1. NETWORK OVERVIEW

The most general view of Xalt's Mobility Component consists of clients, the public cloud and the private enterprise; 1, 2 and 3, respectively from Figure 1. Information flows through a network of clients, the public cloud and the private enterprise, but a fundamental tenet of security is that private enterprise data is never at rest outside of the private enterprise. This paper discusses how the network is used to deliver rich, native user interfaces across a variety of devices while maintaining a secure adherence to the fundamental tenet.

1.1 SSL/TLS SECURITY

One of the most widely used technologies for secure communications is the Secure Sockets Layer (SSL) and the follow-on internet standard known as Transport Layer Security (TLS). All communications between the parties in Figure 1 are performed through SSL/TLS. SSL provides two crucial components of the end-to-end security deployed in apps powered by Xalt: confidentiality and message integrity. Confidentiality is provided by the SSL handshake protocol to establish a shared secret used to encrypt content during the SSL session. Strong encryption codes ensure that content is unreadable by anyone other than the sender and receiver. The handshake protocol also defines a shared secret used to build authentication codes. Authentication codes are used to ensure that content is both authentic and unchanged.
1.2 XALT FIREWALL

Xalt | Mobility runs on a network of multiple and redundant machines to ensure a highly available and scalable service. To ensure that this service is protected from malicious outsiders, the network is hidden behind a firewall that only allows access by the SSL/TLS service mentioned in the previous section. Furthermore, Xalt runs on a well-known domain identified by an RSA certificate signed by a trusted authority. The certificate is used by all client devices to ensure that they are communicating with the true service and not an imposter.

1.3 PRIVATE ENTERPRISE FIREWALL

The private enterprise is protected by its own firewall. Hidden behind the enterprise firewall is a Connector Gateway (satellite) that acts as a gateway and proxy for requests. The service contacts the satellite server from a well-known range of IP addresses. The enterprise firewall prevents unauthorized contact by only routing requests that originate from the well-known addresses to the satellite server. Moreover, the service is configured with the IP address of the satellite server and will only communicate with authorized satellites. Finally, the satellite server is identified by an RSA certificate signed by Hexagon to ensure confidentiality and integrity of communications.

2. PROCESSING OVERVIEW

Client requests are routed through the network by a chain of processes (see Figure 2). This section gives an overview of the processing and the kinds of data used to service client requests.

2.1 THE CONTROLLER LAYER

Xalt | Mobility operates over a distributed set of databases (see Figure 3), but in accordance with the fundamental tenet of security, private enterprise data is never replicated or stored outside of the private enterprise.
The database that exists in the cloud [6] defines user interface structures and conceptual schema views. Secured behind the enterprise firewall is a satellite database [4] that defines connection descriptors and a schema catalog. The administrator populates these databases [6, 4] with metadata that define end user applications. Database concerns are separated and layered such that sensitive data remains at the enterprise behind the enterprise firewall [4, 8].

The administrator is able to define applications using a wide range of metaphors such as workbenches, launchers, lists, maps, graphs, images, forms, menus and toolbars. Rendering a user interface from the metaphors involves processing the multiple databases [4, 6, 8] with a patented technology2 that maps client requests over conceptual and physical data in real time. The mappings are delivered in device-independent formats from data source to target device without caching. Ultimately, smart user interface programs consume the mappings to render native applications tailored to the target device.

2.2 USER AUTHENTICATION AND SESSION MANAGEMENT

The external client is a program running on a target device like a phone, tablet or personal computer. The authentication process begins with a login prompt at the client program.

If using multi-factor authentication, the user retrieves the current one-time-password (OTP) from a generator device like Google Authenticator or a Key Fob [1].

The OTP is generated from a secret code [5] installed on the device by the security administrator. The user then enters a user ID, password and OTP (if applicable) and presses submit. The client program packages the login parameters into a login request and sends it to Xalt [2] over a secure SSL connection.

When setting up a user, the administrator can choose to have user credentials authenticated by Hexagon or by the enterprise. If the admin chooses Hexagon authentication, a cryptographic hash of the password is stored in the cloud. A cryptographic hash cannot be used to reconstitute the password, it can only be used to verify a password that has been encrypted using the same key. This is a secure technique for authenticating users without maintaining a password in storage. If the admin chooses enterprise authentication, then the authentication request is passed to the private enterprise [3, 7] over a secure SSL connection.

The customer can also register their existing OAUTH Authorization Server with Hexagon Extender. They can then choose to allow users to login via a standard OAUTH authorization code flow with their Authorization Server instead of passing user credentials to Hexagon.

A successful authentication and login returns a session handle and a secret hash key2 used in subsequent requests to construct a signature. The hash key prevents a third party from acquiring a user session through a brute-force attack on a session handle. A failed login results in an error message.

It is important to realize that the authentication process simply creates a user session. In essence, a successful login allows a user to make subsequent data requests, but does not guarantee that data access will be allowed. Actual data access occurs inside the enterprise and is subject to the security policy of the data sources. For example, a user can authenticate successfully by way of active directory [9], but later be denied access to a stored procedure because he is not authorized by the DBMS [8].

2.3 ENTERPRISE QUERIES AND TRANSACTIONS

Xalt’s Mobility allows an administrator to create a variety of data views to be presented at the client device. In addition to viewing data, there is support for changing data by submitting transactions to the enterprise. A significant feature of Xalt’s Mobility Component is its ability to generalize queries and transactions independent of data source type. Client programs have no clue as to the data source type or the number of data sources being accessed to present a view. This ability to generalize schemas is useful for creating narrow views of data that adhere to the security principle of least-priveleges4.

Also, generalizing over multiple data sources is useful for creating composite applications that adhere to the principle of least privilege. Composite applications are an advance technique in application integration5.

All user interaction begins with a session handle, action request and request signature submitted by the client to Xalt [2] over a secure SSL connection. In the course of processing an action, the cloud retrieves associated user interface metadata [6], submits necessary data requests to the satellite server over a secure SSL connection [3] and correlates response data for client devices. The user interface metadata, requests sent to the connector gateway and correlated response data are formulated in terms of a conceptual schema defined by Xalt. Actual queries and transactions are performed inside the enterprise and are subject to the security policy of the data sources.

The connector gateway receives data requests in terms of conceptual meta data [3] void of data source details. The gateway must correlate a variety of identifiers in order to execute a physical query or transaction. The gateway uses local metadata defined by the administrator to formulate actual requests with most of the processing occurring in the connector responsible for each data source. Physical queries and transactions are issued by the connectors to the underlying systems [8].

Responses are constructed by reversing the request process so that data returned to the client programs is once again in terms of the conceptual schema defined by Xalt.
3. CLOSED SECURITY ARCHITECTURE

Xalt | Mobility is based on a closed security policy, which means all actions on resources are implicitly denied until authorization is explicitly granted. This is made obvious at the administration console where new users have an empty workbench until explicit permissions are granted by the administrator. Each user or group must be granted access to data objects, and must be granted visibility to lists, maps, graphs, forms, etc.

There are two fundamental approaches to enforcing security policy: open or closed.

Most importantly, the client programs are based on a closed security policy. The client programs do not use consumer-based browser technology for application rendering and navigation. Consumer-based browsers are highly vulnerable to security attacks and spoofing because they were created on the idea of openly navigating the web. Instead, the client programs are built on native runtimes or sandboxed virtual machines programmed to only process requests. The protocol is based solely on JSON RPC that must be served from authorized servers using secure SSL connections.

4. MOBILE IDENTITY MAPPING: GENERAL VS SPECIFIC

Mobile apps introduce external users into the enterprise architecture, and, along with external users, are special considerations and maintenance concerns. One problem in particular is the large increase in users and the overhead involved in creating, assigning and authorising aspects of external access. This problem is exacerbated by the fact that external apps are usually quite different from internal apps for the same users. Xalt | Mobility includes a special consideration for maintaining large groups of external users without compromising enterprise security. It allows large groups of users to be generalized into mobile app groups while maintaining user identity thru application servers. This is a general to specific mapping of user identity. Other identity mappings are available. Table 1 shows how mappings can be combined in various ways to suit the sensitivity level of enterprise applications.

<table>
<thead>
<tr>
<th>Extender</th>
<th>Enterprise</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Specific</td>
<td>Used by enterprises with large numbers of users. Users are organized into groups. Workbenches and user interfaces are defined per group. This mapping minimizes administration and reuses existing enterprise identities.</td>
</tr>
<tr>
<td>General</td>
<td>General</td>
<td>Used by enterprises to secure low risk supply chain applications. This mapping minimizes both Xalt and back-office enterprise administration. An example of this technique is a supplier portal that maps all users as a single supplier organization to a single enterprise identity representing the supplier.</td>
</tr>
<tr>
<td>Specific</td>
<td>General</td>
<td>Used by enterprises that want to manage access to low risk applications discretely without managing an internal account for each external user. This technique has the advantage that internal accounts are not exposed to external users.</td>
</tr>
<tr>
<td>Specific</td>
<td>Specific</td>
<td>A pass through mapping. The user’s identity is the same throughout the system. The enterprise performs authentication and workbenches, and user interfaces are managed discretely per user account.</td>
</tr>
</tbody>
</table>
5. EXAMPLE SATELLITE DEPLOYMENT WITHIN A DMZ

The following diagram is an example gateway deployment within a DMZ. This is only an example since an actual gateway deployment depends on the topology and architecture of the network technology implemented by the enterprise.

5.1 CONNECTOR GATEWAY IP RANGE

Although not strictly required, the connector gateways are usually reserved ports 8095 thru 8105. A single connector gateway utilizes a single port and handles its request/response protocol over HTTP (or HTTPS if not terminating SSL at the firewall).

5.2 ENTERPRISE CONNECTORS JDBC, LDAP, TCP/IP, ETC.

The enterprise admin configures which back office data sources are utilized by Xalt | Mobility. Depending on the data sources used, the enterprise network administrator will need to authorize the proper protocols, ports, and endpoint routings. The example in Figure 4 shows JDBC, LDAP, and an arbitrary TCP/IP connection as an example of three data source connectors.

6. SUMMARY

Xalt | Mobility’s approach to security is inherently conservative, highlighted by: the fundamental tenet that private enterprise data is never at rest outside of the private enterprise; the use of secure cryptographic techniques; the secure separation of concern in databases; the support of the least privilege access principle; and the closed security policy that denies access unless permission is explicitly granted. In spite of the conservative approach to security, Xalt’s Mobility Component offers a flexible and uncompromising SaaS solution for creating rich, native user interfaces over a variety of enterprise data sources.
7. REVISION HISTORY

2013.05.22 Added “Satellite Deployment Example”
2013.05.22 Added “General Identity vs Specific Identity”

8. REFERENCES


[3] Stallings, William; Brown, Lawrie. Computer Security Principles and Practice, Second Edition. Page 52. A cryptographic hash function should meet six requirements: (a) the hash is easy to compute for a message of any size; (b) the hash is easy to compute for any data type; (c) the hash function produces a fixed length output; (d) it is computationally infeasible to generate a message for a given hash; (e) given a message, it is computationally infeasible to find another message with the same hash; (f) it is computationally infeasible to find two messages with the same hash.


About Hexagon
Hexagon is a leading global provider of information technology solutions that drive productivity and quality across geospatial and industrial landscapes.

Hexagon’s solutions integrate sensors, software, domain knowledge and customer workflows into intelligent information ecosystems that deliver actionable information. They are used in a broad range of vital industries.

Hexagon (Nasdaq Stockholm: HEXA B) has more than 18,000 employees in 50 countries and net sales of approximately 3.5bn EUR.

Learn more at hexagon.com and follow us @HexagonAB.

©2019 Hexagon AB and/or its subsidiaries and affiliates. All rights reserved.