Overview of the eGovernment Web Application Security Architecture (WASA) Framework

Rünno Reinu
Lea Hriciková
Uuno Valner

The Challenge

More e-services, higher level of digitalization

means

higher security needs for web applications

The Challenge

How to protect web applications in the context of Enterprise Architecture and Enterprise Security Architecture?

Goal and Deliverables (1)

1. Create a simplistic descriptive guidance (framework):
   - Applying SABSA methodology
   - Containing top-down architectural layers/views
   - To identify technical security requirements for developing web applications

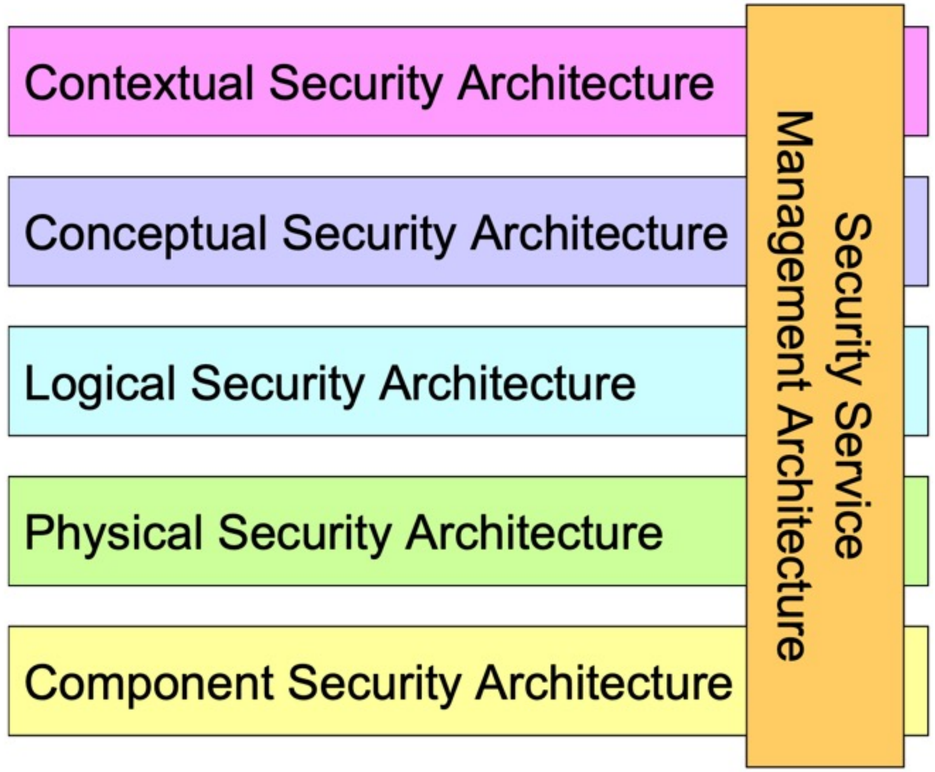
Goal and Deliverables (2)

2. Create a separate tool/checklist which:

- Lists all technical security requirements for web applications
- Helps to select security requirements considering the business value of the information processed by the application

Methodology: SABSA + ASVS

- SABSA as the security architecture methodology
- OWASP ASVS (Application Security Verification Standard) as the methodology of technical requirements for web apps
- Customized for Uganda eGovernment



Contextual Architecture

- What? Why? How? Who? Where? When?
- Risk Assessment

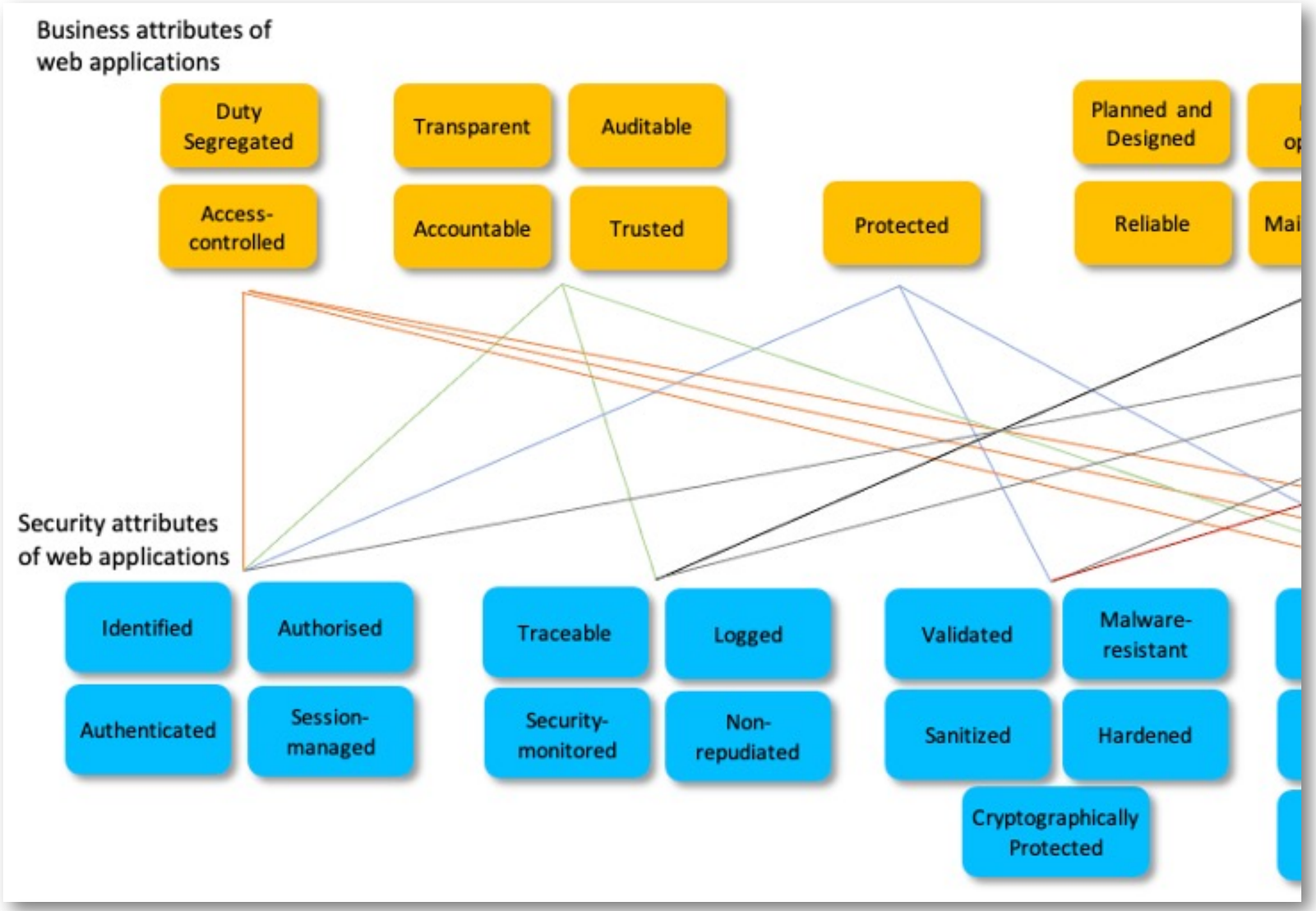
Criteria	Select the level
Based on the national "Security Standard no 1 – Technical Risk Assessment", what is the technical risk level of the web application?	**Medium**
Based on the national "Security Standard no 3 – Security Classification", what is the highest level of information the web application is processing?	**UNCLASSIFIED-PERSONAL**
The OWASP ASVS security verification level of the web application is:	**2**

Contextual Architecture

- Business attributes of web applications

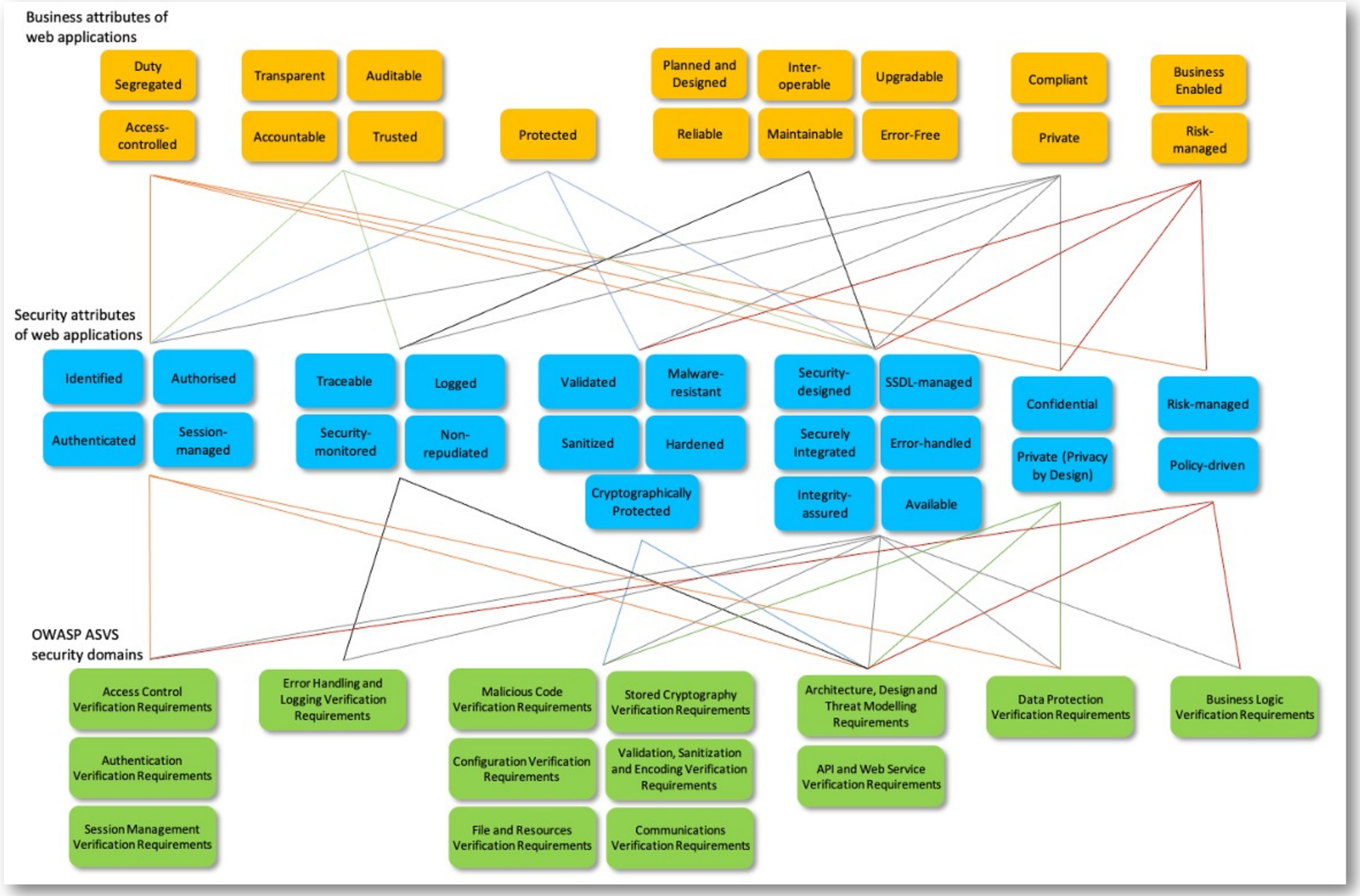
Conceptual Architecture

- Security attributes of the web application



Logical security architecture

Chain of Traceability



Physical security architecture

Requirement ID	Security Requirement	Level 1	Level 2	Level 3	STATUS
V1.14.6	Verify the application does not use unsupported, insecure, or deprecated client-side technologies such as NSAPI plugins, Flash, Shockwave, ActiveX, Silverlight, NACL, or client-side Java applets.		X	X	In Progress
V2.1.1	Verify that user set passwords are at least 12 characters in length (after multiple spaces are combined).	X	X	X	In Progress
V2.1.2	Verify that passwords 64 characters or longer are permitted but may be no longer than 128 characters.	X	X	X	Implemented
V2.1.3	Verify that password truncation is not performed. However, consecutive multiple spaces may be replaced by a single space.	X	X	X	Unsolved
V2.1.4	Verify that any printable Unicode character, including language neutral characters such as spaces and Emojis are permitted in passwords.	X	X	X	Not Applicable
V2.1.5	Verify users can change their password.	X	X	X	Implemented
V2.1.6	Verify that password change functionality requires the user's current and new password.	X	X	X	Implemented
V2.1.7	Verify that passwords submitted during account registration, login, and password change are checked against a set of breached passwords either locally (such as the top 1,000 or 10,000 most common passwords which match the system's password policy) or using an external API. If using an API a zero knowledge proof or other mechanism should be used to ensure that the plain text password is not sent or used in verifying the breach status of the password. If the password is breached, the application must require the user to set a new non-breached password.	X	X	X	In Progress

Component security architecture

- Additional external resources of:
  - Technical guides
  - Standards
  - Cheat sheets
  - Best practices

Service management security architecture

No	Criteria	Status
1	Are all assets (e.g. physical and virtual servers, web servers, database servers, application servers, load balancers) added to the IT asset repository?	Implemented
2	Are primary and secondary administrator assigned to each web application system component?	Implemented
3	Have all system components and platforms hardened configurations (e.g. based on well-known hardening guides provided by CIS Security, NIST NCP or similar)?	In Progress
4	Are all user and service accounts, which are related to the web application's front and backend system components, included in the organisation's primary Identity and Access Management (IAM) process?	Implemented
5	Are all relevant logs, which are related to the web application front and backend system components (e.g., logs of web server, API, application, database queries, TLS-session management), transported to the organisation's central log repository?	Unsolved
6	Are the web application and all its components added to the organisation's primary security monitoring process?	In Progress
7	Are the web application and all its components added to the organisation's primary patch management process?	Implemented
8	Are the web application and its components added to the organisation's primary vulnerability	Implemented

Comments and questions?

Thank You!

Follow us: **egovacademy**

ISO 9001
BUREAU VERITAS
Certification

eGA.ee