Deployment Overview

Last Updated: Thursday, July 6, 2017
www.terainsights.com
info@terainsights.com
thomas@terainsights.com

tera insights

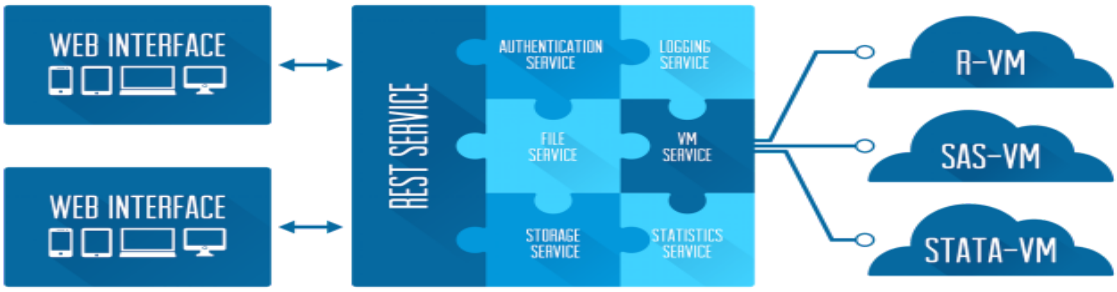
Overview

Rolling out tiCrypt in your organization a three step process: deployment, on-boarding, and tutorials. This document provides an overview of a tiCrypt deployment, system requirements, and a breakdown of what is covered in the deployment fee. For more information please contact thomas@terainsights.com.

Backend Deployment

General info

tiCrypt has a modern, microservice backend with a REST interface. Each component can be hosted, replicated, and managed independently. This design ensures fault tolerant and scalable deployment.



The tiCrypt Architecture is based on the following:

- REST Service allows the communication between application front-ends with the back-end services.
- Authentication Service is responsible for logins, sessions, groups and users.
- File Service manages files, directories and file keys.
- Storage Service is responsible for the the on-disk storage and replication of encrypted file data.
- Logging Service coordinates and tracks system wide logging and maintains the unforgeable log.

- VM Service orchestrates the virtual machine deployment, and manages virtual machine images and encrypted drives.
- Statistics Service provides a global view of system health and usage.

System requirements

The tiCrypt solution offers a simple deployment that is reliant on few dependencies and minimal configuration. Specifically, the back-end tiCrypt server is packaged as digitally signed RPMs and can be installed on RedHat Enterprise or Centos in a couple of minutes. The tiCrypt backend needs few dependencies:

- A distributed file system to connect the main server with the virtual-machines servers. We recommend LustreFS but other scalable, high-performance distributed file systems work as well.
- Java JRE version 8 or higher
- Libvirt and QEMU installed on all virtual-machine servers
- SSH installed on all virtual-machines servers (libvirt management uses SSH)

Recommended hardware

tiCrypt requires a management server that runs most of the tiCrypt backend code and unlimited number of virtual machine servers, which host the virtual machines. Below, we provide details on the specific requirements.

Management server requirements

- 2 Servers
    - 1 Public Facing Server: runs the REST interface and port forwarding
    - 1 Back-end Services Server : runs all other components
- OS: RedHat Linux Enterprise 7 or equivalent
- 16 Core, 64GB RAM for each of the 2 servers
- 1Gbit external connectivity (from public facing server)
- 10Gbit or Infiniband connectivity for the rest of the servers

VM server requirements

- 1+ servers to run virtual machines
- Connected to the fast internal network
- OS: RedHat Linux Enterprise 6+ or equivalent (preferably RedHat 7)
- Minimum req: 8 Core, 16GB RAM

- Prefered req: 20-80 Core, 64GB - 1TB RAM
    - Much higher density
    - Allows much larger VMs

NOTE: tiCrypt can be deployed on new or existing hardware. In general, there are no specific vendors we require or recommend. We strongly encourage you to use vendors you already have a strong relationship with. As part of the initial deployment effort (covered by the deployment fee) we will help you integrate your hardware with tiCrypt.

Network connectivity and isolation

tiCrypt is designed to run isolated from other resources you might have in your organization and to have strictly controlled entry points. The specific networking requirements are:

- External connectivity: 1Gbit/s or better (10Gbit/s prefered)
    - This is the connection to the outside world (i.e. your other organizational infrastructure)
- Internal connectivity: 10Gbit/s or Infiniband (Infiniband prefered)
    - The internal network is used to provide file storage, virtual machine disk images and virtual machine communication. A large amount of internal traffic is expected
- TLS connectivity for the REST interface and static assets
    - Runs on the public facing server
    - Must use strong crypto
- Range of ports (10-100) for SSH Proxying opened on the public facing server
- Firewall protecting the installation in front of the public server
    - The firewall should block all other traffic (outside TLS port 443 and Proxy port range)
    - Backdoor SSH access to the management servers and VM servers (preferably through secondary network)

Bootstrapping the tiCrypt backend

The system uses a site key that is itself signed by a Tera Insight's private key verifiable with a hard-wired (in both the front-end and back-end) public key. The site key is used to bootstrap the security of the entire software deployment. The public part of the site key is readily available to all parts of the system. Its integrity can be verified by checking the digital signature using the hardwired Tera Insights public key.

The following steps are needed (detailed user manuals are available for the specifics)

- The Site Key has to be generated and handed over to Tera Insights for counter-signing
    - The site key is used to bootstrap the entire system
    - It allows adding and managing the escrow users and super-admins

    - It is a very important security component and needs to be carefully used
        - Site Key administration is offline (does not require internet access for added protection)
        - It uses signed certificates exclusively
    - The Site Key administrator has to be carefully selected (head of Research Computing is a good candidate)
- Escrow groups need to be added
    - From Site Key interface
    - Escrow groups allow redundancy in key recovery efforts
    - 3+ groups recommended from different parts of the organization
- Escrow users need to be added
    - Escrow users allow key recovery (in case of loss or malice)
    - Special interface
    - No need to be regular users as well (treated independently)
- Super-admin accounts have to be created
    - At least one but preferably two
    - Super-admins coordinate the rest of the bootstrapping process
    - Activate accounts and change roles

Backend deployment timeline

Once the hardware, network connectivity and distributed file system are in place, we have an estimated *1 week or less* deployment timeline. The main friction points (addressed by careful planning) are:

- Misconfigured front or external firewalls
    - Clients cannot access the REST interface
    - Clients cannot connect to the proxying ports (when needed)
- Internal network problems
- Distributed file system problems
- Organizational issues
    - Not clear who will be the Site Key administrator and who are the Escrow Key users
    - Not clear who will manage the system
    - Not clear who approves accounts

Tera Insights engineers have extensive expertise to help debug any such problems and to advise on how to configure the required subsystems. The deployment fee covers all such consulting.

Backup and recovery

Since tiCrypt uses end-to-end encryption and the keys are never available on the backend, the encrypted data already meets the security requirements and can be handled as secured data. As such, secure backup is not needed thus your organization can use any backup solution you have in place. This includes any cloud backup such as AWS S3 and Google Cloud.

The specific things that need to be backed up are:

1. The database content. All data in the database that is sensitive (such as keys) is encrypted.
2. Files with data fragments/chunks. These files are used to store “file content” in the tiCrypt vault.
3. Virtual machine drives (files in the underlying filesystem). The drives are encrypted (only users can recover the keys), thus they are safe to backup with traditional methods.
4. The audit log. Log files are clear text and they do not contain any sensitive information.

To ensure speed of recovery, we recommend a second file system that can sync the files in the primary file system. Syncing should be performed overnight and allows fast recovery. This is not a replacement for the main backup mechanism that should use long-term storage of files.

Frontend Deployment

Front-end deployment is a two step process. First the user must download the tiLauncher, this is a .exe or .dmg that will serve as the executable to launch and securely connect with the platform. After running the tiLauncher, a browser application will open. The user must drag or add the proper deployment file into the application. The deployment file contains configuration information for the specific deployment. The deployment file pulls all organizational branding, establishes where to pull the auto updates, address of the physical server tiCrypt is running on, and session lockout times. Each time a user logs in, the tiLauncher checks for the latest version, ensuring that the users are always working with the latest frontend. After the deployment file has been added the user just needs to his the launch button to access tiCrypt.

On-boarding

On-boarding includes getting admins using the system, setting up teams/groups for users, getting VMs setup, explaining VM program management to users, and setting up the enhancement and bug reporting system. Additionally, during the on-boarding process, we work with the university in producing tutorial videos, university specific documentation, and manuals.

Estimated time: 4 weeks or less

Training

The training process includes a 2 day onsite visit from our CTO and CMO. During the two day visit, both will hold workshops for admins and users on how to use the system and all of tiCrypt's available features. (VMs, tiAudit Log Analyzer, Admin Tools, User Features, etc).

Estimated time: 2 days