REGIONE TOSCANA

REGIONE TOSCANA - SOGGETTO AGGREGATORE

Direzione Organizzazione e Sistemi Informativi

Settore Ufficio per la transizione al digitale. Infrastrutture e Tecnologie per lo Sviluppo della Società dell'Informazione

Servizi di sicurezza infrastrutturale e applicativa per la PA Toscana, OSCAT continuous integration e controllo qualità del codice sorgente

C.I.G.: 7949486B5D

Relazione Tecnico Illustrativa

Indice

1. Scopo del documento

Il documento illustra le caratteristiche dell'attuale soluzione OSCAT ed il workflow che regola le attività di Vulnerability Assessment.
Per soluzione OSCAT si intendono:

- il complesso del Catalogo delle soluzioni Open Source,
- la Piattaforma a supporto dei:
  - rilasci del software commissionato da Regione Toscana a fornitori terzi
  - processi di Continuous Integration per il controllo della qualità del codice sorgente
  - processi di dispiegamento automatico in staging/certificazione su webApplication, ApplicationServer, piattaforme dockers

Componente centrale del progetto è la Piattaforma per lo Sviluppo e Rilascio di Componenti Software, il cui scopo è quello di supportare in tutte le sue fasi il processo di standardizzazione, attraverso l'adozione di strumenti e modi di operare tipici del mondo Open-Source.

Completano la Piattaforma procedure batch (script bash, java, job datastage) che hanno i seguenti obiettivi:

- verificare che gli applicativi in esecuzione abbiano corrispondente codice sorgente depositato su oscat (e controllato d'ufficio dai processi di continuous integration)

- deposito dei risultati emersi dai processi di Continuous Integration sulla stessa dashboard vaatixfoda che espone i risultati delle scansioni VAA a concentrare le informazioni a beneficio dei capiprogetto responsabili dei vari applicativi

- popolare un database contenente la fonte dati utilizzata dai processi di scansione VAA

2. Acronimi

Acronimi in uso nel presente documento.

API	Application Program Interface
CERT	Computer Emergency Response Team
CI	Continuous Integration
CTI	Cyber Threat Intelligence
GIT	Software di controllo versione distribuito
HW	HardWare
IDM	Identity Manager di Regione Toscana
IoCs	Indicator of Compromise
iOS	iPhone Operating System
LDAP	Lightweight Directory Access Protocol
MISP	Malware Information Sharing Platform https://github.com/MISP/
OSCAT	Open Source CATalog
PA	Pubbliche Amministrazioni
QMSS	Quality Management Software System

SCM	Source Control Management
STIX	Structured Threat Information eXpression; linguaggio strutturato di CTI
SSO	Single Sign On
SVN	Subversion - Software di controllo versione
SW	SoftWare
TAXII	Trusted Automated Exchange of Intelligence Information
VAA	Vulnerability Assessment Applicativi
VAAmobile	Vulnerability Assessment Applicativi per apparati mobile android e iOS
VAI	Vulnerability Assessment Infrastrutturali
VAATIXFODA	Vulnerability Assessment Applicativi Fonte Dati

3. Contesto di riferimento

La piattaforma e il catalogo OSCAT ha concretizzato un approccio “industriale” e integrato all’Open Source che permette di:

- rendere disponibile a capiprogetto e/o coordinatori di uno o più gruppi, enti, ecc. uno strumento attraverso il quale è possibile coordinare e delegare lo sviluppo di parti di un sistema software complesso omogeneizzando, attraverso l’applicazione di regole condivise, e integrando le parti in un unico software;
- garantire il riuso tecnico di progetti;
- garantire il continuo aggiornamento nel tempo della piattaforma e delle soluzioni software di integrazione nelle infrastrutture regionali;
- disporre di un servizio di supporto su tutti gli elementi della Piattaforma e del Catalogo, comprensivo dei servizi di installazione e configurazione degli oggetti a catalogo;
- supporto alla Community e Help-Desk;
- prevedere l’inserimento a catalogo di nuovi componenti a seguito di richieste autorizzate.
- Verificare la qualità del codice depositato nei progetti ospitati nei due repository OSCAT disponibili, SVN[1] e GIT

4. Statistiche OSCAT e CI relative all’anno 2018

- totale progetti OSCAT : 245 progetti pubblici e 780 progetti privati (100 progetti creati nel solo 2018)
- totale progetti Jenkins : 100 (80 progetti creati nel solo 2018)
- totale progetti creati in Sonar/Psonar : 280 (100 progetti creati nel solo 2018)
- 1000 ticket per progetti piattaforma OSCAT e CI
- 300 chiamate al numero telefonico di supporto per progetti piattaforma OSCAT e CI

1 SVN è mantenuto per motivi di compatibilità con i vecchi progetti, ma sono consentite aperture di nuovi progetti che fanno uso esclusivamente di GIT. Al termine della migrazione da FusionForge a GitLab la componente SVN sarà completamente dismessa, anche dal Catalogo delle componenti OpenSource.

5. Piattaforma OSCAT - Funzionalità

La piattaforma supporta il processo di rilascio e verifica di qualità delle componenti software di varia natura quali:

- Proxy Applicativi CART
- Artefatti Docker Image
- Applicazioni Web
- Web Services
- Plug-in di prodotti a catalogo
- Librerie di uso comune
- API

Tali componenti sono realizzati e, ove il progetto lo preveda e ve ne sia richiesta da altre PA, messi a disposizione della comunità open source della Regione Toscana.
Tutti i componenti progettuali condividono in linea di massima lo stesso processo di sviluppo e rilascio del software, previo superamento del controllo qualità, all'interno della piattaforma OSCAT.

Il processo e le modalità d'uso della piattaforma sono descritti dettagliatamente nei seguenti documenti e allegati, pubblicati sulla home page (sezione web ad accesso pubblico) della piattaforma (https://oscat.rete.toscana.it) :

- Manuale d'Uso della Piattaforma per lo Sviluppo e Rilascio di Componenti Software
- Manuale utente QMSS - Continuous Integration

6. Architettura della soluzione

6.1. Componenti Software

L'attuale implementazione è costituita da numero 8 (otto) VM con 4 CPU e 12 (dodici) gigaram ognuna, per una occupazione spazio disco totale (sperimentazioni, sviluppo, certificazione, produzione) di 3 (tre) terabyte.

L'attuale soluzione si basa sui componenti software illustrati nelle seguenti tabelle riassuntive.

Componente	**FusionForge**	Versione	**ultima stabile**
Categoria	Collaborative Development Environment	Maturità	Stabile/Produzione
Linguaggio	PHP	Licenza	GNU GPL
URL	http://www.fusionforge.org/	Produttore	

Tabella 1 - FusionForge

Componente	**Nexus**	Versione	**ultima stabile**
Categoria	Maven Repository Manager	Maturità	Stabile/Produzione
Linguaggio	Java	Licenza	GNU Affero General Public License Version 3
URL	https://www.sonatype.com/nexus-repo-sitory-sonatype	Produttore	Sonatype

Tabella 2 - Nexus

Componente	**Apache Subversion**	Versione	**ultima stabile**
Categoria	Version Control SystemArchitettura della soluzione	Maturità	Stabile/Produzione
Linguaggio	JavaDispiegamento	Licenza	Apache License, Version 2.0
URL	https://subversion.apache.org/	Produttore	Apache Foundation

Tabella 3 – Subversion

Componente	**Git**	Versione	**ultima stabile**
Categoria	Version Control System	Maturità	Stabile/Produzione
Linguaggio	Java	Licenza	
URL	https://git-scm.com/	Produttore	

Tabella 4 - Git

Componente	**Jenkins**	Versione	**2.60.3**
Categoria	Continuous Integration	Maturità	Stabile/Produzione
Linguaggio	Java	Licenza	Creative Commons Attribution-ShareAlike 4.0 license
URL	https://jenkins-ci.org/	Produttore	-

Tabella 5 - Jenkins

Componente	**SonarQube**	Versione	**4.5.4**
Categoria	Continuous Code Quality	Maturità	Stabile/Produzione
Linguaggio	Java	Licenza	GNU Lesser GPL License, Version 3
URL	https://www.sonarqube.org/	Produttore	SonarSource S.A.

Tabella 6 – SonarQube

Componente	**OTRS**	Versione	**4.0.1**
Categoria	Ticket Request System	Maturità	Stabile/Produzione
Linguaggio	Perl	Licenza	Affero General Public License
URL	https://www.otrs.com/	Produttore	OTRS Group

Tabella 7 – OTRS

6.2. Component Diagram

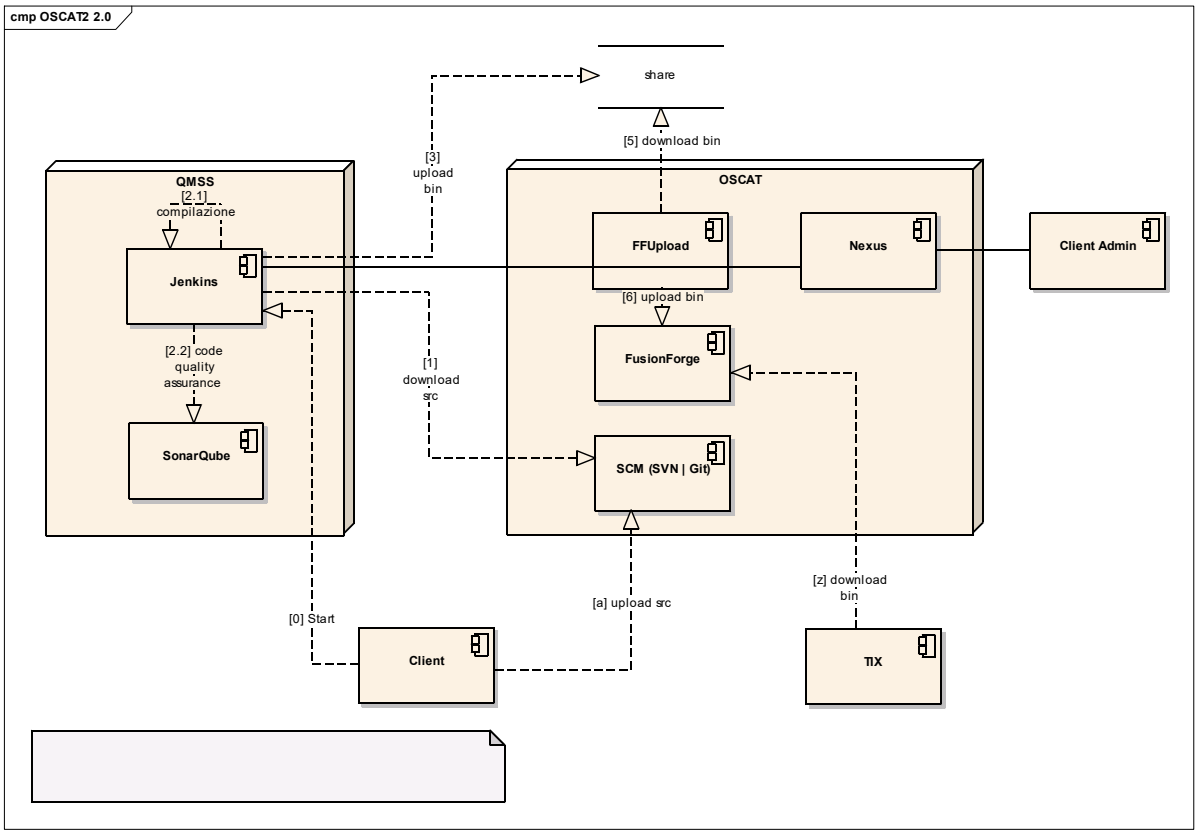


Figura 1 - Component Diagram

6.3. Dispiegamento

Le componenti sopra descritte sono dispiegate all'interno di più macchina virtuali dedicate sulle quali i componenti sono così collocati.

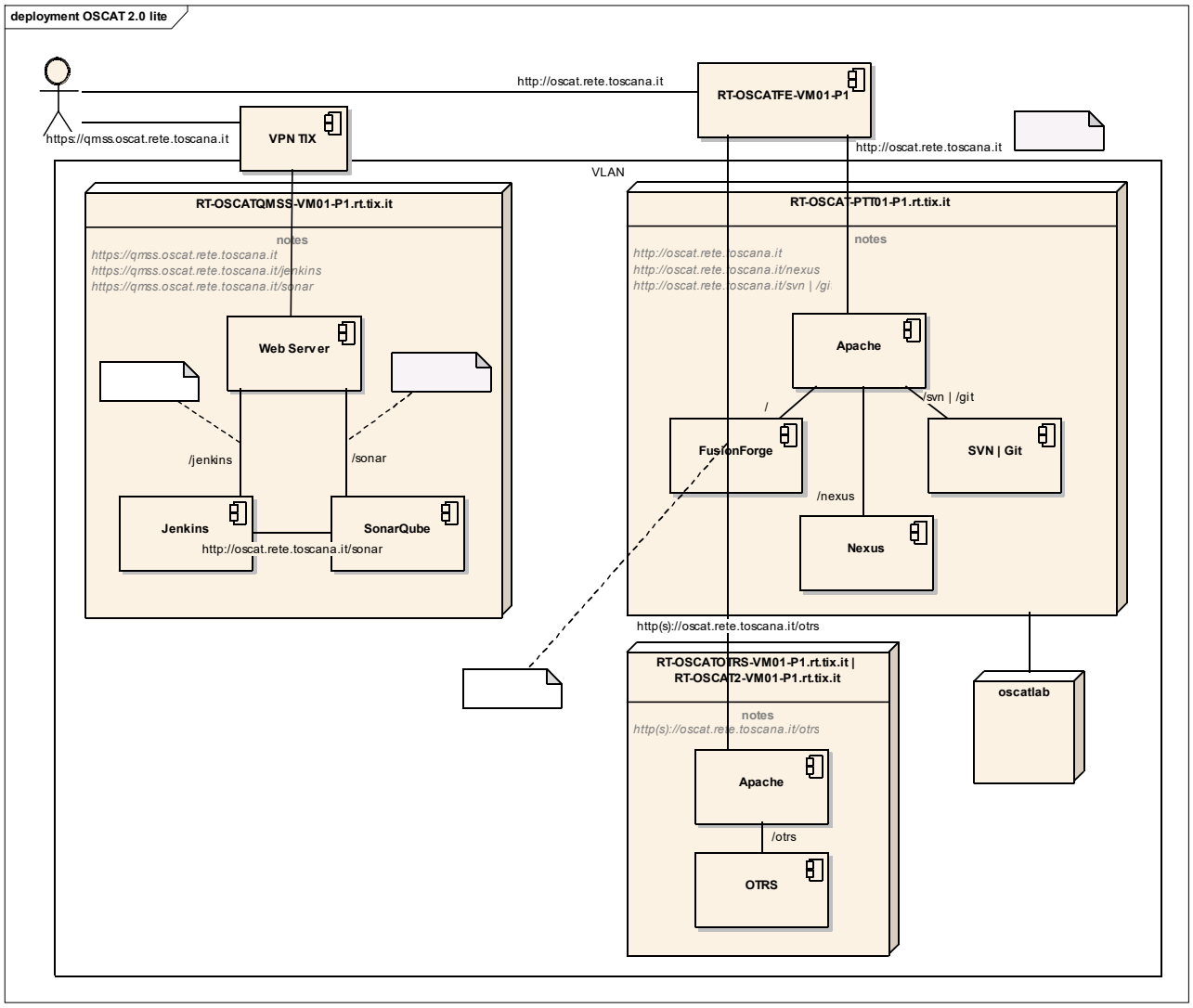


Figura 2 - Deployment Diagram

7. Processo

Il processo di rilascio e verifica è a supporto della seguente struttura organizzativa:

7.1. Gestore di Progetto

Tipicamente un coordinatore di progetto o referente tecnico di Regione Toscana, si occupa di gestire la realizzazione, personalizzazione e rilascio di un componente interagendo con gli altri utenti del sistema. Il Gestore di Progetto ha il compito di autorizzare l'inizio di un progetto OSCAT e quindi la creazione di un nuovo progetto all'interno di OSCAT. Il Gestore di Progetto si identifica sulla piattaforma OSCAT con l'utente del progetto avente il ruolo Amministratore. Per componenti progettuali di tipologia "plugin di prodotti a catalogo", il Gestore di Progetto è rappresentato dal referente dell'Ente interessato ad aderire al contratto Open Source per l'accesso e l'utilizzo delle risorse messe a disposizione dalla piattaforma OSCAT.

7.2. Sviluppatore di Progetto

Tipicamente un fornitore esterno, si occupa della effettiva realizzazione, personalizzazione e rilascio del componente attraverso l'interazione con gli altri utenti del sistema. Il Capo Progetto, lato fornitore, nell'ambito di OSCAT è comunque assimilato allo Sviluppatore di Progetto; in sostanza non è prevista una distinzione di ruolo tra gli attori coinvolti nella realizzazione del progetto OSCAT.

7.3. Responsabile di RT

Si occupa di monitorare l'andamento del processo di sviluppo e rilascio di componenti progettuali nel catalogo OSCAT e di gestire situazioni particolari che vanno al di fuori degli standard definiti e concordati. Il Responsabile del Progetto OSCAT si occupa anche di autorizzare e validare lo sviluppo e la consegna di plug-in di prodotti a catalogo.

7.4. Gestore della Piattaforma

Si occupa della gestione della piattaforma e dei componenti in essa presenti per quanto concerne il supporto tecnico e le modalità di rilascio.

7.5. Gestore istanze dei Componenti

Si occupa dell'installazione, della configurazione e manutenzione dei componenti e di fornire supporto allo sviluppatore di progetto nelle fasi di specializzazione e personalizzazione di particolari istanze di componenti.

7.6. Gestore delle infrastrutture

Ha il compito di fornire supporto alla realizzazione ed al deploy di nuovi servizi erogati tramite le infrastrutture di Regione Toscana, del CART e di ARPA e dell'installazione, della configurazione e manutenzione delle istanze di componenti ad essi afferenti.

8. Continuous Integration per oggetti docker

8.1. Architettura della soluzione Containerization – QMSS/Jenkins & Docker

Quanto descritto di seguito si riferisce agli ambienti di "certificazione" ospitati nei locali del TIX, dove è ormai completata una sperimentazione (quindi utilizzata in produzione) che fa uso, oltre ai componenti standard della piattaforma OSCAT/QMSS, di moduli ad hoc.

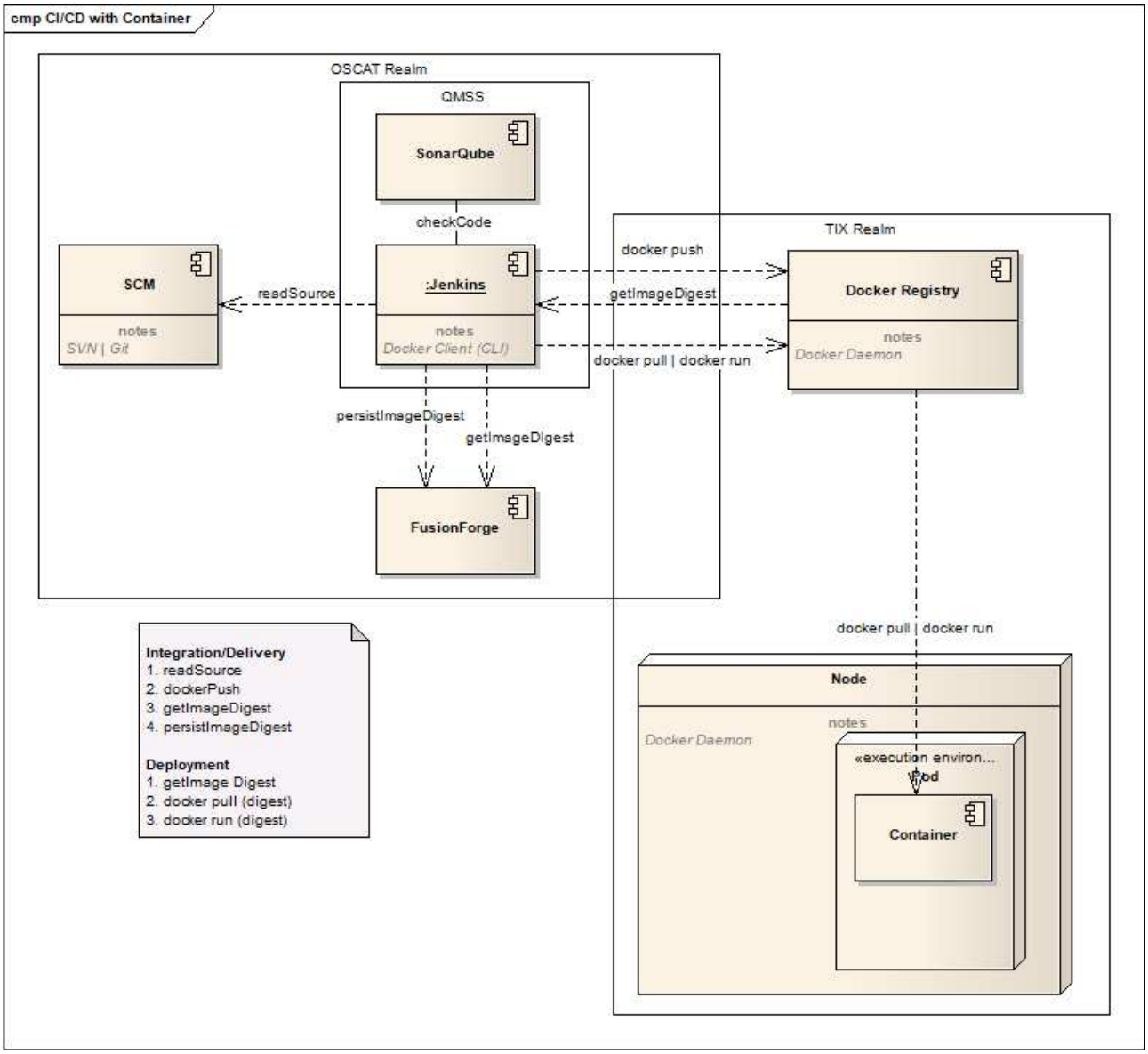
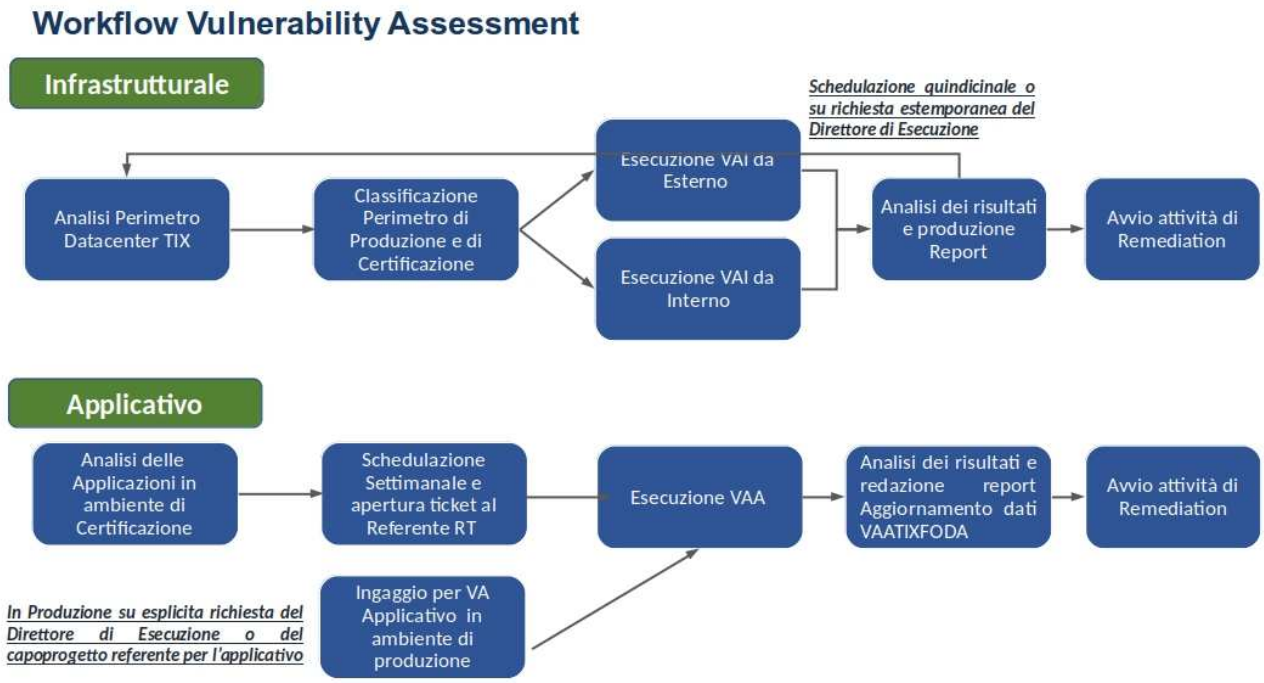


Figura 8 - QMSS/Docker. Component/Deployment Diagram

9. Scansioni VAI, VAA, VAAmobile.

9.1. Scansioni VAI, VAA

Di seguito sono presentati i workflow operativi relativi alle scansioni di tipologia infrastrutturale e di applicativi in esecuzione.



RUOLI E RESPONSABILITA' NEL PROCESSO DI GESTIONE DELLE VULNERABILITA'

Nr	ATTIVITA' 1	ATTIVITA' 2	Team VA	Team Gestione VA	NOC TIX	Resp Sicurezza RT	Capo Progetto RT	Team Applicativo
1	Invio lista subnet e applicazioni per il VA		I	C		RA		
2	Esecuzione VA e invio report vulnerabilità infrastrutturali		R	I	I	IA		
3	Esecuzione VA e invio report vulnerabilità Applicativi		R	I	I	IA	I	
4	Suddivisione e smistamento (prima analisi delle vulnerabilità, suddivisione (tipologie e priorità) e predisposizione lista ticket da aprire)			R	I	CA	I	
5	Gestione vulnerabilità							
6		Apertura ticket		R		IA		
7		Analisi report e analisi impatti			R		I	
8		Piano di Rientro		I	R	IA	C	
9		Comunicazione esito Remediation		I	R			
10		Redazione Report per SAL Mensile		R	I	IA	I	
11	Gestione vulnerabilità applicative							
12		Apertura ticket/invio mail		R		IA	I	
13		Analisi report					R	
14		Coinvolgimento Team applicativo					R	I
15		Analisi report e analisi impatti			I			R
16		Piano di Rientro		I	R	IA	C	C
17		Comunicazione esito Remediation		I		IA	R	I
18		Redazione Report per SAL Mensile		R		IA	I	
19	Monitoraggio Gestione Vulnerabilità							
20		Invio Report periodico (Trimestrale) su andamento delle Vulnerabilità e Remediation		R	I	IA		

9.2. Scansioni VAAmobile. Il laboratorio Mobisec

Regione Toscana – Giunta Regionale, tra le acquisizioni effettuate in ambito sicurezza informatica, ha acquisito da tempo una piattaforma laboratorio di dynamic mobile security in grado di verificare che le applicazioni mobile non siano esposte a vulnerabilità di design, concezione e sviluppo. Oltre alle verifiche statiche di routine, sono prevista la verifica che la struttura, la progettazione e la combinazione delle componenti dell'applicazione siano sicure e che la applicazione mobile sia stata realizzata per poter essere distribuita ed installata in un device mobile, senza rischi e vulnerabilità note.
La tecnologia del laboratorio, di marca Mobisec, le cui licenze non sono state rinnovate, è costituita di una parte Server, chiamata Station, ed una parte client, residente nei dispositivi mobili associati alla Station. Permette di effettuare in modalità automatica e semiautomatica i test di analisi di sicurezza delle applicazioni per apparati mobile iOS e android.

Di seguito viene descritto nei dettagli il laboratorio fornito elencando anche le componenti HW/SW di cui era costituito per la realizzazione on-premises del laboratorio stesso:

- Mobisec Security Analysis On Premises 1.3
- Mobisec Station fissa Appliance Apple, modello: Mac Mini i7, CPU Intel Core i7 dual-core a 3,0GHz, RAM 8GB di SDRAM LPDDR3 a 1600MHz. Accessori: Magic Mouse (Wireless), Apple Wireless Keyboard, Standard 22" lcd Monitor, Hub USB 2.0 12 porte. Baseline Software: iDeviceInstaller., Java 1.8+, Python 2.6+, Brew, MongoDB
- Apple Iphone 5S iOS 8
- Samsung Galaxy s6 Android

A corredo del laboratorio sono disponibili manuale tecnico di istruzioni e documentazione delle best practice relative alla sicurezza delle applicazioni mobili, per poter indirizzare alcuni principi di security design.

10. Processi di popolamento fonte dati AnalizzatoreDispiegamentoApplicazioniTIX e VAATIXFODA

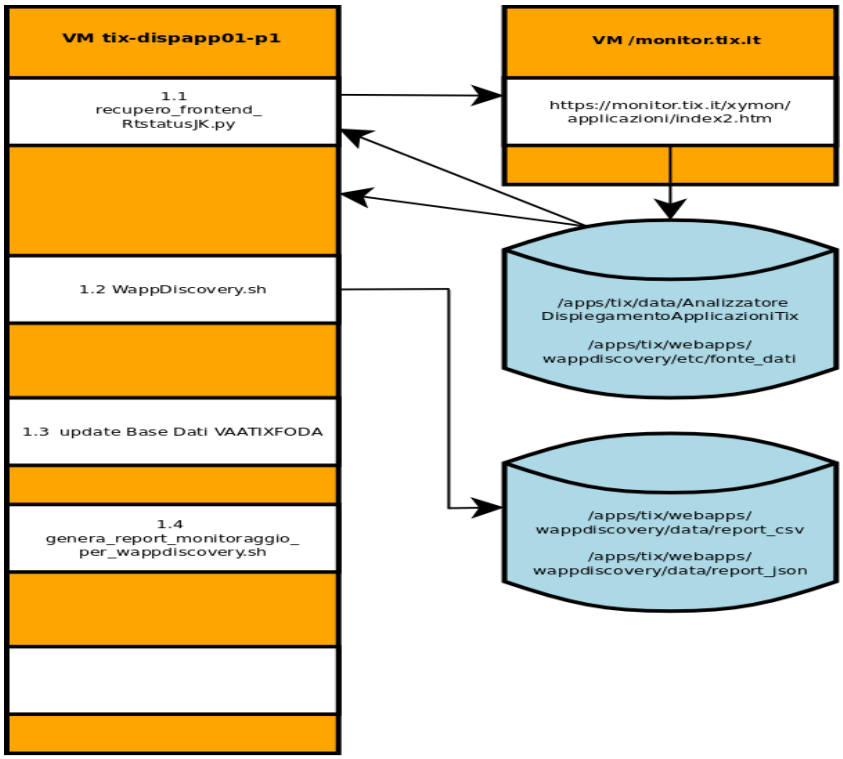
Di seguito sono elencate le componenti software in esecuzione che alimentano le basi dati utili all'esecuzone dei processi AnalizzatoreDispiegamentoApplicazioniTixe VAATIXFoDa.

Le componenti sono depositate tutte, nella loro ultima versione stabile, nel progetto oscat WappDiscovery

I processi sono eseguiti tutti sulla VM tix-dispapp01-p1 locata in rete riservata e sono nello specifico I seguenti, elencati in ordine di esecuzione:

1. recupero_frontend_RtstatuJK.py
2. WappDiscovery.sh
3. genera_report_monitoraggio_per_wappdiscovery.sh
4. AnalizzatoreDIspiegamentoApplicazioniTix.jar

di seguito uno schema di massima ed il dettaglio delle varie voci


VM tix-dispapp01-p1
1.1
recupero_frontend_
RtstatusJK.py
1.2 WappDiscovery.sh
1.3 update Base Dati VAATIXFODA
1.4
genera_report_monitoraggio_
per_wappdiscovery.sh
VM /monitor.tix.it
https://monitor.tix.it/xymon/
applicazioni/index2.htm
/apps/tix/data/Analizzatore
DispiegamentoApplicazioniTix
/apps/tix/webapps/
wappdiscovery/etc/fonte_dati
/apps/tix/webapps/
wappdiscovery/data/report_csv
/apps/tix/webapps/
wappdiscovery/data/report_json

10.1.1. recupero_frontend_RtstatuJK.py

Questo script in python viene eseguito alle ore 07:00, 13:00, 17:00 cioè un'ora prima dell'esecuzione di AnalizzatoreDispiegamentoApplicazioniTix, che è una sua dipendenza

la script interroga https://monitor.tix.it/xymon/applicazioni/index2.htm , recuperando l'elenco dei VirtualHost apache e per ognuno di essi controlla se c'e' attivo il contesto RtstatusJK, in modo da creare appunto un file contenente solo tali url

I file creati sono due : */apps/*tix/webapps/wappdiscovery/etc/fonte_dati/
/apps/tix/data/AnalizzatoreDispiegamentoApplicazioniTix/fe.list

10.1.2. WappDiscovery.sh

Questa script bash viene eseguita alle ore 08:00, 14:00, 18:00
interroga per ogni riga contenuta nel seguente file /apps/tix/webapps/wappdiscovery/etc/fonte_dati

il front-end corrispondente ed estrae le mappature JkMount creando I seguenti file:
/apps/tix/webapps/wappdiscovery/data/report_csv
/apps/tix/webapps/wappdiscovery/data/report_json
(I due file contengono quindi gli stessi dati, ma in format diverso)

questi due report sono utilizzati da dalla script per popolare la base dati di vaatixfoda attraverso le funzioni contenute in tnsnames.ora)

in questo processo viene anche identificata la natura delle Virtual Machine (se IAAS o PAAS) attraverso le info generate al punto 1.3 genera_report_monitoraggio_per_wappdiscovery.sh

10.1.3. Aggiornamento base dati VAATIXFODA

Il report CSV generato viene utilizzato come tracciato record per aggiornare la tabella DISPAPP.WEB_APP che costituisce la componente di backend dell'applicazione VAATIXFODA.

Per ogni entry del report, viene verificato con una query se la mappatura è presente o meno sulla base dati :

```
SELECT  APPID from DISPAPP.WEB_APP where APP_SERVER_IP='....' and APP_SERVER_NAME='...' and URL='…' ;
```

Se la query sopra indicata non restituisce alcun APPID, allora la mappatura non è presente e si procede ad aggiornare la base dati come nell'esempio seguente :

```
INSERT into web_app ( APPID, APP_SERVER_IP, APP_SERVER_NAME, BILANCIATO, CONTESTO, DATA_ULTIMO_AGGIORNAMENTO, NOME_APP,TIPO_GESTIONE, URL, VIRTUAL_HOST, AMBIENTE)
values (web_app_seq.NEXTVAL, '172.16.21.123'.'RT-PENTAHO-TC02-P1.rt.tix.it', 0,'https://web.rete.toscana.it/pentaho', sysdate, 'pentaho', 'non_rilevato', 'https://web.rete.toscana.it/pentaho', 'web.rete.toscana.it'.'produzione');
```

dove la sequence per l'aggiornamento del campo APPID è stata definita come di seguito :

```
CREATE SEQUENCE web_app_seq
 START WITH     2300
 INCREMENT BY   1
 NOCACHE
 NOCYCLE;
```

11. VAATIXFODA-DS - Processi di controllo congruità dei contenuti dei progetti OSCAT e la vase dati VAATIXFODA

Attraverso il job datastage vaatixfoda-ds e' verificata la presenza di oggetti nei progetti oscat che trovano corrispondenza con i nomi dei target (contesti) applicativi contenuti in vaatixfoda, e viceversa.

12. Servizio Infosharing MISP

Attualmente Regione Toscana sta conducendo una sperimentazione con il servizio opensource MISP, che permette lo scambio di di IoCs tra PA e CERT che fanno uso di servizi analoghi.

MISP usa STIX/TAXII solo per lo scambio delle informazioni immagazzinate in formato JSON ed è usata normalmente come storage di dati e correlazione di IoCs. Attualmente si riscontra un forte utilizzo da parte delle Pubbliche Amministrazioni per lo scambio di IoCs e link con le altre agenzie Europee.

Il servizio è già attivo, e per lo stesso devono essere assicurate la gestione, l'assistenza sistemistica, gli aggiornamenti, l'attività di iterazione con gli sviluppatori MISP, allo scopo di personalizzare le configurazioni richieste da Regione Toscana e necessarie per l'attività di infosharing.

13. WIKI - servizio di documentazione http://wiki-int.regione.-toscana.it

Wiki ad uso interno utilizzato da 5 redattori per la preparazioni di pagine informative tecniche.