Enterprise Security Architecture—A Top-down Approach

Implementing security architecture is often a confusing process in enterprises. Traditionally, security architecture consists of some preventive, detective and corrective controls that are implemented to protect the enterprise infrastructure and applications. Some enterprises are doing a better job with security architecture by adding directive controls, including policies and procedures. Many information security professionals with a traditional mind-set view security architecture as nothing more than having security policies, controls, tools and monitoring.

The world has changed; security is not the same beast as before. Today’s risk factors and threats are not the same, nor as simple as they used to be. New emerging technologies and possibilities, e.g., the Internet of Things, change a lot about how companies operate, what their focus is and their goals. It is important for all security professionals to understand business objectives and try to support them by implementing proper controls that can be simply justified for stakeholders and linked to the business risk. Enterprise frameworks, such as Sherwood Applied Business Security Architecture (SABSA), COBIT® and The Open Group Architecture Framework (TOGAF), can help achieve this goal of aligning security needs with business needs.

SABSA, COBIT and TOGAF and Their Relationships

SABSA is a business-driven security framework for enterprises that is based on risk and opportunities associated with it. SABSA does not offer any specific control and relies on others, such as the International Organization for Standardization (ISO) or COBIT processes. It is purely a methodology to assure business alignment.

The SABSA methodology has six layers (five horizontals and one vertical). Each layer has a different purpose and view. The contextual layer is at the top and includes business requirements and goals. The second layer is the conceptual layer, which is the architecture view. Figure 1 shows the six layers of this framework.

![Figure 1—SABSA Framework Layers](source: SABSA, SABSA White Paper, 2009. www.sabsa.org/sabsa-white-paper. Reprinted with permission.)

COBIT® 5, from ISACA, is “a comprehensive framework that assists enterprises in achieving their objectives for the governance and management of enterprise IT.”! This framework includes tool sets and processes that bridge the gap between technical issues, business risk and process

Rassoul Ghaznavi-Zadeh, CISM, COBIT Foundation, SABSA, TOGAF

Has been an IT security consultant since 1999. He started as a computer network and security professional and developed his knowledge around enterprise business, security architecture and IT governance. Ghaznavi-Zadeh is an IT security mentor and trainer and is author of several books about enterprise security architecture and ethical hacking and penetration, which can be found on Google Play or in the Amazon store.
TOGAF is a framework and a set of supporting tools for developing an enterprise architecture. The TOGAF architecture development cycle is great to use for any enterprise that is starting to create an enterprise security architecture. Similar to other frameworks, TOGAF starts with the business view and layer, followed by technology and information (figure 5).

requirements. The goal of the COBIT 5 framework is to “create optimal value from IT by maintaining a balance between realising benefits and optimising risk levels and resource use.” COBIT 5 aligns IT with business while providing governance around it.

The COBIT 5 product family has a lot of documents to choose from, and sometimes it is tough to know exactly where to look for specific information. Figure 2 shows the COBIT 5 product family at a glance. COBIT Enablers are factors that, individually and collectively, influence whether something will work.

The COBIT framework is based on five principles (figure 3). Applying those principles to any architecture ensures business support, alignment and process optimization.

By using a combination of the SABSA frameworks and COBIT principles, enablers and processes, a top-down architecture can be defined for every category in figure 2. As an example, when developing computer network architecture, a top-down approach from contextual to component layers can be defined using those principles and processes (figure 4).
TOGAF is a useful framework for defining the architecture, goals and vision; completing a gap analysis; and monitoring the process.

By using SABSA, COBIT and TOGAF together, a security architecture can be defined that is aligned with business needs and addresses all the stakeholder requirements. After the architecture and the goals are defined, the TOGAF framework can be used to create the projects and steps, and monitor the implementation of the security architecture to get it to where it should be.

**Using the Frameworks to Develop an Enterprise Security Architecture**

The fair question is always, “Where should the enterprise start?”

If one looks at these frameworks, the process is quite clear. This must be a top-down approach—start by looking at the business goals, objectives and vision.

The initial steps of a simplified Agile approach to initiate an enterprise security architecture program are:

- Identify business objectives, goals and strategy
- Identify business attributes that are required to achieve those goals

---

**Figure 4—SABSA and COBIT Integration**


**Figure 5—TOGAF Framework Life Cycle**

*Source: TOGAF 9.1 Architecture Development Cycle. Reprinted with permission.*

- Identify all the risk associated with the attributes that can prevent a business from achieving its goals
- Identify the required controls to manage the risk
- Define a program to design and implement those controls:
  - Define conceptual architecture for business risk:
    - Governance, policy and domain architecture
    - Operational risk management architecture
    - Information architecture
    - Certificate management architecture
    - Access control architecture
    - Incident response architecture
    - Application security architecture
    - Web services architecture
    - Communication security architecture
  - Define physical architecture and map with conceptual architecture:
    - Platform security
    - Hardware security
    - Network security
    - Operating system security
After all risk is identified and assessed, then the enterprise can start designing architecture components, such as policies, user awareness, network, applications and servers.

- Directory security
- File security
- Database security, practices and procedures
- Define component architecture and map with physical architecture:
  - Security standards (e.g., US National Institute of Standards and Technology [NIST], ISO)
  - Security products and tools (e.g., antivirus [AV], virtual private network [VPN], firewall, wireless security, vulnerability scanner)

It is that simple. After all risk is identified and assessed, then the enterprise can start designing architecture components, such as policies, user awareness, network, applications and servers.

Figure 6 depicts the simplified Agile approach to initiate an enterprise security architecture program.

Figure 6—Enterprise Security Architecture Model
A Real-Life Example

This section describes a simple and practical example of the steps that can be taken to define a security architecture for an enterprise.

The enterprise in this example is a financial company, and their goal is to have an additional one million users within the next two years. Some of the business required attributes are:

- **Availability**—Systems need to be available to customers at all times.
- **Customer privacy**—Customers’ privacy needs to be ensured.
- **Accuracy**—Customers’ and company information must be accurate.
- **Regulatory**—Company is under regulation (Payment Card Industry [PCI] in this case) and must be aligned with regulatory requirements.

Some of the business risk includes:

- Not having a proper disaster recovery plan for applications (this is linked to the availability attribute)
- Vulnerability in applications (this is linked to the privacy and accuracy attributes)
- Lack of segregation of duties (SoD) (this is linked to the privacy attribute)

Some of the controls are:

- Build a disaster recovery environment for the applications (included in COBIT DSS04 processes)
- Implement vulnerability management program and application firewalls (included in COBIT DSS05 processes)
- Implement public key infrastructure (PKI) and encryption controls (included in COBIT DSS05 processes)
- Implement SoD for the areas needed (included in COBIT DSS05 processes)
- Implement PCI DSS controls

All of the controls are automatically justified because they are directly associated with the business attributes.

Like any other framework, the enterprise security architecture life cycle needs to be managed properly.

"Like any other framework, the enterprise security architecture life cycle needs to be managed properly."

©2017 ISACA. All rights reserved. www.isaca.org
The life cycle of the security program can be managed using the TOGAF framework. This is done by creating the architecture view and goals, completing a gap analysis, defining the projects, and implementing and monitoring the projects until completion and start over (figure 5).

Using CMMI to Monitor, Measure and Report the Architecture Development Progress

Finally, there must be enough monitoring controls and key performance indicators (KPIs) in place to measure the maturity of the architecture over time.

The first phase measures the current maturity of required controls in the environment using the Capability Maturity Model Integration (CMMI) model. The CMMI model has five maturity levels, from the initial level to the optimizing level. For the purpose of this article, a nonexistent level (level 0) is added for those controls that are not in place (figure 7).

The aim is to define the desired maturity level, compare the current level with the desired level and create a program to achieve the desired level.

This maturity can be identified for a range of controls. Depending on the architecture, it might have more or fewer controls.

Some example controls are:

- **Procedural controls**
  - Risk management framework
  - User awareness
  - Security governance
  - Security policies and standards

- **Operational controls**
  - Asset management
  - Incident management
  - Vulnerability management
  - Change management
  - Access controls
  - Event management and monitoring

- **Application controls**
  - Application security platform (web application firewall [WAF], SIEM, advanced persistent threat [APT] security)
  - Data security platform (encryption, email, database activity monitoring [DAM], data loss prevention [DLP])
  - Access management (identity management [IDM], single sign-on [SSO])

### Figure 7—CMMI Maturity Levels

<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Optimizing</td>
<td>Measured and controlled: Organization is data-driven with quantitative performance improvement objectives that are predictable and align to meet the needs of internal and external stakeholders. Stable and flexible: Organization is focused on continuous improvement and is built to pivot and respond to opportunity and change. The organization’s stability provides a platform for agility and innovation.</td>
</tr>
<tr>
<td>4 Quantitatively Managed</td>
<td>Proactive, rather than reactive: Organizationwide standards provide guidance across projects, programs and portfolios.</td>
</tr>
<tr>
<td>3 Defined</td>
<td>Managed on the project level: Projects are planned, performed, measured, and controlled.</td>
</tr>
<tr>
<td>2 Managed</td>
<td>Unpredictable and reactive: Work gets completed, but is often delayed and over budget.</td>
</tr>
<tr>
<td>1 Initial</td>
<td>Unpredictable and reactive: Work gets completed, but is often delayed and over budget.</td>
</tr>
</tbody>
</table>

Source: CMMI Institute. Reprinted with permission.
• **Endpoint controls**
  – Host security (AV, host intrusion prevention system [HIPS], patch management, configuration and vulnerability management)
  – Mobile security (bring your own device [BYOD], mobile device management [MDM], network access control [NAC])
  – Authentication (authentication, authorization, and accounting [AAA], two factor, privileged identity management [PIM])

• **Infrastructure controls**
  – Distributed denial of service (DDoS), firewall, intrusion prevention system (IPS), VPN, web, email, wireless, DLP, etc.

The outcome of this phase is a maturity rating for any of the controls for current status and desired status. After the program is developed and controls are being implemented, the second phase of maturity management begins. In this phase, the ratings are updated and the management team has visibility of the progress.

**Conclusion**

Regardless of the methodology or framework used, enterprise security architecture in any enterprise must be defined based on the available risk to that enterprise. The enterprise frameworks SABSA, COBIT and TOGAF guarantee the alignment of defined architecture with business goals and objectives.

> After the program is developed and controls are being implemented, the second phase of maturity management begins.

---

**Figure 8—Maturity Reporting and Dashboard Example**


---

©2017 ISACA. All rights reserved. www.isaca.org
Using these frameworks can result in a successful security architecture that is aligned with business needs:

- COBIT principles and enablers provide best practices and guidance on business alignment, maximum delivery and benefits.
- The COBIT Process Assessment Model (PAM) provides a complete view of requirement processes and controls for enterprise-grade security architecture.
- SABSA layers and framework create and define a top-down architecture for every requirement, control and process available in COBIT.
- The TOGAF framework is useful for defining the architecture goals, benefits and vision, and setting up and implementing projects to reach those goals.
- The CMMI model is useful for providing a level of visibility for management and the architecture board, and for reporting the maturity of the architecture over time.

The simplified agile approach to initiate an enterprise security architecture program ensures that the enterprise security architecture is part of the business requirements, specifically addresses business needs and is automatically justified.

Endnotes

3 Op cit, ISACA