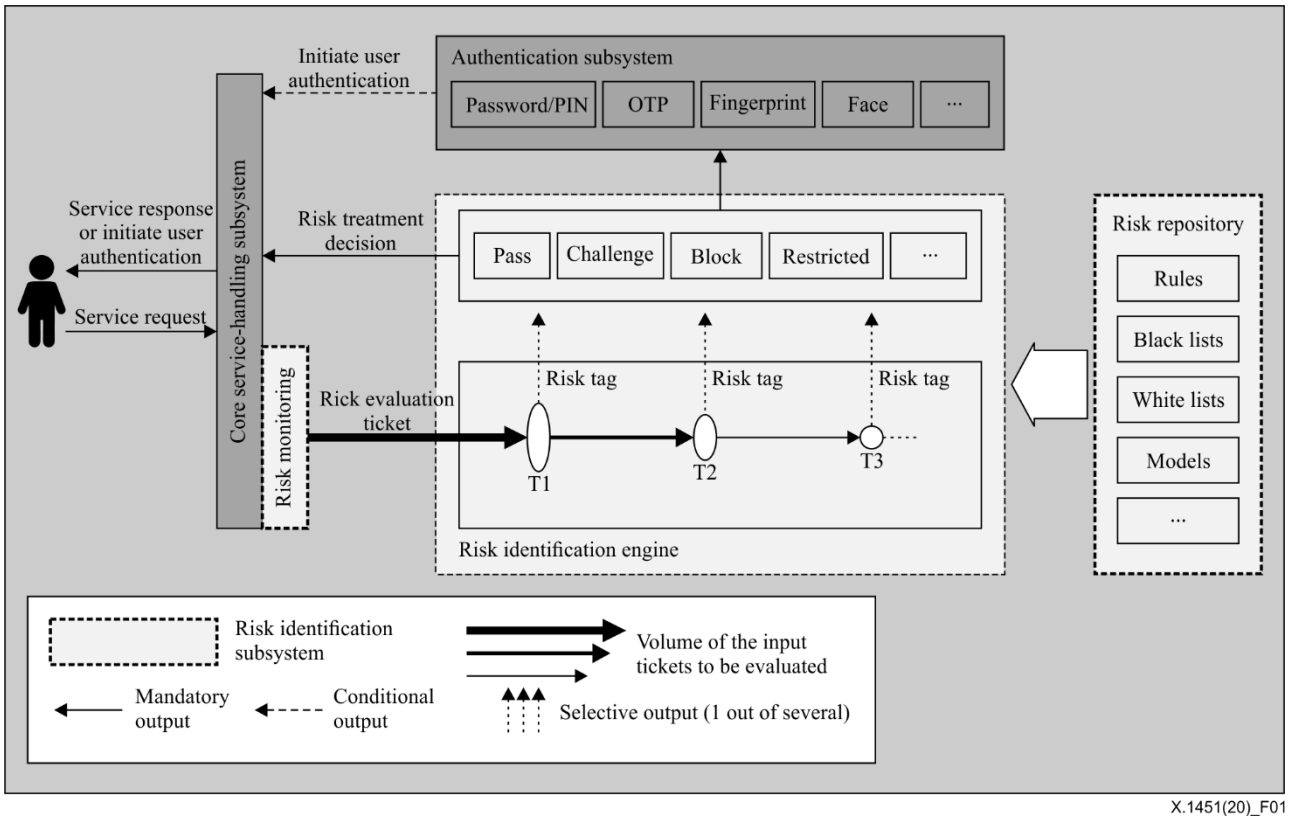

Figure 1 – Reference model of an ICT service system with risk identification to optimize user authentication

In this reference model, the risk identification subsystem is shown as a new function introduced into a traditional ICT service system.

The risk identification subsystem composes the following components, shown in boxes with dashed borders in Figure 1.

– The **risk-monitoring module** resides in the core service-handling subsystem side. It monitors and collects information relevant to risk identification when the core service-handling subsystem receives a service request from the user. The relevant information includes, but is not limited to, information about the user device, user behaviour and user account, and information about the service request itself. All information collected by the risk-monitoring module regarding a given service request should be passed to the risk identification engine in real time.

– The **risk identification engine** is located between the core service-handling and the authentication subsystems. It identifies and evaluates the risk(s) associated with a given service request based on the risk evaluation ticket fed by the risk-monitoring module and the

knowledge provided by the risk repository module. It returns a risk treatment decision to the core service-handling subsystem.

– The **risk repository module** supports the risk identification engine.It contains risk identification resources, e.g., the blacklist or whitelists, rules and models that are used by the risk identification engine to perform its function. These lists, rules and models may differ for different services, and in different jurisdictions. They may be updated manually by configuration or automatically using technologies such as machine learning.

In this reference model with risk identification subsystem, the authentication subsystem is only invoked when the risk identification engine decides that a service request shall not be processed unless a preceding user authentication is fulfilled.

8 Functional components of risk identification subsystem

8.1 Risk-monitoring module

The risk-monitoring module monitors and collects information relevant to risk identification when the core service-handling platform receives a service request from the user, also called an event. All information collected by the risk-monitoring module regarding a given service request, called a risk evaluation ticket, should be passed to the risk identification engine in real time.

The risk identification engine should pre-define a number of templates for different types of events in a given ICT service system, as depicted in Figure 2. The risk-monitoring module should create a risk evaluation ticket for each event using the appropriate template in real time.

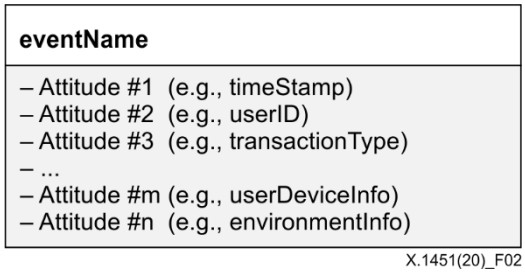

Figure 2 – Risk evaluation ticket template

The relevant information includes, but is not limited to, information about the user device, user behaviour, user account and information about the event itself.

8.2 Risk repository module

The risk repository module contains blacklists or whitelists, rules and models that are used by the risk identification engine to perform its function.

Known risks are organized into types, as shown in Figure 3. Different services may define differing sets of risk types. For example, typical risk types of a mobile payment service may include, but are not limited to, theft risks, fraud risks and operation risks, as described in Appendix II.

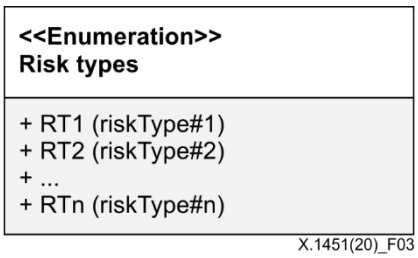

Figure 3 – Risk types

Each risk type has its own risk strategy, which is composed of a set of risk identification rules. A risk identification rule is designed to identify a certain level and type of risk in an event. Each rule contains a single condition or a set of conditions. A condition is the combination of a variable, a constant and a comparison operator between them.

A rule is satisfied only if all its conditions are met. A whitelist rule is designed to identify trustworthy events. In contrast, a blacklist rule is designed to identify risky events. The risk strategy for a specific risk type may contain either blacklist or whitelist rules or both.

This relationship is depicted in Figure 4.

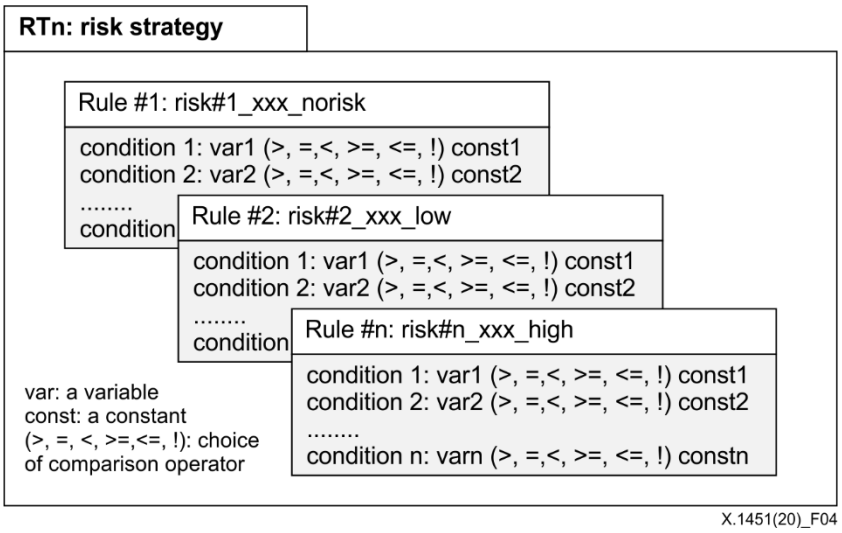

Figure 4 – Risk strategy, risk identification rules and conditions

A variable is the digital notation of a piece of characteristic information about an event, which is fed into the risk identification engine by the risk-monitoring module. There are three types of variables, described as follows.

1) Raw variable: a piece of characteristic information that can be directly obtained from the event, e.g., the age of the payer who initiates a payment request or the device type (PC, mobile phone, etc.) on which the request is initiated. Each raw variable represents an attribute in the risk evaluation ticket (see Figure 2).

2) Tag variable: a qualitative value that is derived from a raw variable, e.g, the positive or negative result of checking a raw variable against the blacklist or whitelist(s).

3) Model score variable: a quantitative value that is the output of a risk identification model. A model may take several raw or tag variables as its input.

Figure 5 illustrates the three variable types, and their relationship with the condition, rule and risk strategy.

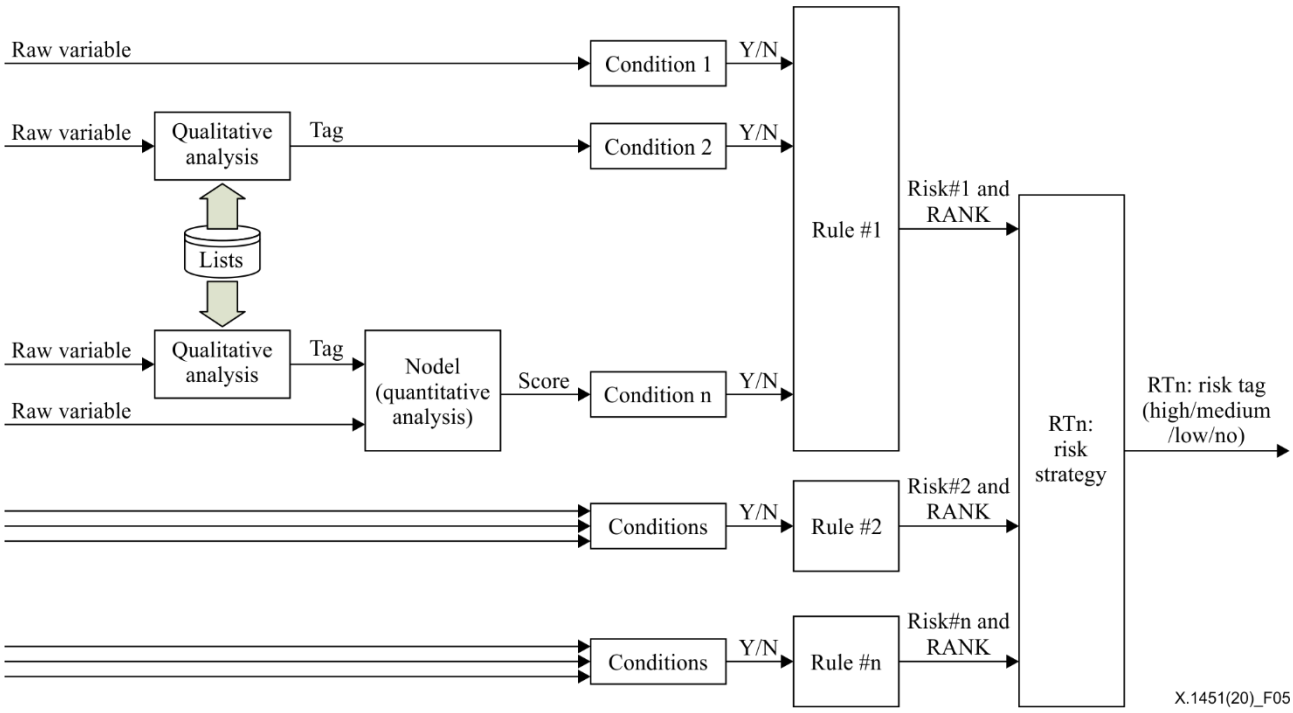

Figure 5 – Variable types and their relationship with the condition, rule and risk strategy

The risk repository module contains risk strategies for all the known risk types, including all the rules, together with all the blacklists, whitelists and models that are used by these rules.

To cope with the endless battle of attacking and defending, the risk repository should be updated regularly to reflect the most up-to-date risk landscape of the concerned service. It may be updated manually by configuration or automatically using technologies such as machine learning.

NOTE – The rules, lists and models may differ for different services and in different jurisdictions. The process of developing risk identification rules, lists and models for a specific service in a specific jurisdiction lies outside the scope of this Recommendation.

8.3 Risk identification engine

8.3.1 Multi-tier risk identification processing

It is recommended that the risk identification process performed in the risk identification engine be arranged into multiple tiers to achieve a balance between security and convenience, cost and performance (See Figure 1).

More details about multi-tier processing are provided in clauses 8.3.1.1 and 8.3.1.2. Further, Appendix I describes a use case that demonstrates the multi-tier process in the login scenario.

8.3.1.1 Division of tiers

As described in clause 8.2, risk identification rules are organized into different risk types. In each risk type, the rules are further divided into several tiers when assigned to the risk identification engine based on the basic principles described as follows.

An event that can satisfy a whitelist rule in a given risk type should be regarded as free of such risks. An event that can satisfy a whitelist rule in all the known risk types should be regarded as reliable and trustworthy. To reduce the delay in service response, requests in such events should be released immediately from the risk identification engine, and the core service-handling subsystem should go on processing without additional user authentication. Therefore, the first tier should follow the following principles.

– **Principle D.I**: the first tier should focus on identifying reliable and trustworthy events.

– **Principle D.II**: each of the known risk types should assign all whitelist rules to the first tier.

– **Principle D.III**: each of the known risk types should assign at least one rule in the risk strategy to the first tier to avoid risk omissions.

– **Principle D.IV**: the first tier should always be a synchronous tier whose conclusion has to be delivered before a service request can be released from the risk identification engine.

The remaining tiers should follow the following principles.

– **Principle D.V**: the second and subsequent tier(s), when they exist, should focus on identifying real risks and contain only blacklist rules.

– **Principle D.VI**: the number of synchronous tiers should be restricted to avoid causing unreasonable delay to the service response.

– **Principle D.VII**: there might be zero or more asynchronous tier(s), whose conclusion might be delivered after a service request is released and processed.

– **Principle D.VIII**: if some of the rules would cause significant delay (exceeding a certain threshold) to the service request, they should be assigned to an asynchronous tier. Otherwise, they can be assigned to a synchronous tier.

Based on the assumption that most events in a healthy system should be reliable and trustworthy, there is a general principle as follows for the division of tiers.

– **Principle D.IX**: rules in each of the known risk types should be assigned to different tiers in such a way that most events can hit a rule in each type within the first tier, and only a small proportion of the events has to enter the subsequent tier(s).

NOTE – This principle is illustrated in Figure 1 with input arrows becoming less bold from T1 to T3, and dotted risk tag output arrows. Further explanation can be found in Appendix I.

For example, a well-optimized risk identification engine can filter more than 95% of events with its first tier and only allow fewer than 0.1% of events into the last asynchronous tier.

8.3.1.2 Aggregating risk identification results from multi-tier processing

Risk identification rules are the core of the risk identification engine. These rules are organized into different risk types in the risk repository as mentioned in clause 8.2, and further divided into several tiers when assigned to the risk identification engine as described in clause 8.3.1.

When an event (abstracted by a set of variables in the risk evaluation ticket) enters the risk identification engine, it will be checked against the rules tier by tier in each of the known risk types.

Each risk type should define, according to its own risk strategy, how to aggregate the results from different rules into a risk tag, represented by "RTn: risk tag" in Figure 5. A risk tag associated with a risk type is an indication of the evaluated risk level of a service request in the risk type concerned.

For example, the following principles can be applied when aggregating the risk identification results in the same risk type.

– **Principle A.I**: in the same risk type, rules could run in parallel or in series according to a pre-defined routing mechanism.

– **Principle A.II**: in the same risk type, whitelist rules should have a higher priority than blacklist rules. Once the event hits a whitelist rule, its risk tag associated with this risk type should be set to “no”.

– **Principle A.III**: if the event hits more than one blacklist rule, which indicates different risk levels (high/medium/low), its risk tag associated with this risk type should be set to the highest level.

– **Principle A.IV**: if the event hits no rule at all, which means it is not trustworthy, but the risk is unknown, its risk tag associated with this risk type should be set to "no".

The risk identification engine should also define how to aggregate the risk tags of all the known risk types into an overall risk tag, represented by "RT_all: risk tag" in Figure 6, in order to make a final treatment decision.

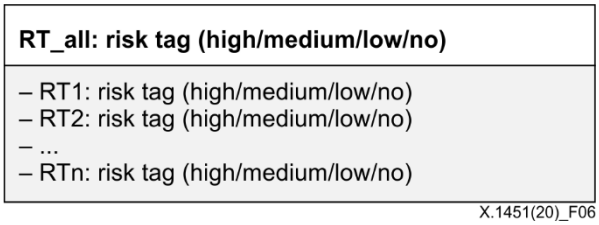

Figure 6 – Risk tags

The principles of cross-type risk aggregation should be as follows.

– **Principle A.V**: rules in different risk types should run in parallel. A hit in one risk type should not stop the analysis of other types.

– **Principle A.VI**: the aggregated risk tag should be set to the highest level of the risk tags associated with all the known risk types.

A risk tag is a short string generated from the risk identification engine to indicate the evaluated level of risk associated with a given event. A risk tag should always be associated with a context, e.g., a specific risk type, or an overall decision considering all the known risk types.

8.3.2 Risk treatment decision

As described in clause 8.3, when a risk evaluation ticket goes through the multi-tier risk identification process, an overall risk tag will be attached to it. This risk tag indicates whether the risk associated with this event is high, medium, low or zero risk. The next step is to make a decision on how to mitigate the identified risks.

An ICT service system may define its own risk treatment options, as shown, for example, in Figure 7:

– pass: the service request should be processed immediately;

– warning: the service request should be processed immediately, but the user should be warned of potential risk(s);

– block: the service request should be refused immediately, and the user should be warned of the identified risk;

– restricted: not only should this service request be refused immediately, but also any future operations should be restricted until certain conditions are met;

– challenge: user authentication should be initiated before the service request can be processed further.

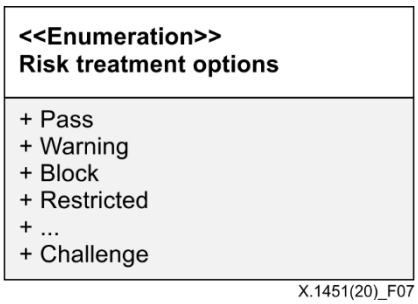

Figure 7 – Example risk treatment options

The risk treatment decision should be made by the risk identification engine based on the attached risk tag, taking into consideration several other factors, e.g., service rules, regulatory requirements, user experiences and system performance, as shown in Figure 8.

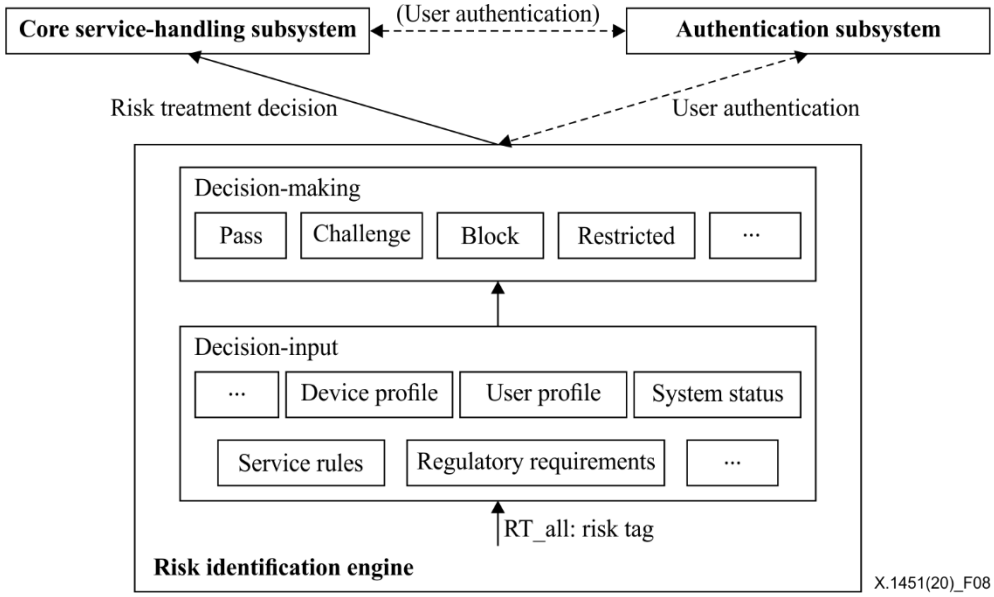

Figure 8 – Risk treatment decision process

Some of the treatment options can be executed directly by the core service-handling subsystem, e.g., pass, block or restricted. If the decision is challenge, the risk identification engine should contact the authentication subsystem, fetch the authentication parameters (or a pointer or link to the parameters) and return to the core service-handling subsystem together with the risk treatment decision.

NOTE – Figure 1 and Figure 8 illustrate this with a solid arrow from the risk identification engine to the core service-handling subsystem, and two dashed arrows between the risk identification engine, the authentication subsystem and the core service-handling subsystem.

9 Authentication subsystem

It is the authentication subsystem's responsibility to choose appropriate authentication mechanism(s) for a service request which needs a preceding challenge treatment.

The authentication subsystem manages a collection of available authentication mechanisms, e.g., as shown in Figure 9.

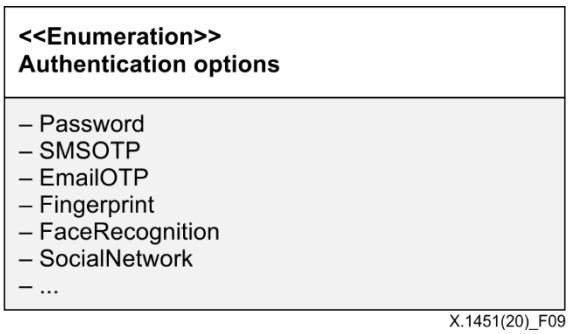

Figure 9 – Authentication options

It is recommended that the authentication subsystem provide configurable policies to match the available authentication mechanisms with various authentication scenarios, as in Figure 10b), rather than develop a collection of authentication mechanisms for each scenario, as in Figure 10a).

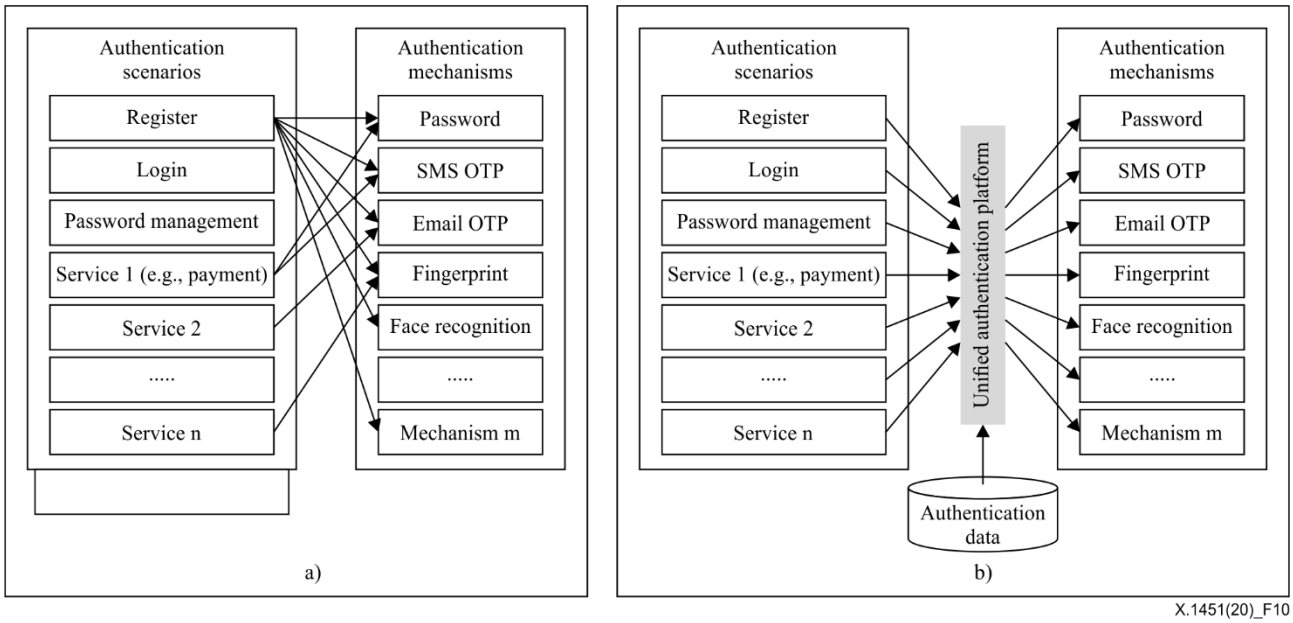

Figure 10 – Authentication scenarios and mechanisms

In response to the request from the risk identification engine, the authentication subsystem may return multiple candidate authentication options. Either the risk identification engine or the core service-handling subsystem can further choose one or more of them, which will eventually be presented to the user.

10 Alternative processing designs for risk identification engine

In clause 8.3.1, the risk identification engine adopts a multi-tier processing design, which is one way to produce risk evaluation results. This clause explores an alternative processing design for the risk identification engine, which belongs to the security rating systems category.

A risk-rating system is a function of an ICT security solution whose purpose is to provide a verdict, e.g., an internet proxy will need to take decisions based on the verdict of Internet protocol (IP) addresses, a range of IP addresses, an internet domain, a URL, etc.

In this Recommendation, the risk identification engine needs to take decisions based on a risk evaluation ticket.

Risk rating can be based on algorithms and strategies that can be described in the context of voting systems in mathematics. Voting systems are defined by a collective choice operator, which is the algorithm to aggregate the final verdict. As there is no perfect collective choice operator, they will:

– impact performance positively or negatively;

– create more or less false negatives;

– create more or less false positives; and

– introduce arbitrariness.

Therefore, there is a permanent quest to find the best possible collective choice operator. See Appendix III for an interpretation of the multi-tier processing design presented in this Recommendation.

Annex A

Non-functional design considerations

(This annex forms an integral part of this Recommendation.)

This annex provides non-functional considerations for designers to deliver a complete solution based on this Recommendation.

A design is a specification for the construction of a system. A criterion is a principle by which something may be judged or decided. A design criterion is a criterion that applies to a given design.

Designers should consider the criteria in this Annex in order to ensure the solution can effectively be put in production.

A.1 Non-functional design criteria

The solution should meet at minimum the non-functional design criteria listed in Table A.1.

Table A.1 – Non-functional design criteria

Non-functional design criteria		Description
Stability		The resistance of the solution to environment changes. It can be broken down into more detailed criteria: scalability, high availability, resiliency and performance
	Scalability	The capacity of the solution to adapt to increased workload
	High availability	The capacity of the solution to adapt to system failures within its data centre premise
	Resiliency	The capacity of the solution to adapt to overall data centre failures and major outages
	Performance	The capacity of the solution to deliver maximum throughput
Security		The resistance of the solution to security attacks
Flexibility		The long-term stability of the design of the solution against new functional requirements
Ease of integration		The ease with which a solution integrates itself into its surrounding environment
Manageability		The capacity of the solution to be managed. For example, it can include configurability, operational management, lifecycle management, etc.
Auditability		The capacity of the solution to be audited

A.2 Stability

The solution in the production environment should meet the criteria listed in Table A.2.

Table A.2 – Stability criteria

Criteria	Consideration
Scalability	The risk identification engine instances should not impose any constraints on scalability and scalability should be offered by the underlying technical infrastructure in any scenario (physical servers, cloud platform, infrastructure as a service, etc.)
High availability	Implementation should consider both cases when the risk identification engine instances are state-full or state-less. From this analysis, the implementation should establish the appropriate high availability strategy, assuming all surrounding components are themselves highly available, in particular on the risk repository side (e.g., databases, directory services and big data)
Resiliency	The risk identification engine instances should not impose any additional constraints on infrastructure resiliency plans between data centres.
Performance	The risk identification engine instance as described in this Recommendation is designed to optimize the performance through its multi-tier processing. There are other designs as described in clause 10

A.3 Security

The security of the risk identification engine shall be ensured.

Should the risk identification engine be compromised, the impact might be even bigger than the core service-handling subsystem being compromised.

A.4 Flexibility

The risk identification engine architecture flexibility is measured by its over-time stability and should not require architect re-designing or architecture changes should be minimal. The implementation should respect the flexibility that is inherent in this Recommendation.

For example, as security architectures move to a zero-trust concept, the need for flexibility lies in the natural involvement of such a solution in a bigger integrated cyber defence strategy of the calling platform that could both enrich the risk identification engine resources, get information from the risk engine itself or make it part of a bigger orchestration strategy. Should it be the case, risk identification will need to be extended with the right interfaces exhibiting the specific standards and protocols required.

A.5 Ease of integration

The implementation of a risk identification engine should optimize its capacity to be easily integrated into its surrounding environment, which may consist, for example, of management systems, monitoring systems and auditing systems. The self-contained nature of the risk identification engine helps to meet these criteria.

A.6 Manageability

A.6.1 General

Each risk identification engine instance can be deployed with its own management capabilities that are left to implementation choices. However, in large deployments of implementations, risk identification engine instances will be managed through an external management platform as part of a larger supporting platform.

In this case, it is recommended that a risk identification supporting platform be placed beside the risk identification engine and the risk repository module. The risk identification supporting platform should at least contain the following functions illustrated in Figure A.1:

– risk configuration manager;

– risk strategy manager;
– catalogue of risk evaluation processors.

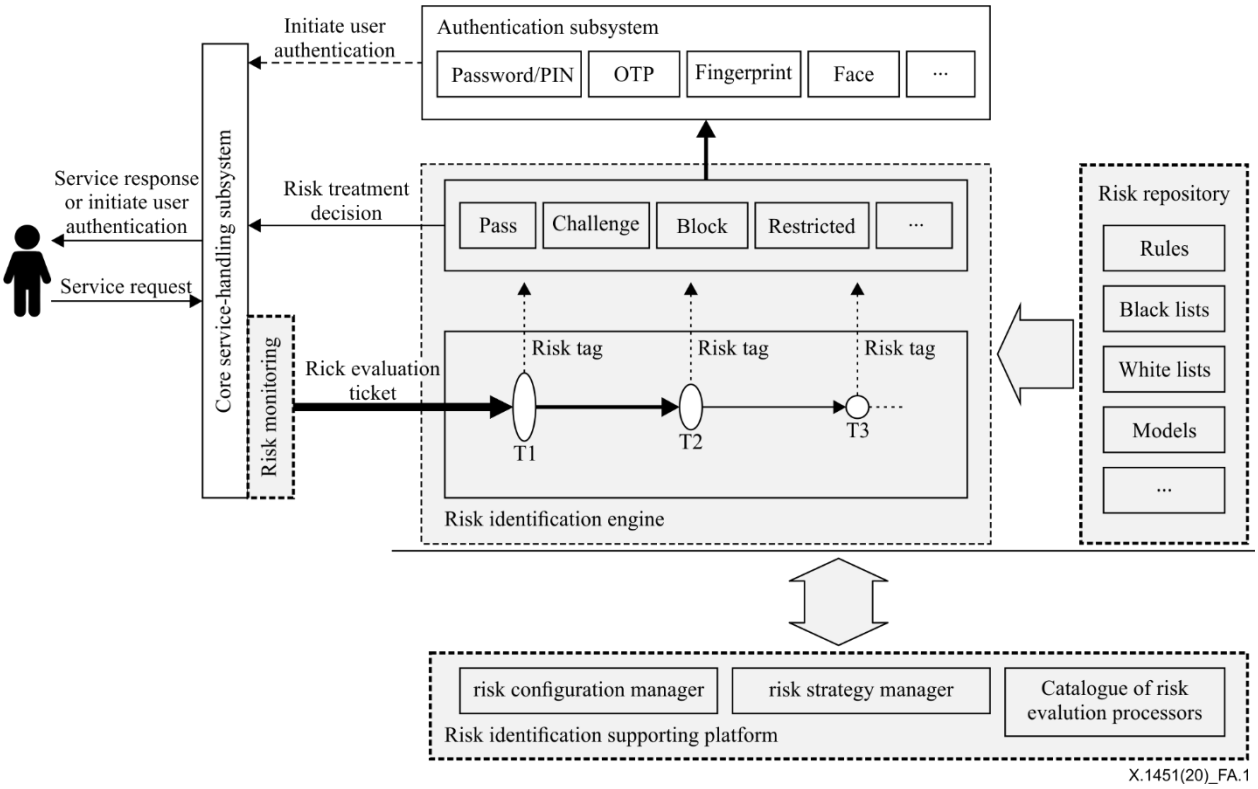

Figure A.1 – Extended model with a risk identification supporting platform

A.6.2 Risk configuration manager module

This module is responsible for the parameterization of the risk identification engine and its interfaces with external modules.

A.6.2.1 Constituencies that should be supported

This module should contain the following.

– The private configuration of the entire module (IP addresses, service names, administration accounts, risk operators, risk evaluation processor, etc.).
– The configuration of the external modules: risk monitoring; core service-handling subsystem; authentication subsystem and risk repository. In particular, it contains the public configuration of each risk identification resource.

A.6.2.2 Alternative risk evaluation processor configurations

There are alternatives to optimize other aspects than just performance. An implementation offering a risk evaluation processor configuration should allow more flexibility to optimize the risk identification engine against:

– performance;
– false positives;
– false negatives.

A.6.3 Risk strategy manager module

This module contains the library of potential risk strategies as defined in clause 8.

For example, this risk strategy manager module could lead to the establishment of an entire strategy manager language, allowing very rich strategies to be put in place.

A.6.4 Catalogue of risk evaluation processors

The implementation may offer templates or profiles to describe the possible risk evaluation processors and the parameters that can be used by the risk identification engine.

A.6.5 Compilation considerations

The supporting platform may therefore compile the outcomes of both the risk configuration manager module and the risk policy manager module to fit the runtime-oriented design and detailed configuration, as well as the policy implementation of the risk evaluation engine.

A.7 Auditability

The implementation may need to fulfil auditing requirements, which may imply, for example, that all the changes in the strategy and configuration manager modules are not only logged, but also their integrity shall be protected.

Appendix I

Use case: Risk identification to optimize login authentication

(This appendix does not form an integral part of this Recommendation.)

In this example, the login scenario is used to demonstrate how the multi-tier risk identification is applied to enhance the security and convenience of user authentication.

Tier 1: Edge control

T1 is used to identify the risks that can be decided, based on the information collected by the client software. T1 processing should be real time.

When the user logs in using the client software, the client software collects information, such as user account and device identifier (ID), and forwards these data to T1 with the login request. T1 checks the user account and device ID against the blacklist or whitelist that are provided by the risk repository module. If the user account or device ID is in the blacklist, T1 will return a high-risk tag and the risk identification engine will suggest that the service platform block this login request directly or that the authentication system challenge the user based on a strict multi-factor authentication mechanism. Otherwise, if the user account or device ID is in the whitelist, T1 will return a low risk tag and the risk identification engine will suggest that the service platform let through this login request directly or that the authentication system challenge the user, based on a simple authentication mechanism such as a static password or personal identification number (PIN).

In other cases, when T1 cannot decide on the risk level based on the information collected by the client software, the login request will be passed on to the next tier for further investigation.

See Figure I.1.

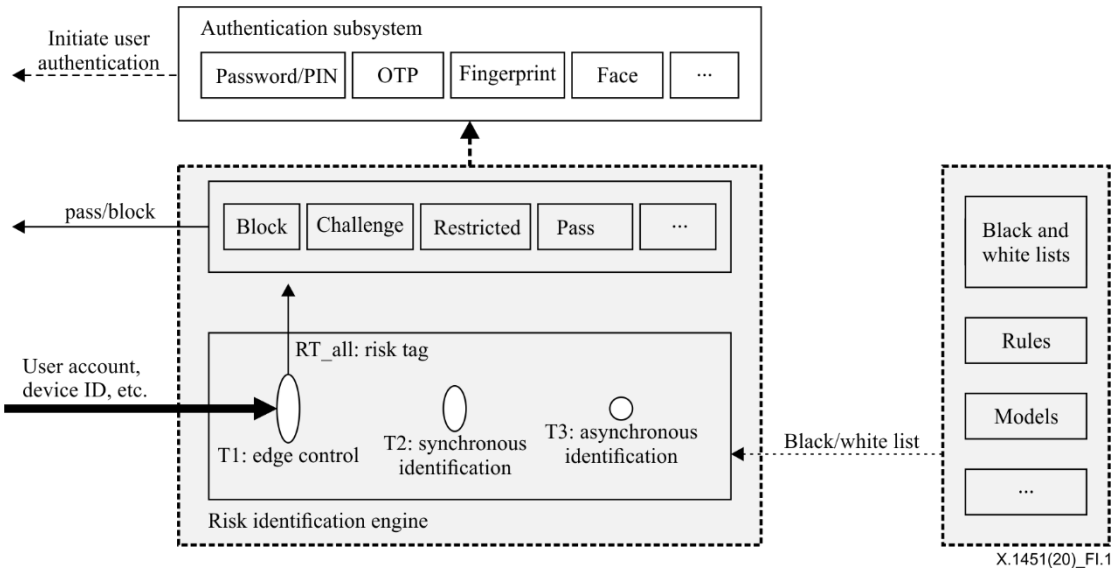

Figure I.1 – T1: Edge control

Tier 2: Synchronous risk identification

T2 is used for further risk identification, if T1 cannot make a decision. Compared to T1, this tier collects more information about the request, runs more rules and models to identify potential risks and returns disposal suggestions to the service platform and authentication system. T2 processing should be real time.

When receiving the login request released by T1, T2 acquires the user account, device ID and additional information, such as user location (if authorized by the user) and time of previous login requests, and checks these against the T2 rules and models.

For example, if the user is logging in on their usual device in their usual location, there may be a T2 rule to attach a low risk tag and the risk identification engine will suggest that the service platform let through this login request directly or that the authentication system challenge the user based on a very simple authentication mechanism, such as a static password or PIN. Otherwise, if the user is using a different device or is logging in too frequently, the risk tag will be medium and the user may be prompted to enter an SMS one-time password (OTP) or answer a completely automated public Turing test to tell computers and humans apart (CAPTCHA) challenge in addition to a static password or PIN.

In other cases, when T2 cannot decide on the risk level after running all the T2 rules and models on the login request, T2 will attach a risk tag of "unknown" and activate the authentication system to initiate a default authentication mechanism. These suspicious login requests will also be passed on to the next tier for asynchronous investigation.

See Figure I.2.

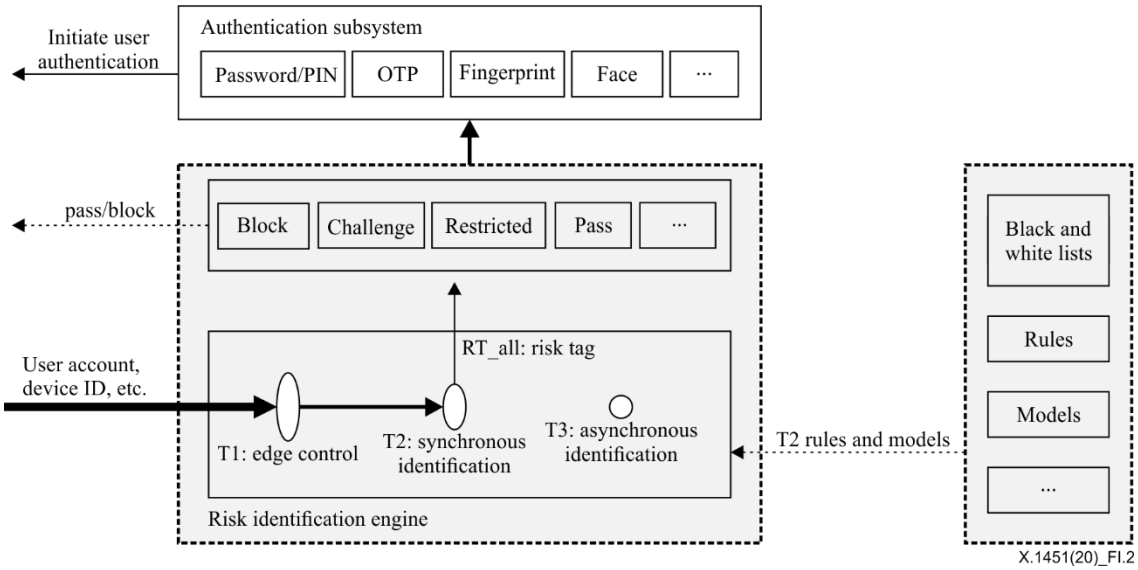

Figure I.2 – T2: Synchronous risk identification

Since the user has to wait for the outcome from T2 before the request can be processed, if the number of T2 rules and models is too large, the T2 rules and models could be further divided into several subtiers, e.g.:

– T2-1: A set of simple rules and models for fast identification, which can make fast decision for more than 95% requests released by T1;

– T2-2: A set of more time-consuming rules and models for deep identification.

Whether and how to divide sub-tiers for T2 is up to the service requirements and can be optimized dynamically in practice. In this simple use case, T2 is not further divided.

Tier 3: Asynchronous risk identification

With a carefully designed and continuously maintained risk repository, T1 and T2 should be able to identify most known patterns of normal and malicious login requests. The more insidious malicious login requests not identified by T1 and T2, which should be very rare, will be processed by T3 asynchronously or even offline, i.e., T3 processing is not real time and the user can proceed without waiting for the T3 analysis result.

When receiving a login request released by T1 and T2, T3 may collect more extensive information, such as the user's subsequent actions, and feeds them into a "high risk operation sequence" model. T3 will return a risk tag based on the outcome of the model. If the risk is high, the risk identification engine may suggest remedial measures. Examples of each follow.

– An example of a "high risk operation sequence" might be: Login; checking account balance; transferring money to some other account(s); remaining balance is zero or very low; logout or timeout; never login again. This kind of sequence may be followed by a genuine owner if that person decides to close this account, but it may also indicate that the contents of the account have been stolen.

– An example of remedial measures might be: Make a phone call to the account owner; business roll-back; start a forensic process; or make compensation.

See Figure I.3.

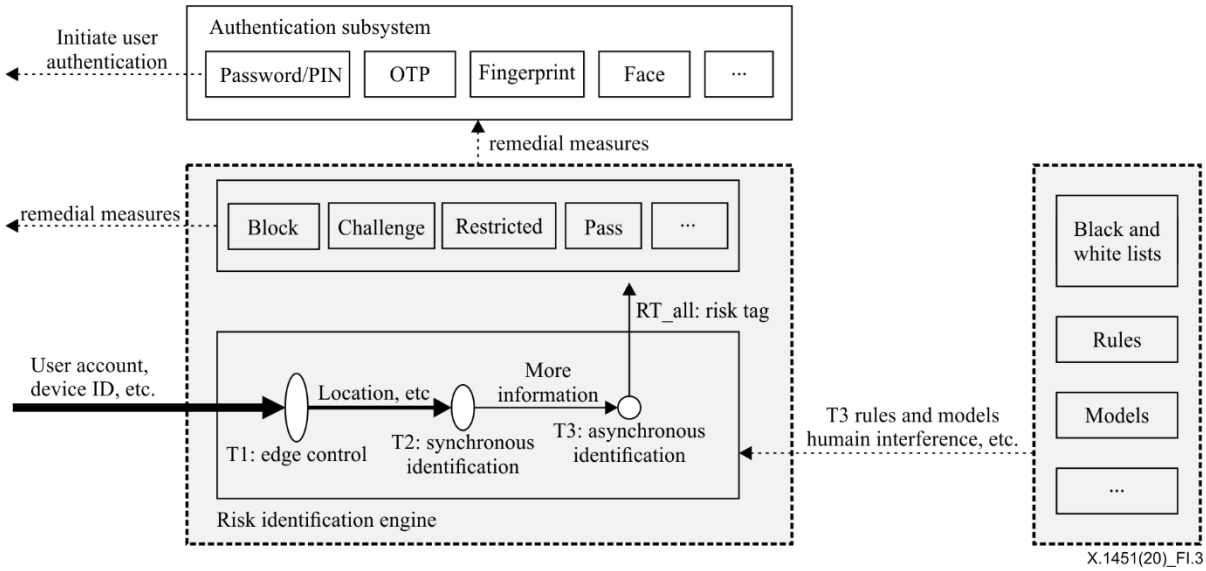

Figure I.3 – T3: Asynchronous risk identification

User complaints, e.g., concerning stolen account contents, may also activate T3 processing of a previous login request, no matter which risk tag was attached to that request. T3 processing is not just rules and models, but may also involve human intervention, such as phone calls and interviews.

Appendix II

A risk repository example for mobile payment system

(This appendix does not form an integral part of this Recommendation.)

The structure of an example risk repository for a mobile payment system is presented to illustrate the risk landscape of a real implementation.

It should be noted that this risk repository never ceases to evolve as the result of the ever-lasting battle between attack and defence.

Risk types

The risks associated with a mobile payment system can be organized into three major risk types: theft risks; fraud risks; and operation risks, as in Figure II.1.

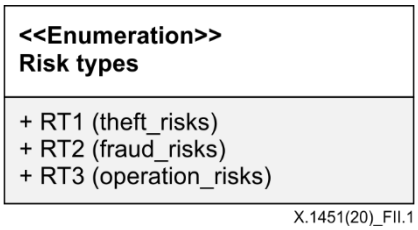

Figure II.1 – Risk types

Theft risks refer to cases in which an unauthorized entity directly withdraws or transfers money from a user's account. The key characteristic of this type is that the risky behaviours are not carried out by users themselves.
Fraud risks refer to cases in which adversaries adopt deceptive identities or present bait to entrap a user to make a transfer to or deposit money in a specific account. The key characteristic of this type is that risky behaviours are carried out by users themselves, but do not reflect their real intentions.

Operation risks refer to cases in which adversaries trick mobile payment service providers or related financial institutions for unjustified enrichment by means of dishonest or even illegal transactions.

Risk strategies

In each of the risk types, there are a number of rules to identify potential risks originating for various reasons. For example, the reasons for theft risks include but are not limited to the following.

– Lost mobile phone: A lost mobile phone may disclose private information about the owner, and result in, for example, a stolen payment account.
– Reused mobile phone number: A deregistered mobile phone number may be recycled by the mobile operator and assigned to another customer; there is a risk if the original owner has linked this mobile phone number to a payment account, but failed to delete the link before deregistering.
– Trojans: A user may unconsciously install a Trojan, which may steal the user's password or even control their device.
– Accounts: Attackers may obtain user data, such as email addresses or mobile phone numbers, by compromising vulnerable websites in order to find payment accounts associated with these data, and then attempt to reset the passwords for these payment accounts. In an extreme case where a user registers the same pair of username and password on a vulnerable website and in a payment system, the payment account can be easily controlled by a hacker.

– Phishing: Adversaries may induce users to click on a phishing link and input information about their bank account, identification card, username and password, etc. These types of private information may result in a payment stolen from the account.
– Social engineering: Adversaries may deceive users to divulge their private information. This private information may result in a payment stolen from the account.

In a real implementation, many variants of such risks arise depending on the payment scenarios, and the rules to identify them need to be carefully designed, regularly maintained and updated. All of these rules together comprise the strategy for theft risks, as shown in Figure II.2.

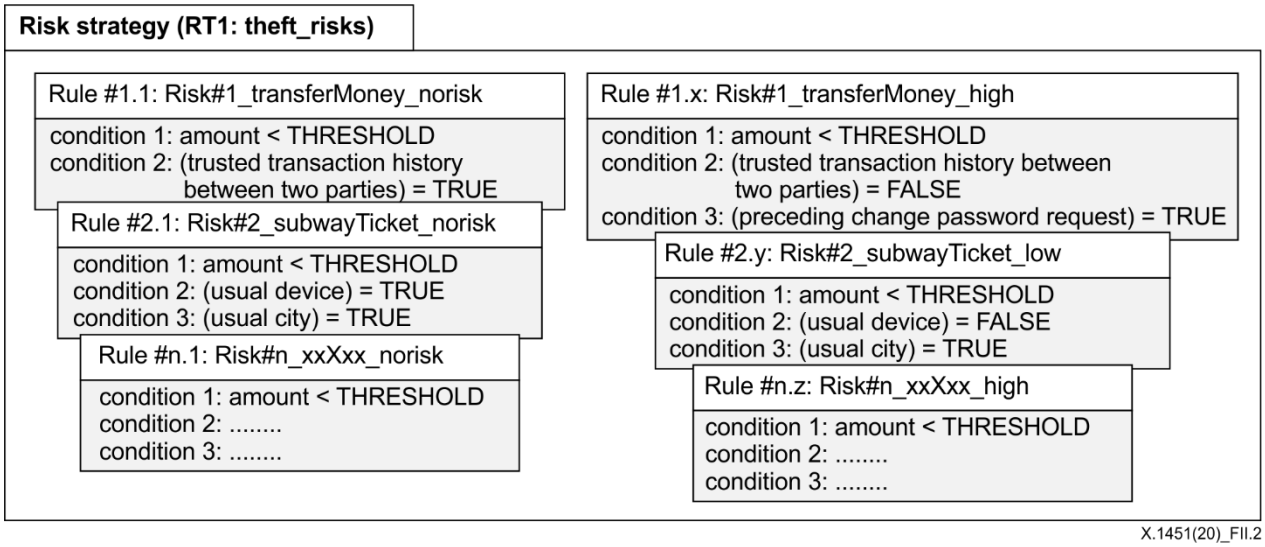

Figure II.2 – Risk strategy for RT1

Similarly, the reasons for fraud risks include, but are not limited to:

– telecom and network fraud: Adversaries may use telephone or instant messaging tools to adopt deceptive identities such as government authorities, acquaintances and customer service executives, and deceive a user into transferring money to or depositing it in a specific account;
– illegal fund-raising: Adversaries may present bait, such as high interest returns, to lure users into transferring funds, and then misappropriate the money collected;
– impersonation account opening: Adversaries may use others' ID cards, bank cards, business licenses, etc., to open payment accounts.

Finally, the reasons for operation risks include, but are not limited to:

– cheating in sales promotions: Adversaries may use technical means to accumulate coupons, loyalty points, etc., in order to obtain illegal income from them;
– payment accounts: Adversaries may illegally use payment accounts for fraud, credit card cashing, money laundering, gambling, disposal of stolen goods, etc.

Fraud risk type and operation risk type have their own respective risk strategy, just like that for theft risk type in Figure II.2.

Appendix III

A mathematical interpretation of a multi-tier processing design

(This appendix does not form an integral part of this Recommendation.)

The multi-tier processing design can be interpreted in collective choice operators as follows.

- The first tier of processing:
  - This is a first voter with a veto right on any subsequent voter.
  - The intention of this first voter is to focus its attention on the fastest obtainable verdicts, which often means the "obvious choices".
  - A famous voter candidate here is the Pareto rule.
  - If the result of this voter is ambiguous, it will allow the vote to continue to other voters, in our case the second tier.
- The second tier of processing:
  - This is a second voter, also with a veto right on any subsequent voter.
  - The intention of this second voter is to focus its attention on more refined analysis looking deeper at more information.
  - If the result of this voter is ambiguous, it will allow the vote to continue to other voters, in our case a third tier.
- The third tier of processing:
  - This is the third and last voter that can use more time and resources to compute a score.

So the design proposed in this Recommendation uses a very specific collective choice operator meant to have a compromise between:

- computation (and therefore speed and cost as per granularity design criteria); and
- precision (false positives and false negatives); with
- no censorship intentions.

Implementations can consider many alternative processing strategies with a vast spectrum of possibilities, e.g., processing in parallel vs. in series, specific known collective choice operators among the many classes of operators found since the problem started to be heavily studied after 1945.

Bibliography

[b-ISO Guide 73] ISO Guide 73:2009, *Risk management – Vocabulary*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2020