The UbiVault Fabric security model

Marek Podgorny, PhD., UbiVault LLC, January 14, 2019

InterVault™ is a software *fabric* for end-to-end (E2E) security in peer-to-peer (P2P[1]) communications. InterVault™ Fabric (hereafter referred to as Fabric) is a software-defined secure routing system running on of the Internet. While it focuses on P2P communication, client-server communication is supported as a default case addressed below.

Security is a broad term, in our case providing a secure communications path between computing devices, processes and resources such as data storage. The goal is to mitigate risks from hostile activity, injected or embedded at any node, from outside actors or from internal human intervention. The intent is to provide an efficient, least complicated solution for securing data in motion and at rest, regardless of code execution in any distributed node in a network. This security goes beyond just data encryption.

Examples of P2P applications include all types of messaging[2] between individuals, including bots. Such applications are typically symmetrical, i.e., same type of information flows back-and-forth between the peers. E2E encryption for such apps means that the messages/files/protocols encrypted by the sender can only be decrypted by the addressee.

Perhaps the easiest way to explain E2E is to consider an app NOT providing E2E encryption: Alice sends a message to a web server using HTTPS. Servers decrypt the payload and then forward it to Bob connected to the same server. The message is sent over a secure layer protocol (HTTPS/TLS) using a different pair of public/private keys. But the server is in possession of decrypted message, even for a short time. Such architecture defines a potential security hole, namely a man-in-the-middle agent between peers[3]. The Fabric mitigates this issue.

Original concept behind the Fabric first appeared in late nineties by CollabWorx, a DoD contractor under direction of the author. It was used in government agencies for secure collaboration. Many years later, key elements were redesigned and appeared in P2P applications such as Telegram, Whisper/Signal, Snapchat, and WhatsApp. Security encryption

---

[1] P2P defines a network consisting of two or more nodes that allow synchronous and asynchronous connections. Nodes are addressable using internet protocol (IP) addressing. A node is defined as any physical system that supports software enabled processes.

[2] To the Fabric, messages refer to the delivery of a data payload in the form of data packets that conform to TCP and UDP protocols. This goes beyond simple text and includes arbitrary data (blobs) that an application can process.

[3] The term E2E often describes a client-server link across firewalls/other network infrastructure. We describe it as a complete path between multiple nodes crossing different environments.

schemes used by these apps are all based on Off-the-Record Messaging (OTR), a cryptographic protocol for secure instant messaging, a subset of the Fabric.

The Fabric is also used to secure standard client-server links. While this is not an E2E/P2P application in a traditional sense many InterVault features significantly enhance the HTTPS protocol, providing an impenetrable security envelope for any cloud-based Services, FINTECH, supply chain, insurance, and healthcare, to name a few. The following sections describe how the InterVault authentication security protocol works

InterVault Proxy Authentication

Fabric transparency

Sending text/media/file messages is native to the Fabric and integrates with any type of messaging application. In this scenario the Fabric first authenticates the application user. Authentication service is done by the InterVault Authentication Server (IVAS), a.k.a. SafePerimeter.

The Fabric also provides *transparent* secure transport for any arbitrary cloud application or process. Transparency implies that there are *no* integration points between the Fabric and the application being protected. Specifically, the Fabric authentication and the authentication used by the Service are by design, *distinct and separate*. (Note: we refer to any application protected by the Fabric as “Service” with upper-case S.)

Human vs. software agent authentication

To achieve this transparency the Fabric implements the notion of an InterVault Client/Proxy (IVC/P). Depending of the application network topology IVC/Ps are implemented as a client-side proxy (a proxy accepting and forwarding network payload), a server-side proxy (a proxy receiving network payload from the Fabric and forwarding it to the target application), or a symmetric bi-directional proxy (i.e., a messaging app). Multi-platform client-and server-side Intervault proxies handle all currently used asynchronous request/response protocols, including secure HTTP and its extensions (HTTP/2, SPDY, QUIC, etc.)

IVC/P instances must first be authenticated and authorized for a specific cloud service before activation[4]. The method to do so is described below. Software agent authentication service is also provided by InterVault SafePerimeter.

An authenticated instance of IVC/P becomes a gateway to the Fabric network - the *only way* for users to access a networked Service. This effectively shuts down all known network attack vectors, *including DDNS and human factor attacks*.

Traceability and Audit

Investigation of network security breaches are difficult due to lack of sufficient network traffic traceability. For the most part, cybersecurity solutions focus on perimeter defense, application authentication and rules, user identification procedures and permission tables. These are costly to maintain and monitor. There are several mechanisms in place to support transaction auditing of all traffic traversing the Fabric. It is important to remember that transaction content remains opaque and by design, cannot technically be retrieved or diverted inside the Fabric.

Controlled distribution of application code

IVC/P binary code is designed to be ***unique at every node*** (device, server, cloud process.) It is obtained from a secure UbiVault[5] repository. In the case of mobile devices, each installation download requires user email address and mobile phone number, both subject to verification. Downloaded application code is assigned a global, unique UbiVault identification token (UUID), which in turn is embedded in the IVC/P code together with user data[6], all digitally signed by UbiVault.

[4] To appreciate the difference between InterVault and "legacy" security approach consider the common browser application. InterVault approach would require that the *browser app itself* is authorized to access specific cloud Services.

[5] UbiVault LLC markets and supports solutions based on InterVault Fabric.

[6] While the UUID token is primarily to authenticate a node, it can include biometric data that authenticates a useer of a node such as a mobile phone or computer.

Identification token and user data are registered in a database that is accessible to InterVault's SafePerimeter. Verification procedures embedded in IVC/P code prevent execution if the code has been modified or replaced. Each downloaded instance of the IPV/C code is hence unique[7].

Message signing certificates: generation and use

Upon 1st execution IVC/P automatically attempts registration with SafePerimeter. Based on a combination of the publisher certificate[8] and an UUID hash IVC/P logs into SafePerimeter, which, in turn, return a unique identifier serving as a unique user name (identifier for application execution) and a one-time key to encrypt the UUID, a hash of which serves as a password. In addition, SafePerimeter generates and returns a pair of private/public keys that IVC/P uses to sign messages.

Non-repudiation support

Signing selected messages (not necessarily those transporting payload) provides a proof of each transaction origin within the Fabric. But combined with UUID there is a non-repudiable trace of transaction workflow. Legally significant non-repudiation is also possible by time-synchronized overlay of the transaction logs from the Fabric and any Service.

Complexity

While the security process in the Fabric seems cumbersome, it is algorithmic and completely automated except for the request to user to provide user details such as email address and mobile telephone number, performed only once.

User identity

Having established secure Fabric gateway, we now turn to user identity. User authentication is performed with every Service access. By "user" we understand (a) a person owning mobile device, (b) a person authorized to access the Service. User identification is supported by two respective elements:

1. User must be biometrically authenticated with the IVC/P. This step provides a link between the user, the device, and the proxy that enables secure access.
2. User must then authenticate with a Service. Upon accessing IVC/P the user is connected to the Service landing or if necessary, authentication page.

[7] It should be noted that the mobile app version of IPV/C is currently not designed for download from Google or Apple stores. IPV/C is not suitable for general public use, doesn't support ads, and it is hence of little interest for the store owner. UbiVault may, in the future, release InterVault Fabric aimed at general public. Google and Apple stores permit implementation of a controlled distribution process similar to the UbiVault process.

[8] Digital certificates are used for one time registration of the unique binary payload representing IVC/P.

Keep in mind, Fabric and Service authentication processes are distinct and separate. Together, they form an access-control cascade. The outcome is a double-blind security envelope. The Fabric does now know a user's identity, and the Service doesn't know the authentication basis of IVC/P providing access through the Fabric to the Service user. Any attacker would have to compromise both systems to gain access.[9]

User identification loop and KYC principle

In principle, the Fabric and Service authentication processes are independent. But, in some cases it may be useful to combine the data albeit at the cost of breaking the double-blind argument. The SafePerimeter API provides a way for the Service authentication server(s) to access some of the data used (or created) by the Fabric. The Service might be able to use such data to construct a legally valid proof of transactions submitted by a particular Service user. Further, combining Service and Fabric data may aid Service operator in construction of the dynamic elements of the operator KYC database.

Summary

We introduced UbiVault Fabric as a secure authentication and data transport solution, designed for ease of integration and scalability, in onboarding nodes in a centralized or distributed network architecture. The combination of user and device identifiers, encrypted UUID hashes for every node (device, server, cloud process) and SafePerimeter functions, allows the Fabric's security (InterVault) for use in many environments, targeted at P2P networks. It is well suited for mobile applications that require E2E security at all network points.

Author

Marek Podgorny is a founder and CSO at UbiVault. He holds a PhD in Theoretical Physics and was an early pioneer in the expansion of the Internet, managing the Syracuse University Notheast Parallel Architectures Center, a testbed for academic research and government projects.

marek@ubivault.com

[9] The Fabric protects any Service against DDNS attacks using a whitelist of Internet resources authorized for access.