НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

УДК 004.94

«ПОГОДЖЕНО»
Декан факультету
інформаційних технологій
Глазунова О.Г., д.п.н., професор
_____
____ _____ **202_ р.**

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»
Завідувач кафедри комп'ютерних наук
Голуб Б.Л., к.т.н., доцент
_____
___ _____ **202_ р.**

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему __ Система моніторингу сервісів кваліфікованих надавачів довірчих послуг України
Спеціальність ___ 121 інженерія програмного забезпечення_____
Освітня програма ____ Програмне забезпечення інформаційних систем_____
Орієнтація освітньої програми освітньо-професійна

Гарант освітньої програми
к.т.н, доцент _____ Голуб Б.Л
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Керівник магістерської кваліфікаційної роботи
_ д.т.н., професор _____ Хиленко Володимир Васильович
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав _____ _____Хоменко Андрій Олеговичч_____
(підпис) (ПІБ студента)

КИЇВ-2021

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри_____

_____ _____ _____

(науковий ступінь, вчене звання) (підпис) (ПІБ)

“ ” _____ 20 ___ року

З А В Д А Н Н Я

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Хоменко Андрій Олегович

(прізвище, ім'я, по батькові)

Спеціальність __інженерія програмного забезпечення 121

(код і назва)

Освітня програма ___Програмне забезпечення інформаційних систем

(назва)

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи__ Система моніторингу сервісів кваліфікованих надавачів довірчих послуг України _____

затверджена наказом ректора НУБіП України від 29.10.2020 № 1636 “С”

Термін подання завершеної роботи на кафедру 30.11.2021

Вихідні дані до магістерської кваліфікаційної роботи _____

_____

_____

_____

_____

Перелік питань, що підлягають дослідженню:

1. _____
2. _____
3. _____

Перелік графічного матеріалу (за потреби) _____

_____

Дата видачі завдання 29.10.2020р.

Керівник магістерської кваліфікаційної роботи _____ Хиленко Володимир Васильович

(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ Хоменко Андрій Олегович

(підпис) (прізвище та ініціали студент)

Оглавление

ПЕРЕЛІК ВИКОРИСТАНИХ У РОБОТІ СКОРОЧЕНЬ

АЦСК – Акредитований центр сертифікації ключів

DNS – Domain Name System – система доменних імен

SQL – Structured Query Language – мова структурованих запитів

СУБД – Система управління базою даних

KPI – Ключові показники ефективності

CSV – Comma-Separated Values – значення, розподілені комами

ТЗ – технічне завдання

JSON – JavaScript Object Notation – текстовий формат обміну даними

TSP – timestamp

OCSP – Online Certificate Status Protocol – протокол стану мережевого сертифікату

CRL – Certificate Revocation List – Списки відкликаних сертифікатів

Вступ

Багато сервісів під час карантину в Україні, переходять в електронну форму й більшість з них використовують кваліфікований електронний підпис, робота якого залежить від сервісів кваліфікованих надавачів довірчих послуг, а саме: кваліфікованої позначки часу(tsp service), онлайн перевірки статусу сертифіката користувача(ocsp service), перевірка наявності сертифікату користувача в списку відкликаних сертифікатів(crl service).

Функціонування зазначених сервісів усіх кваліфікованих надавачів електронних довірчих послуг, впливає як на роботу самих сервісів державних послуг, так і на можливість громадянам України, отримати послуги даних сервісів. Контроль і моніторинг точності даних сервісів, дозволить контролювати та повисити швидкість вирішення проблеми з сервісами кваліфікованих надавачів електронних довірчих послуг.

Вирішення проблеми актуально як для адміністраторів даних систем, так і для усіх громадян України які використовують дані служби. Метою дослідження є моніторинг стану сервісів TSP, OCSP, CRL.

Що таке електронні довірчі послуги (ЕДУ)?

Це послуги, надані певними суб'єктами (постачальниками) із взаємодії кількох користувачів, які довіряють постачальнику. Довірчий перелік, який ведеться Мін'юстом, містить перелік лише кваліфікованих продавців. Виникає можливість проводити ідентифікацію електронними засобами, полегшує функцію обміну інформацією.

До ЕДУ належать окремо або разом різні послуги, пов'язані з електронною доставкою (з фіксуванням часу передачі даних та їх захистом).

При цьому для надавачів, які є кваліфікованими в розумінні закону, встановлені певні умови: внесення коштів на відшкодування можливої шкоди в майбутньому.

Риси використання послугами інших постачальників, розподіляються за умовами договору

Як проводитиметься ідентифікація?

Ідентифікація, зокрема, проводитиметься за електронним підписом та електронним друком: простим, покращеним і кваліфікованим.

Рівень довіри до ідентифікації може бути високим (максимально запобігає зловживанням та заміні особи; забезпечується кваліфікованими підписами/печатками), середнім (істотно знижує зловживання; забезпечується удосконаленими підписами/печатками) або низьким (обмежений; знижує ризик зловживань та спростування ідентичності).

В яких сферах можна використовувати?

ЕДУ за домовленістю про порядок ідентифікації можна використовувати в будь-яких відносинах, що вимагають обміну електронними даними, за умови, що на паперових аналогах закон не вимагає їх підписання власноруч.

У разі наявності вимоги щодо обов'язковості власноручного підписання паперових носіїв, а також у відносинах із владою — учасники мають використовувати кваліфіковані ЕДУ. Відповідно органи влади використовують лише кваліфіковані сертифікати, підписи/печатки та мають додатковий порядок їх використання.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

На сайті Центрального засвідчувального органу Міністерства цифрової трансформації України, представлені дані, за 2019 та 2020 рік, за якими можна зробити висновки щодо актуальності. Наведені данні, щодо "Кількісті сформованих кваліфікованих сертифікатів електронних підписів".

Кваліфікований надавач	Загалом	З використанням алгоритму		
		ДСТУ	RSA	ECDSA
АТ "ПРИВАТБАНК"	2 579 521	2 579 518	3	0
АТ "Українська залізниця"	32 642	32 639	3	0
АТ "УкрСиббанк"	6 882	6 882	0	0
АЦСК Нацбанку	2 712	2 712	0	0
Військова частина 2428	6 338	6 338	0	0
Генеральний штаб Збройних Сил України	4 970	4 721	249	0
Державна казначейська служба України	83 396	83 396	0	0
ДП "Оператор ринку"	271	271	0	0

Рис. 1 Кількість сформованих кваліфікованих сертифікатів електронних підписів за 2019 рік

Тобто, якщо поглянути на статистику, то АТ «приватбанком», за 2019 рік було сформовано 2 мільйони 500 тисяч сертифікатів для електронних цифрових підписів. У 2020 році дана цифра збільшилась у двічі.

Кваліфікований надавач	Загалом	З використанням алгоритму		
		ДСТУ	RSA	ECDSA
АТ "ПРИВАТБАНК"	5 284 751	5 284 744	6	1
АТ "Українська залізниця"	29 677	29 669	7	1
АТ "УкрСиббанк"	92 684	92 684	0	0
АЦСК Нацбанку	1 279	1 279	0	0
Військова частина 2428	6 911	6 911	0	0
Генеральний штаб Збройних Сил України	8 394	8 296	97	1
Державна казначейська служба України	85 281	85 281	0	0
ДП "Оператор ринку"	248	248	0	0

Рис. 2 Кількість сформованих кваліфікованих сертифікатів електронних підписів за 2020 рік

Тому проблема яка є дуже актуальною, та яка не вирішується вже багато років, це централізоване відстеження працездатності сервісів усіх Акредитованих Центрів Сертифікації Ключів України. Саме які надають довірчі послуги.

Щодо сервісу TSP. Наприклад документ має бути підписаний до 31 числа, до 12-00. І якщо документ підписується у останні хвилини доби, і є розбіжність у часі, то електронний цифровий підпис на документі може вважатися не валідним. І слідом за цим можуть впроваджуватись штрафи на компанію за не вчасність завантаження звітів, а все через те, що є розбіжність у сервісі позначки часу.

Другий сервіс АЦСК, який також не менш важливий, це OCSP сервіс. Він відповідає за перевірку статусу сертифікату для ЄЦП. Тобто робочий сертифікат чи ні. Він також має позначку часу, також потрібно перевіряти правильність часу на даній позначці.

1.2 Аналіз вимог до програмної системи

Нині ситуація з сервісами кваліфікованих надавачів електронних послуг України виглядає таким чином, що кожна компанія, та кожен користувач має сам моніторити стан цих сервісів. Маючи глибокі знання у даній предметній області, та маючи навички володіння інструментарієм командного рядка, можна власноруч перевіряти стан даних сервісів. Деякі компанії, використовують звичайне автоматизоване програмне забезпечення для моніторингу. Яке можна як правило налаштувати для моніторингу будь-чого, у мережі. Але для такого моніторингу, необхідні спеціалісти, які вміють працювати з даним програмним забезпеченням, та його підтримка впродовж часу.

Тож з основних вимог до програмної системи, можна визначити наступні пункти:

- доступність (як для фізичних осіб так і для юридичних осіб)
- cross-platform software
- зручність у користуванні (головна інформація вже на першій сторінці)
- легкість та зрозумілість (щоб не виникало запитань навіть у «не продвинутих» користувачів)
- нотифікації (для швидкого trouble shooting-у підписка на нотифікації, щодо проблем у роботі сервісів, для адміністраторів даних сервісів)

1.3 Огляд інформаційних джерел та існуючих рішень

1.3.1 СИСТЕМА МОНІТОРИНГУ«ZABBIX»

ZABBIX – відкрите програмне забезпечення написане Олексієм Владишевим. Zabbix створений для моніторингу та відстеження статусів різноманітних сервісів комп'ютерної мережі, серверів та мережевого обладнання. Для зберігання даних використовується MySQL, PostgreSQL, SQLite або Oracle. Веб-інтерфейс написано на PHP. ZABBIX підтримує декілька видів моніторингу. Simplechecks може перевіряти доступність і реакцію стандартних сервісів, таких як SMTP або HTTP без встановлення будь-якого програмного забезпечення на спостерігається хості. External check – виконання зовнішніх програм. ZABBIX також підтримує моніторинг через SNMP.

Рис. 3 Панель моніторингу системи моніторингу Zabbix

Огляд можливостей Розподілений моніторинг до 1000 вузлів. Конфігурація молодших вузлів повністю контролюється старшими вузлами, що знаходяться на більш високому рівні ієрархії

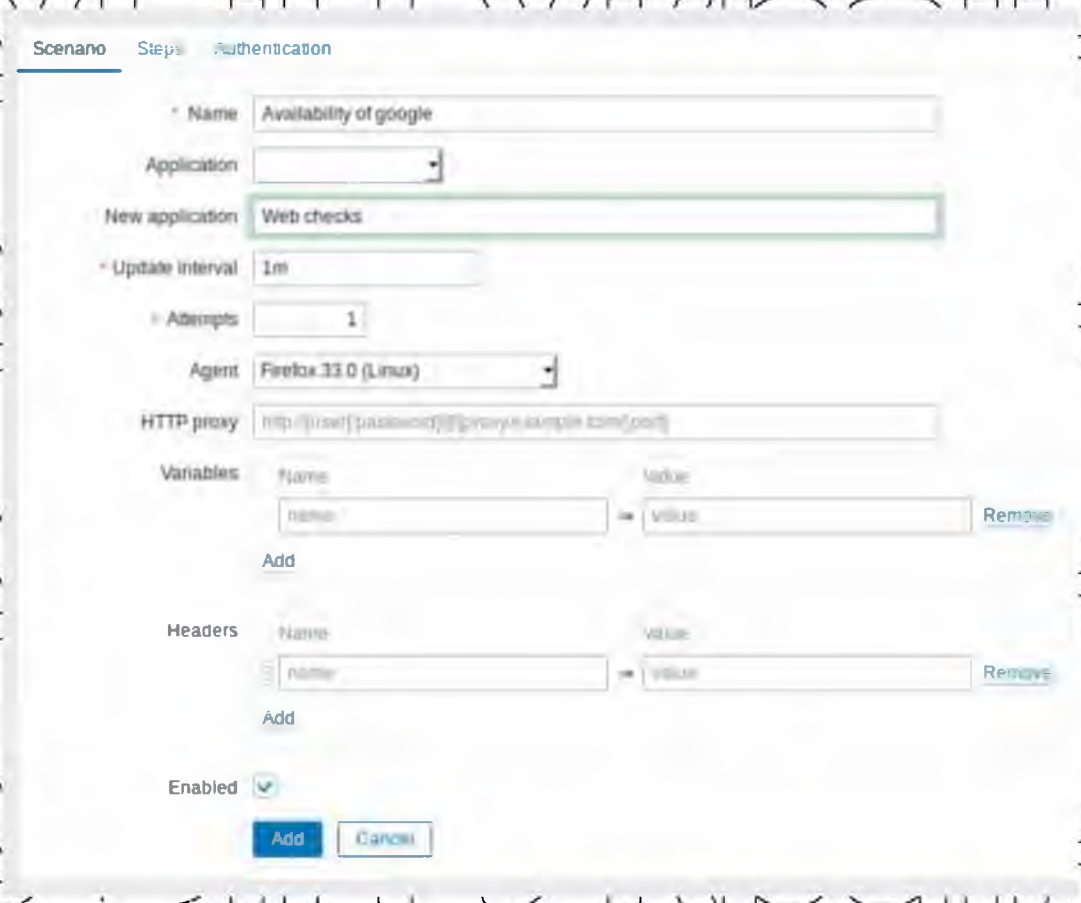


Рис. 4 Сценарій для моніторингу у системі моніторингу Zabbix

- Сценарії на основі моніторингу
- Автоматичне виявлення
- Централізований моніторинг лог-файлів
- Веб-інтерфейс для адміністрування та налаштування
- Звітність та тенденції
- SLA моніторинг

- Підтримка високопродуктивних агентів(zabbix-agent) практично для всіх платформ
- Комплексна реакція на події
- Підтримка SNMPv1,2 Гнучка система шаблонів та груп
- Можливість створювати карти мереж

Рис. 5 Графіки та візуалізація у системі моніторингу Zabbix

Автоматичне виявлення

- Автоматичне виявлення по діапазону IP-адрес, доступним сервісам і SNMP перевірка Автоматичний моніторинг виявлених пристроїв

- Автоматичне видалення відсутніх хостів розподілення по групам і шаблонам в залежності від результату, що повертається.

1.3.2 СИСТЕМА МОНИТОРИНГА «TCLMON»

TclMon (http://tclmon.vsi.ru/) - це система моніторингу обладнання, написана на платформонезалежній мові Tcl і призначена для моніторингу мереж невеликого та середнього масштабу (до декількох тисяч об'єктів). TclMon являє собою систему, що складається з сервера, що концентрує і обробляє дані, що надходять від мережевих пристроїв, і клієнта, що забезпечує візуалізацію цих даних, і працює з сервером за простим текстовим протоколом.

Це кардинально спрощує процедуру налаштування системи моніторингу - в більшості випадків при додаванні об'єкта, що підлягає моніторингу, досить просто додати у файл конфігурації базову інформацію про цей об'єкт (IP-адреса, налаштування таймерів поллінга і список методів збору, аналізу та збереження інформації), а TclMon автоматично виявить частини, складе список їх параметрів, і буде самостійно підтримувати цей список в актуальному стані при змінах апаратної та програмної конфігурації об'єкта, а також топології мережі, складовою якої цей об'єкт є.

TclMon самостійно оцінює стан об'єктів та складових їх частин, і передає свою оцінку клієнтській програмі у вигляді оповіщень про проблеми (alarm'ів). Це дозволяє виключити з клієнтської програми всю логіку оцінки ситуації, зробити її простою, універсальною і не вимагає внесення змін при додаванні нових класів об'єктів та методів збору та аналізу інформації. Взаємодія TclMon

з клієнтським додатком відбувається за простим текстовим протоколом, що працює поверх TCP, що у разі потреби легко дозволяє використовувати TclMon як джерело первинних даних про мережевих пристроях. Завдяки яскраво вираженій модульній структурі, єдиній схемі внутрішнього зберігання об'єктних даних та парадигмі "серверів" TclMon дуже легко розширюється.

Завдяки тредовой моделі TclMon забезпечує досить хорошу масштабованість - до кількох тисяч об'єктів верхнього рівня, і хороший розподіл навантаження на багатопроцесорних системах.

Поточні можливості:

Підтримка обладнання Cisco, 3Com, Allied Telesyn, D-Link, APC, Ascend (сервера доступу MAX6000), Zyxel (DSLAMи AES-100 / IES-1000 / IES-2000), Huawei (маршрутизатори серії NetEngine, комутатори серії MA5600 / MA5605), серверів з UCD-SNMP / Net-SNMP та ін. стану плат і модулів пристроїв, пулів IP-адрес, сервісів (DNS, NTP, POP3, SMTP, HTTP, FTP, NNTP, RADIUS, MySQL, Oracle) та багато іншого.

Самостійне виявлення зв'язків між об'єктами на основі аналізу описів інтерфейсів. Можливість гнучкого включення необхідних методів збору та збереження інформації для кожного окремо взятого об'єкта. Збереження даних, отриманих з об'єктів, у БД RRD. аналіз стану об'єктів і складових їх частин, і генерація оповіщень про проблеми (alarm'ів), які можуть або відправлятися по e-mail, або передаватися клієнтській програмі (передбачена можливість гнучкого управління підпискою на групи alarm'ів, що цікавлять).

Взаємодія з клієнтом за спеціальним протоколом, що забезпечує передачу даних від сервера на запит клієнта, миттєву передачу клієнту оповіщень про проблеми та виконання на стороні сервера функцій над значеннями змінних об'єктів (наприклад, побудова графіків їх зміни). Сам

TclMon поширюється у відкритих вихідних кодах під ліцензією BSD. Однак, деякі модулі, необхідні для його запуску, а також клієнтська програма Netstate поширюються тільки у вигляді готових виконуваних файлів або динамічних бібліотек.

Системні вимоги

Операційна система з підтримкою тредів (для кращої масштабованості дуже бажана підтримка тредів на рівні ядра), рекомендується Sun Solaris.Tcl 8.4. або вище, зібраний за допомогою тредів (з ключем --enable-threads). Бібліотека Thread 2.6 або вище.

Бібліотека Tcllib 1.9 або вище. RRDTool 1.2.19 або вище, зібраний за допомогою Tcl. Бібліотека Tcludp 1.0.8 або вище (потрібна лише у разі необхідності моніторингу сервісів, що працюють за протоколом UDP - DNS, NTP або RADIUS). Бібліотека Mysoltcl 3.03 або вище (потрібна лише у разі потреби моніторингу СУБД MySQL). Бібліотека Oratcl 4.4 або вище (потрібна лише у разі необхідності моніторингу СУБД Oracle або збереження даних).

Крім того, будуть потрібні деякі додаткові бібліотеки до Tcl, що поширюються лише у вигляді готових динамічних бібліотек:

- Tclsnmp - Thread-Safe SNMP-бібліотека для Tcl.
- Tclsyslog – Thread-Safe Syslog-бібліотека для Tcl.

1.3.3 СИСТЕМА МОНІТОРИНГУ «SNMPC»

SNMPc (http://snmpc.ru/) - провідний SNMP менеджер, призначений для контролю стану вашої мережі, незалежно від її розмірів та складності. Продукт надає можливість контролю та управління всією мережевою інфраструктурою, що складається з обладнання: будь-яких виробників маршрутизаторів, комутаторів, серверів та будь-яких інших пристроїв, що підтримують протокол SNMP. Навіть за відсутності необхідної бази MIB система може керувати пристроєм, створювати звіти по роботі за період багато іншого.

Основні параметри програмного комплексу:

- Безпека

Безпечне управління пристроями з використанням SNMP v3 автентифікацію ішифрування.

- Масштабованість

Можливість використання розподілених polling agent (агент опитування) та серверних частин для управління робочою групою, великим сегментом локальної мережі або розподіленою глобальною мережею.

- Пов'язаність

Можливість на E-Mail або пейджер.о Доступність Доступ до сервера з віддаленого місця за допомогою клієнта SNMPc (SNMPc Windowsclient) або WEB консолі.

- Комплексність

Автоматичний експорт побудованої карти мережі, подій, що відбулися, і збору статистики в стандартні бази даних. Навіть за відсутності необхідної бази MIB система може керувати пристроєм.

- Діяльність

Моніторинг LAN/WAN мереж та перевірка доступностей сервісів із запланованими WEB звітами знижують зайву марнотратність ресурсів і дозволяють дізнатися вузькі місця мережі.

- Адаптованість

Можливість налаштування перегляду змінних, таблиць даних, і додаткових меню. Розробка графічного вигляду пристроїв за допомогою програми BitView та різних програмних інтерфейсів.

SNMPc Enterprise забезпечує доступ до сервера через JAVA консоль або консолі програми Windows. Кожен віддалений користувач має свій рівень безпеки, що забезпечує розмежування прав управління з можливістю розмежування огляду мережі. Можливість розмежовувати область перегляду особлива корисна у великих корпоративних мережах, де за роботу різних адміністративних ділянок відповідають різні люди.

SNMPc Enterprise дозволяє розподілити механізм опитування та виявлення за допомогою використання так званих агентів опитування. Ці агенти опитують певні ділянки мережі і передають на сервер лише зміну стану або аварії. Цей механізм дуже ефективний при моніторингу ділянок мережі з обмеженою пропускною спроможністю, або що знаходяться в окремих адміністративних сегментах (наприклад, у VLAN).

Для безпеки всі консольні команди проходять через центральний сервер SNMPc і потім на відповідний агент опитування. Це дозволяє підтримувати управління мереж з IP адресами, що перехрещуються, і середовищах NAT. SNMPc Enterprise може бути впроваджений як ієрархічна система управління, що забезпечує загальний збір інформації від декількох регіональних серверів SNMPc.

Підтримується повнофункціональна архітектура зворотного успадкування, де кожен сервер SNMPc Enterprise може брати на себе функції як регіонального сервера, так і сервера верхнього рівня. Це дозволяє побудувати масштабовану, стійку до відмов архітектуру управління мережею.

SNMPc автоматично виявляє та опитує SNMP/ICMP, WEB, FTP, SMTP та TELNET сервіси, аж до 16 опитувань користувацьких TCP сервісів на елемент. Можливе опитування додатків, налаштованих на збіг "рядків відповідності" щодо опитування. Поряд з опитуванням реального часу, SNMPc

Enterprise також забезпечує звіти про доступність пристроїв, які можна переглядати через WEB.

Після встановлення та налаштування тренд звітів (звітів за період), агенти опитування стежать за всіма звітними змінними за певний період і вираховують базову лінію для типових значень. Якщо вказано відповідне налаштування, SNMPc може генерувати помилку, коли змінна перевищує вказане значення. SNMPc змінює колір пристрою об'єктів карти та робить інші дії, грунтуючись на налаштуваннях фільтрів подій. Фільтри подій можуть бути налаштовані безпосередньо при отриманні події від пристрою, по клацанню на відповідному записі в журналі подій.

Можна зробити такі дії:

- Email, Пейджер
- Програти WAV
- Запустити програму
- Переслати SNMP трап
- Вивести аварійне вікно

SNMPc Enterprise може автоматично генерувати заплановані статистичні звіти по днях, тижнях і місяцях. Формати звіту можуть бути у вигляді графіків, гистограм, розподілів та сумарної статистики . Звіти можуть бути експортовані в різні системи, включаючи принтер, файл або WEB сервер.

Налаштування звіту вкрай просте. Після вибору групи елементів виберіть звіт,звітний стиль, призначення та завдання. SNMPc Enterprise автоматично збере дані та згенерує звіт. Можливо використовувати включений у поставку додаток TrendView або виводити звіти через WEB браузер з будь-якої робочої станції. SNMPc Enterprise може автоматично

експортувати всю збережену довгострокову статистику до бази даних з використанням стандарту ODBC.

Можливо використовувати поширені засоби такі як Seagate Crystal Reports або Microsoft Access для створення тренд звітів у базі даних. Базова функціональність включає виведення топології карти, перегляд журналу подій, і виведення таблиць / графіків MIB в реальному часі. Довгострокова статистика може бути переглянута за допомогою календаря. Також включений JAVA проксі Telnet для настроювання маршрутизаторів Cisco та/або інших пристроїв.

2 Моделювання системи моніторингу сервісів кваліфікованих надавачів довірчих послуг України

2.1 Логічна модель даних

Логічна модель даних, відграє ключову роль у створенні програмного забезпечення, оскільки позначає зв’язки між даними. Логічна модель даних, може бути дещо схожа на модель бази даних, але вона не обов’язково має бути ідентична їй. Модель бази даних відображає які таблиці є, та які зв’язки між таблицями, та які відношення. А саме логічна модель даних показує лише

відношення між даними, які можуть не співпадати з таблицями баз даних. На рис. 6 зображена логічна модель даних для створюваного нами програмного забезпечення.

Тут зображені зв’язки між «лінками ацск», та «їх сервісами». А от з CRL-ями, відношення немає, оскільки це офлайн перевірка статусу сертифікатів, яка повноцінно має усі необхідні для роботи поля, та підтягує й аналізує великі обсяги даних зі сховища даних кожні півтора-два часи. Сховище даних натомість завчасно завантажує до себе ці дані, та вже далі надає їх до «Actual_CRLS».

Рис. 6 Логічна модель даних (створена на основі бази даних)

2.2 Діаграма прецедентів

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених межею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть діаграми прецедентів полягає в тому, що проєктована система подається у вигляді множини сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (англ. use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система під час діалогу з актором. При цьому нічого не говориться про те, яким чином буде реалізовано взаємодію акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації
- включення
- розширення
- узагальнення

При цьому загальні властивості варіантів використання можна подати трьома різними способами, а саме — за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації — одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується під час побудови всіх графічних моделей систем у формі канонічних діаграм.

Включення (англ. include) у мові UML — це різновид відношення залежності між базовим варіантом використання і його окремим випадком. При цьому відношенням залежності(англ. dependency) є таке відношення між двома елементами моделі, за якого зміна одного елемента (незалежного) спричиняє зміну іншого елемента (залежного).

Відношення розширення (англ. extend) визначає взаємозв'язок базового варіанту використання з іншим варіантом використання, функціональна поведінка якого залучається базовим не завжди, а тільки за виконання додаткових умов.

Діаграма прецедентів яка відображає систему моніторингу кваліфікованих надавачів електронних довірчих послуг України знаходиться на рис . На ній зображені такі актори як адміністратор електронних довірчих послуг України, користувач довірчих послуг України (звичайний громадянин, який користується послугами ЄЦП), адміністратор системи моніторингу.

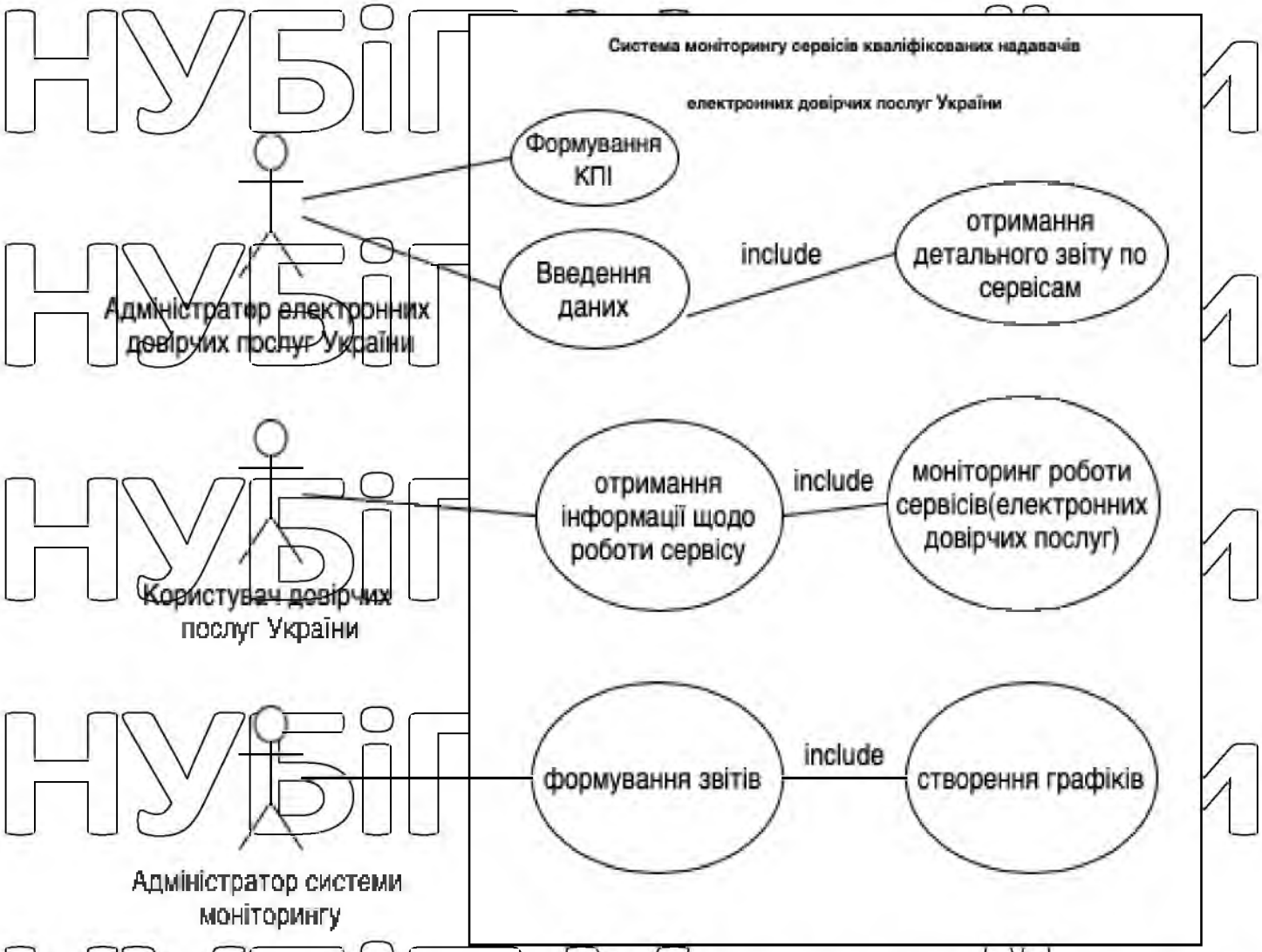


Рис. 7 Діаграма прецедентів

2.3 Топологія системи

Топологія даної системи дуже схожа на топологію систем SAN. Оскільки дані мають зберігатися як у базі, так і у сховищі даних, то дана топологія має найбільшу схожість з SAN топологією.

Мережа зберігання даних — це архітектурне рішення для підключення зовнішніх пристроїв зберігання даних, таких як дискові масиви, стрічкові бібліотеки, оптичні приводи до серверів таким чином, щоб операційна система розпізнала підключені ресурси як локальні.

Для того, щоб підвищити ефективність систем зберігання даних та можливості нарощування їхньої ємності, компанії зазвичай застосовують SAN рішення. Згідно з трактуванням Асоціації виробників мережевих засобів зберігання (Storage Networking Industry Association – SNIA), сховище даних має ознаки SAN, якщо:

1. Метою є передача даних між різними системами зберігання даних або між системами зберігання даних та серверами клієнтів. Апаратно-програмне середовище SAN містить фізичні з'єднання між системами зберігання даних та клієнтами, а також пристрої керування сховищами, сервери та мережеві пристрої. Іноді SAN визначається як провайдер введення/виведення даних.
2. Система зберігання даних, у свою чергу, містить пристрої зберігання та інші пристрої, комп'ютерне обладнання, програмне забезпечення та мережні пристрої.

Є можливість під'єднання до SAN різних пристроїв зберігання, як, наприклад, дискових підсистем, CD бібліотек, накопичувачів на магнітних стрічках та стрічкових бібліотек, і вона забезпечує обслуговування операцій введення/виведення даних за допомогою хабів або комутаторів через мережеві з'єднання.

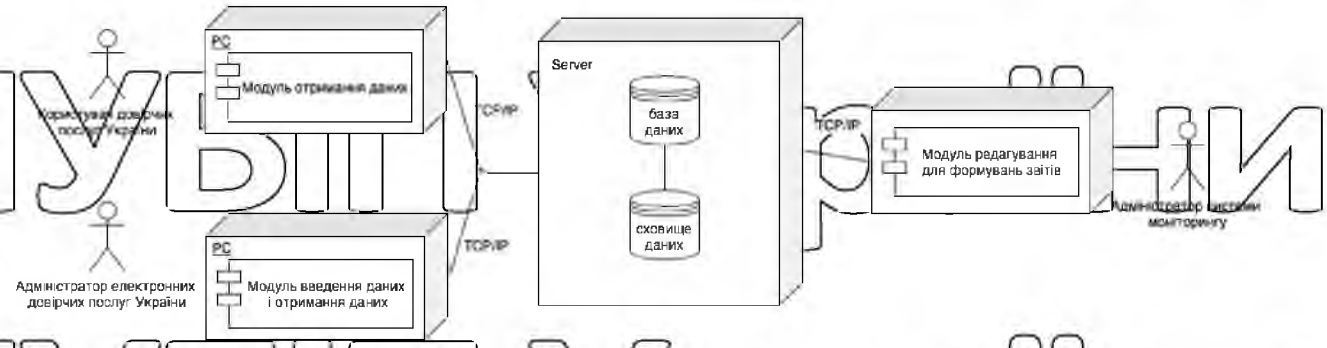


Рис. 8 Топологія системи

2.4 Вибір системи управління інформаційною базою

Перед виконанням роботи виконувався вибір між трьома системами баз даних: PostgreSQL, SQLite3 та MySQL.

PostgreSQL – об'єктно-реляційна система керування базами даних. В PostgreSQL таблиці можуть успадковувати характеристики та набори полів від інших таблиць. При цьому дані, які додаються до породженої таблиці, автоматично будуть брати участь в запитах до батьківської таблиці. Це великий плюс. Також вона є базою даних з відкритим кодом, що надає ще більше популярності для цієї системи керування базою даних.

SQLite3 - полегшена реляційна система керування базами даних. Особливістю SQLite3 є те, що вона не використовує парадигму клієнт-сервер, тобто рушій SQLite3 не є окремим процесом, з яким взаємодіє застосунок, а надає бібліотеку, з якою програма компілюється і рушій стає складовою частиною програми. Таким чином, як протокол обміну використовуються виклики функцій (API) бібліотеки SQLite3.

MySQL — вільна система управління реляційними базами даних, розроблена компанією «TCX» підвищення швидкодії обробки великих баз даних. Ця система управління базами даних (СУБД) із відкритим кодом була створена як альтернатива комерційним системам. MySQL спочатку була дуже схожа на mSQL, проте згодом вона все розширювалася і зараз MySQL – одна з найпоширеніших систем керування базами даних. Вона використовується насамперед для створення динамічних веб-сторінок, оскільки має відмінну підтримку з боку різних мов програмування.

MySQL з'явилася як спроба застосувати mSQL до своїх розробок компанії. таблиці, для яких використовувалися ISAM - підпрограми низького рівня індексного доступу до даних. В результаті було вироблено новий SQL-інтерфейс, але API-інтерфейс залишився після mSQL.

Логотип MySQL як дельфіна носить ім'я Sakila. Він був вибраний із великого списку, запропонованих користувачами "імен дельфіна". Ім'я Sakila було відправлене Open Source-розробником Ambrose Twebaze.

Тому був обраний варіант MySQL, за усі його переваги. Зручність, гнучкість, зрозуміла документація, та якість перевірена роками.

2.5 Опис вузлів системи, які поставляють дані по сховищу

Системи які поставляют дані по сховищу, це джерела даних. В системі моніторингу сервісів кваліфікованих надавачів електронних послуг України це саме вхідні дані про сервіси та їх доступність. Система перевіряє доступність сервісів tcp, за допомогою tcp/ip протоколу. Ці дані записуються та використовуються для їх аналізу та побудови графіків. Кореляція даних будується з початку на назві АЦСК(Акредитований центр сертифікації ключів). Далі вже обравши який АЦСК нас цікавить, можна подивитися доступність його сервісів, та переглянути графіки, на яких зображено повну картину за деякий період часу. На ньому зображені дані по доступності сервісів tcp, ocsp, crl. Саме ці дані, по доступності сервісів, складаються до сховища даних та утворюють факти, на основі яких будуються графіки, та кінцевий користувач бачить результат.

Ось так виглядають у коді вузли системи, які передають дані до бази даних, а з бази даних вони потрапляють до сховища.

```
if todo == 'add_crl':
    request_sql = (
        "INSERT INTO actual_crls (dt, name_acsk, name_file, last_update, next_update, link_crl) \
        VALUES (%(dt)s, %(name_acsk)s, %(name_file)s, %(last_update)s, %(next_update)s, %(link_crl)s)")
    data = {
        "dt": kwargs['create_file'],
        "name_acsk": kwargs['name_acsk'],
        "name_file": kwargs['name_file'],
        "last_update": kwargs['last_update'],
        "next_update": kwargs['next_update'],
        "link_crl": kwargs['link_crl'],
    }
```

Рис. 9 Додання даних щодо CRL сервісів

```
elif todo == 'update_crl':
    request_sql = (
        "UPDATE actual_crls SET dt = %(dt)s, last_update = %(last_update)s, next_update = %(next_update)s \
        WHERE (name_acsk = %(name_acsk)s AND name_file = %(name_file)s)")
    data = {
        "dt": kwargs['create_file'],
        "name_acsk": kwargs['name_acsk'],
        "name_file": kwargs['name_file'],
        "last_update": kwargs['last_update'],
        "next_update": kwargs['next_update'],
    }
```

Рис. 10 Оновлення даних по сервісу CRL

Таким самою чином додаються і дані про TSP, хоча їх дещо менше

```
elif todo == 'tsp_add':
    request_sql = (
        "INSERT INTO tsp (dt, name_acsk, time_tsp) VALUES (%(dt)s, %(name_acsk)s, %(time_tsp)s)")
    data = {
        "dt": kwargs['time_server'],
        "name_acsk": kwargs['name_acsk'],
        "time_tsp": kwargs['time_tsp'],
    }
else:
    return
```

Рис. 11 Додання даних по сервісу TSP

2.6 Сховище даних

Як бази даних, так і сховища даних, можуть будуватись на основі певної системи керування базами даних (СКБД) (реляційна, постреляційна тощо). СКБД забезпечує загальний репозиторій для зберігання і опрацювання структурованих даних. СКБД підтримує набір взаємозв'язаних послуг і дозволяє розробникам зосередитись на специфічних проблемах їх застосувань, а не на завданнях, які виникають при потребі в узгодженому й ефективному керуванні великими обсягами даних. Проте СКБД вимагають, щоб всі дані знаходилися під єдиним адміністративним керуванням і відповідали єдиній схемі. У відповідь на задоволення цих обмежень СКБД можуть забезпечити розвинені засоби маніпулювання даними та опрацювання запитів зі зрозумілою і строгою семантикою, а також строгі транзакційні гарантії оновлень, паралельного доступу і довготривалого зберігання (так звані властивості ACID).

Враховуючи специфіку, сховище даних має такі особливості проектування та побудови:

1) отримання інформації з різних джерел даних (у тому числі і з реляційних баз даних) у деталізованому та агрегованому вигляді (зберігаються

результати застосування функцій агрегації – суми, середнього значння, максимуму, мінімуму тощо);

2) багатовимірне подання інформації – ігноруються деякі вимоги нормалізації (дотримують максимум 3-ої нормальної форми), що значно підвищує швидкість опрацювання інформації, оскільки зменшує кількість операцій з'єднання;

3) наявність метаданих для опису джерел метаданих та структури самого сховища даних – у базах даних також використовують словники для опису структур даних, а у сховищах даних мета дані (словники, дані про дані) повинні будуватися за класифікаційною схемою Захмана. За цією схемою описую об'єкти (що?), суб'єкти (хто?), місцезнаходження (де?), час (коли?), фактори впливу, чинники (чому?), способи (як?);

4) наявність пакетного завантаження даних в сховище даних та вивантаження даних;

5) наявність процедур аналізу даних та отримання нових даних;

6) орієнтованість даних на аналітичне, а не на статичне опрацювання.

Зведення даних – предметно орієнтована, історична і унікально зв'язана множина нормалізованих таблиць, які підтримують одну або більше функціональних предметних областей. Це – гібридний підхід, що поєднує кращі особливості 3-ої нормальної форми (3НФ) і схеми «зірка». Модель гнучка, масштабується, послідовна і пристосована до потреб різних предметних областей. Вона відповідає потребам сховища даних і відкидає потребу у використанні вітрин даних та, на відміну від гібридного підходу Хекні, не вимагає подвійної роботи для надбудови архітектури шина над архітектурою корпоративної фабрики.

Зведення даних може керувати масивними наборами грануляованих даних в меншому, більш нормалізованому фізичному просторі, наприклад 3НФ і схемі «зірка». Базується на математичних принципах, які

підтримують нормалізовані моделі даних. Внутрішня частина моделі зведення даних – близькі структури, які відповідають традиційним визначення схеми «зірка» і 3НФ, що включають виміри, зв'язки багато-до багатьох і стандартні табличні структури. Відмінності полягають в подані зв'язків, структуризації поля і гранульованому,

пов'язаному з часом, зберіганні даних.

Є такі підвиди сховища даних: вітрина даних, оперативне сховище даних.

Вітрина даних (ВД) — зріз сховища даних, масив тематичної, вузькоспрямленої інформації, що орієнтований, наприклад, на користувачів однієї робочої групи або департаменту.

Дворівнева архітектура сховища даних передбачає побудову вітрин даних без створення центрального сховища, при цьому інформація надходить із реєстраційних систем і обмежена конкретною предметною областю. При побудові вітрин використовуються основні принципи побудови сховищ даних, тому їх можна вважати сховищами даних у мініатюрі.

Операційне сховище даних (ОСД) – це предметно-орієнтований, інтегрований, змінюваний набір консолідованих даних, який містить поточну (не історичну) деталізовану інформацію.

На перший погляд, операційне сховище даних дуже схоже на сховище за структурою і змістом. Зазвичай за деякими характеристиками ОСД і сховище даних дуже схожі, але ОСД має ряд властивостей, які істотно відрізняють його від сховища. Як ОСД, так і сховище даних є предметно-орієнтованим інтегрованим набором консолідованих даних. З цієї точки зору вони схожі, оскільки як в одному, так і в іншому випадку дані повинні бути завантажені з транзакційних систем. Але на цьому їх схожість закінчується. ОСД містить дані, що змінюються, тоді як в сховищі

дані після завантаження не змінюються. Інша відмінність полягає у тому, що операційне сховище містить тільки дані, актуальні на певний момент часу, тоді як в сховищі містяться як поточні, так і історичні дані. При цьому актуальність даних в сховищі значно нижча, ніж в операційному сховищі. Як правило, в сховищі містяться дані, завантажені протягом останніх 24 годин, тоді як актуальність даних в ОСД може вимірюватися секундами. Ще однією відмінністю ОСД від сховища є те, що в ньому містяться тільки детальні дані, тоді як сховище містить як детальні, так і агреговані дані.

Архітектура сховища даних розглядається з точки зору чотирьох рівнів:

- рівень користувача – описує програмний інтерфейс доступу користувачів до сховища даних;
- рівень застосувань (ППЗ) – описує засоби роботи з даними;
- рівень даних – представляє засоби для розробки структур та моделей даних, правила збереження даних, режими контролю доступу до даних;
- рівень калькуляції – містить сумарні, підсумкові дані, що полегшує та прискорює доступ до даних.

Для всіх розглянутих далі операцій потрібно першочергово з’єднати середовище «Visual Studio 2019» з «Microsoft SQL Server Management Studio».

Спочатку потрібно впевнитися, що SQL Server працює, та є доступ до бази даних. Це ми перевіримо за допомогою «Microsoft SQL Server Management Studio».

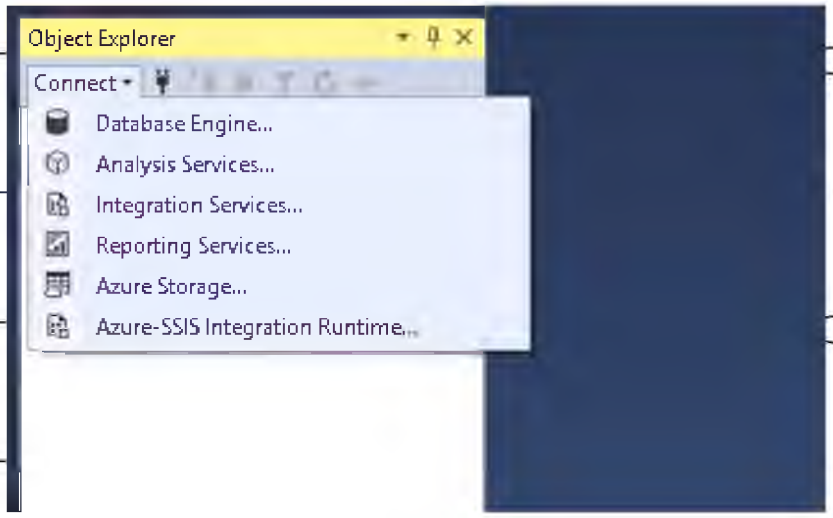


Рис 12 Підключення бази даних до серверу

Спочатку виконується приєднання до бази даних(Database Engine), а потім до системи аналізу(Analysis Services). Після успішного приєднання, маємо спостерігати таку картину.

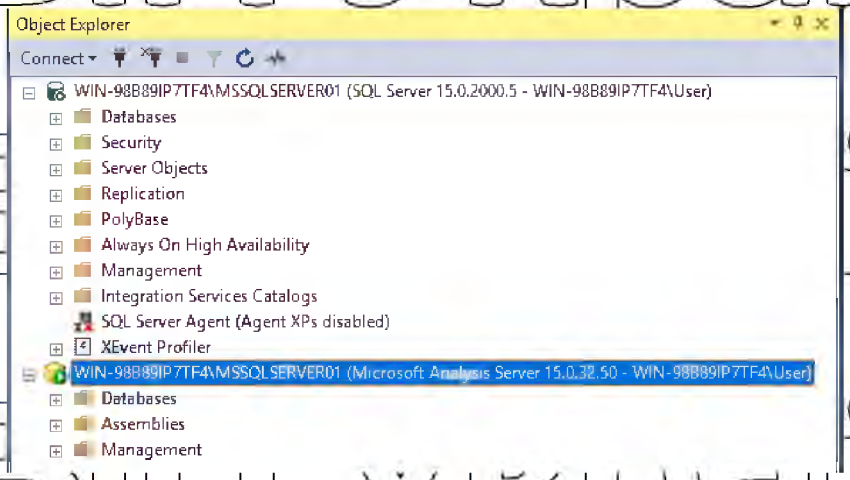


Рис. 13 Підключені база даних та система аналізу

Далі виконається підключення бази до «Visual Studio 2019».

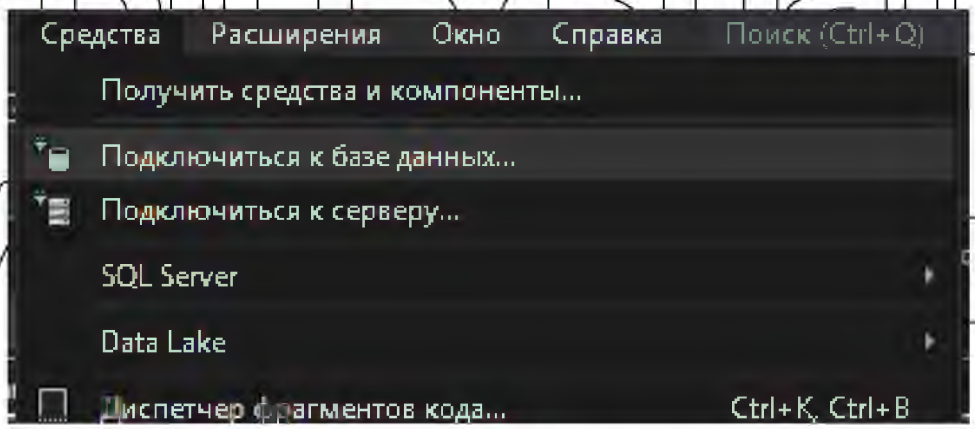


Рис. 14 Підключення бази до Visual Studio

Після підключення має з'явитись панель з підключеною базою.

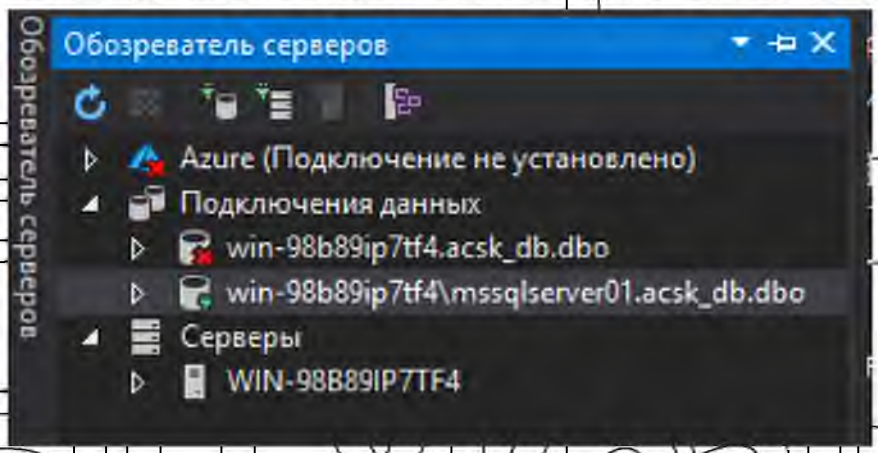


Рис. 15 перевірка підключеної бази даних

Опис BI

BI (business intelligence, інтелектуальний аналіз даних, бізнес-аналітика) - комп'ютерні методи і інструменти для організацій, що забезпечують переклад транзакційної ділової інформації в форму, придатну для бізнес-аналізу, а також засоби для роботи з обробленою таким чином інформацією.

Терміни BI і «бізнес-аналітика» найчастіше використовуються як синоніми, але між ними є різниця. Бізнес-аналітика (у вузькому розумінні), на відміну від BI, має справу з уже очищеними, підготовленими для аналізу

даними, використовує статистичні та кількісні інструменти для оцінки поточної ситуації та прогнозування, тому її все частіше називають «поглиблена аналітика».

Business Intelligence, BI спочатку займається очищенням, консолідацією даних, перетворенням їх у зручний для аналізу формат, такі завдання - інтерпретувати велику кількість даних, загострюючи увагу лише на ключових факторах, що впливають на ефективність, моделювати результат різних варіантів дій, відстежувати результати прийняття рішень. Основне призначення BI - це саме прийняття рішень для бізнесу.

BI підтримує прийняття безліч бізнес-рішень - від операційних до стратегічних. Основні операційні рішення включають в себе позиціонування продукції або цін на неї. Стратегічні бізнес-рішення включають в себе пріоритети, цілі і напрямки. BI-система найбільш ефективна, коли вона об'єднує дані, отримані з ринку, на якому працює підприємство (зовнішні дані), з даними з джерел на підприємстві, такими як фінансові та виробничі (внутрішні дані). У поєднанні зовнішні і внутрішні дані дають повнішу картину бізнесу, тобто аналітику, яку не можна отримати в результаті аналізу даних тільки від одного з цих джерел.

BI-системи розвиваються за чотирма основними напрямками:

Збереження даних. Дані в сховищі BI-системи (data warehouse, DW) структуруються спеціальним чином для більш ефективного аналізу і обробки запитів (на відміну від звичайних баз даних, де інформація організована таким чином, щоб оптимізувати час обробки поточних транзакцій).

Інтеграція даних. Для формування і підтримки сховищ даних використовуються ETL-засоби - інструменти, що забезпечують отримання даних (extract), їх перетворення (transform), тобто приведення до

необхідного формату, і завантаження (load) даних в сховище або в іншу базу.

Аналіз даних. Для всебічного аналізу даних використовуються OLAP-інструменти (on-line analytical processing). Вони дозволяють розглядати різні зрізи даних, виявляти тренди і залежності (за регіонами, продуктами, клієнтами і т.п.).

Представлення даних. Для представлення даних використовуються різні графічні засоби - звіти, графіки, діаграми. Загальноприйнятим засобом візуалізації даних є інформаційні панелі (dashboards), на яких результати відображаються у вигляді індикаторів і шкал, що дозволяють контролювати поточні значення вибраних показників, порівнювати їх з мінімально / максимально допустимими і таким чином виявляти потенційні загрози для бізнесу.

Реалізація отримання даних за допомогою Data Flow

Передачу даних з бази даних до вітрини було здійснено за допомогою Data Flow. Процес складався з двох задач.

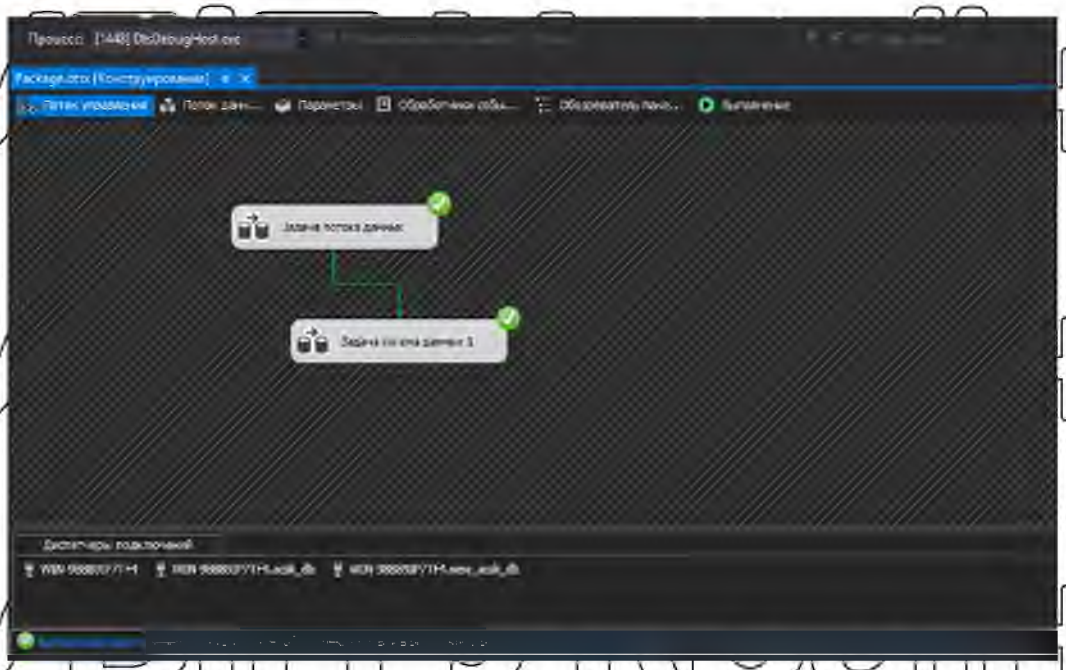


Рис. 16 Два рівні передачі даних Data Flow

У першій задачі передаються дані з таблиць Event_dim, Acsk_dim, Time_dim. Це виглядає таким чином.

Рис. 17 Передача даних вимірів

На другому рівні, дані передавалися з таблиці фактів Statistic_event_facts.

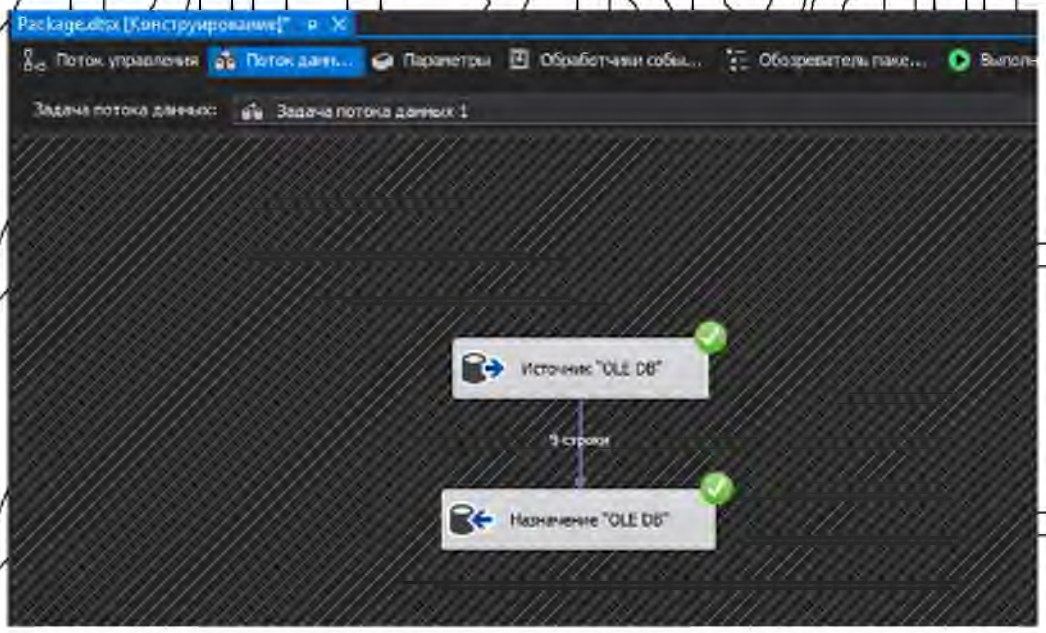


Рис. 18 Передача даних фактів

Створення BI в середовищі проекту служби SSAS

Для створення BI використовується Visual Studio. У середовищі створено проєкт бізнес-аналітики. До проєкту підключена база даних яка служить осередком для створення вітрини даних.

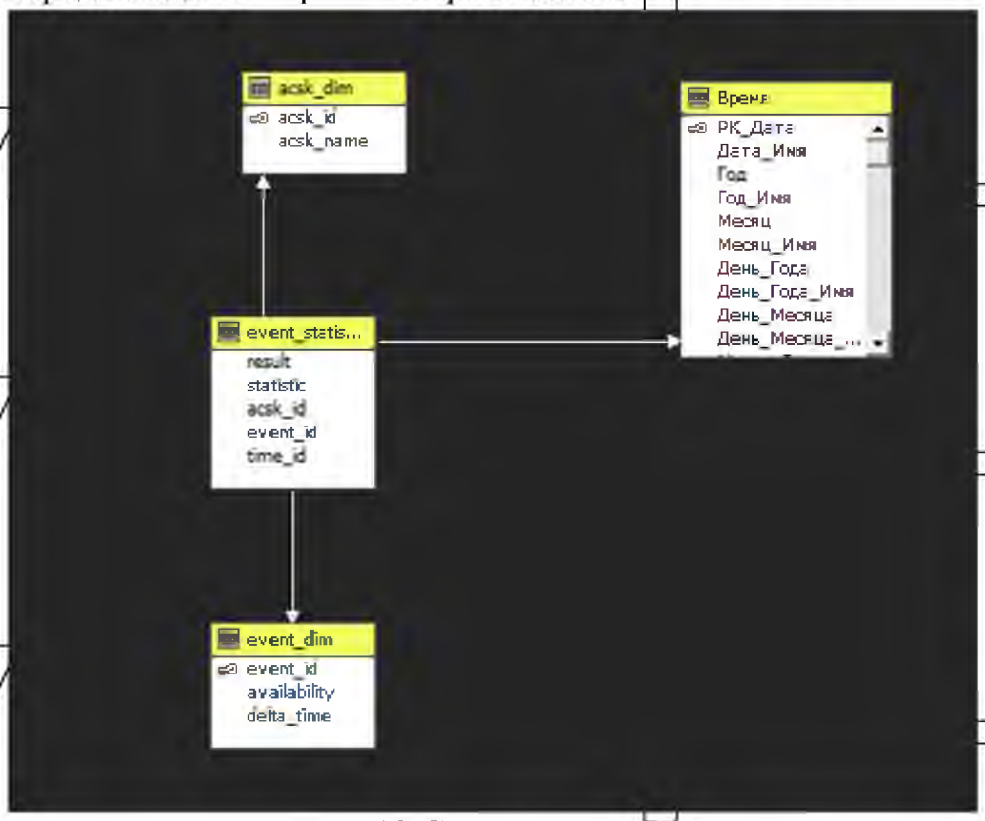


Рис. 19 Створений куб

На основі отриманої вітрини побудовано OLAP-куб із наступними вимірами: події, ацск, час, статистика подій(факти).

Розрахунки КПІ

Середня кількість падінь сервісів в день / місяць

Кожна система повинна працювати що дня, без падінь сервісів. Але все ж таки бувають ситуації коли є перебої в мережі або будь-які інші проблеми. Тому система моніторингу досліджує частоту цих падінь.

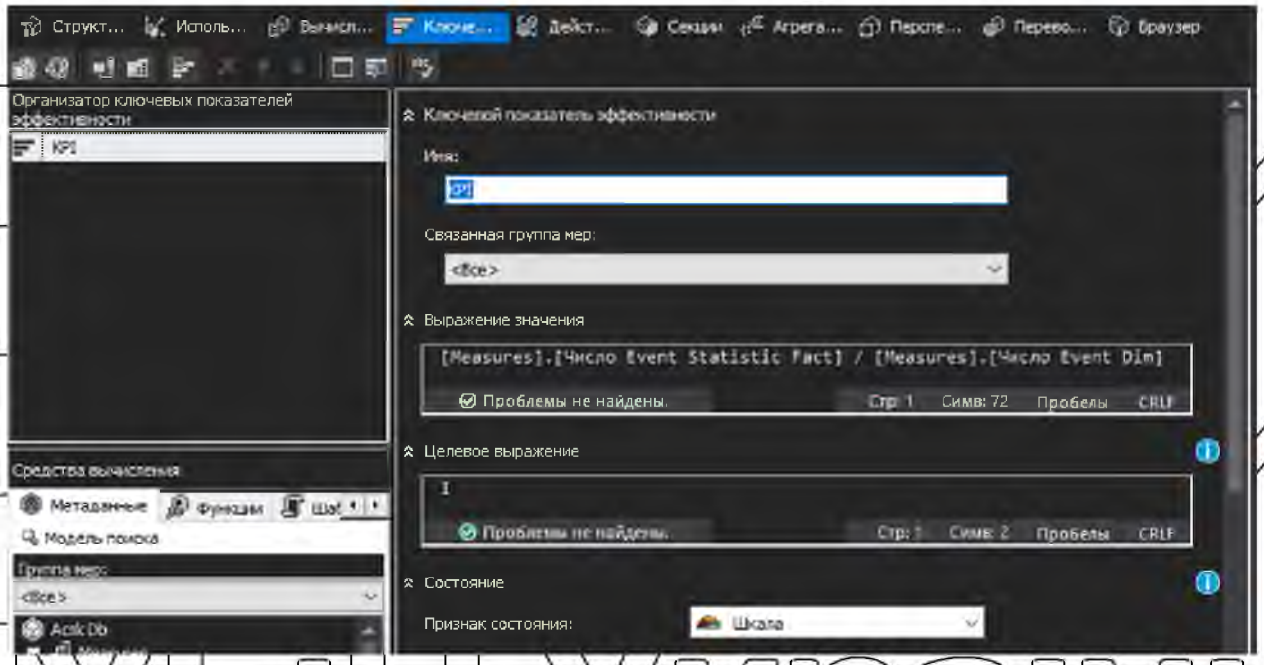


Рис. 20 Розрахунок КПІ

Для визначення середньої кількості падінь сервісів, кількість падінь поділяється на кількість днів.

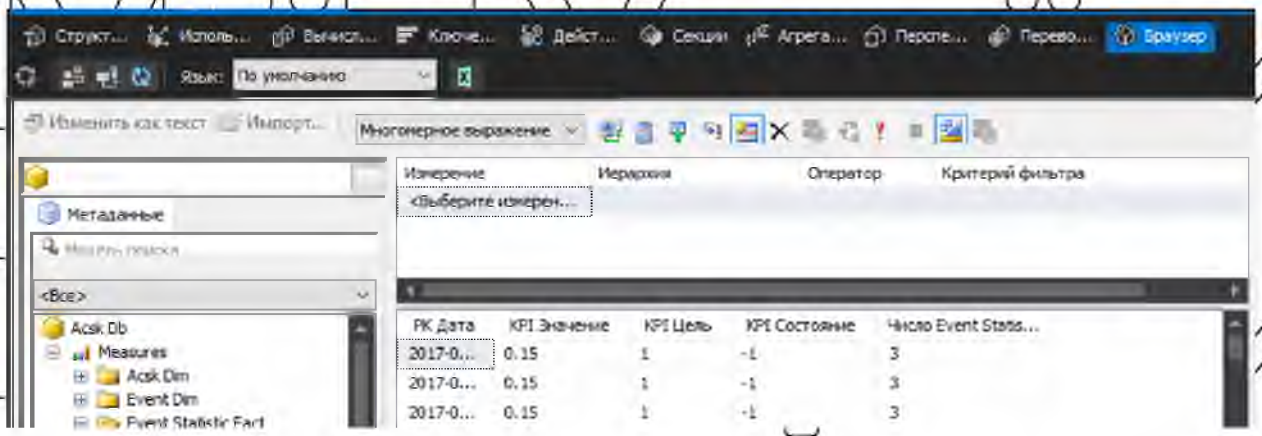


Рис. 21 Результат вирахування КПІ

3 Розробка системи моніторингу сервісів кваліфікованих надавачів довірчих послуг України

3.1 Вибір інструментарію для створення ППЗ

Серверна частина розроблювалась за допомогою стеку мов програмування python + php.

Python — це інтерпретована мова програмування загального призначення високого рівня. Його філософія дизайну підкреслює читабельність коду з використанням значних відступів. Його мовні конструкції, а також його об'єктно-орієнтований підхід спрямовані на те, щоб допомогти програмістам писати зрозумілий, логічний код для малих і великомасштабних проектів.

Python динамічно типується. Він підтримує декілька парадигм програмування, включаючи структуроване (зокрема, процедурне), об’єктно-орієнтоване та функціональне програмування. Його часто описують як мову «в комплекті з батареями» через його повну стандартну бібліотеку.

Гвідо ван Россум почав працювати над Python наприкінці 1980-х, як наступник мови програмування ABC, і вперше випустив його в 1991 році як Python 0.9.0. Python 2.0 був випущений в 2000 році і представив нові функції, такі як осмислення списків і систему збору сміття з визначенням циклу (на додаток до підрахунку посилань). Python 3.0 був випущений у 2008 році і був великою редакцією мови, яка не повністю сумісна з зворотною версією. Python 2 було припинено з версією 2.7.18 у 2020 році.

Python постійно займає місце як одна з найпопулярніших мов програмування.

Саме за ці та інші переваги, мова python була обрана щоб на ній виконувалась обробка та підрахунок часових позначок на серверній частині.

PHP — це мова сценаріїв загального призначення, орієнтована на веб-розробку. Спочатку він був створений датсько-канадським програмістом Расмусом Лердорфом у 1994 році. Довідкова реалізація PHP тепер розробляється The PHP Group. Спочатку PHP означає Personal Home Page, але тепер це означає рекурсивний ініціалізм PHP: Hypertext Preprocessor.

PHP-код зазвичай обробляється на веб-сервері інтерпретатором PHP, реалізованим у вигляді модуля, демона або виконуваного файлу Common Gateway Interface (CGI). На веб-сервері результат інтерпретованого та виконаного PHP-коду – який може бути будь-яким типом даних, наприклад, згенерованим HTML або двійковим зображенням – утворює повну або частину відповіді HTTP. Існують різні системи веб-шаблони, системи керування веб-контентом та веб-фреймворки, які можна використовувати для організації або полегшення генерації відповідної реакції. Крім того, PHP можна використовувати для багатьох завдань програмування поза веб-контекстом, таких як автономні графічні програми та роботизоване керування дроном. PHP-код також можна виконати безпосередньо з командного рядка.

Стандартний інтерпретатор PHP, що працює на платформі Zend Engine, є безкоштовним програмним забезпеченням, випущеним під ліцензією PHP. PHP був широко портований і може бути розгорнутий на більшості веб-серверів на різних операційних системах і платформах.

W3Techs повідомляє, що станом на квітень 2021 року «PHP використовується 79,2% усіх веб-сайтів, чию мову програмування на сторон сервера ми знаємо».

Спираючись на переваги мови PHP, було прийнято рішення, про використання даної мови, як частина серверної архітектури яка пов язуватиме базу даних, сховище даних, та відображення певної інформації на сторінці у веб-браузері.

Також важливо зауважити, що сервер на якому розроблюється система, знаходиться на платформі DIGITALOCEAN, за допомогою якої дуже зручно адмініструвати сервер, та займатися розробкою. Сама панель, виглядає як на рисунку 22.

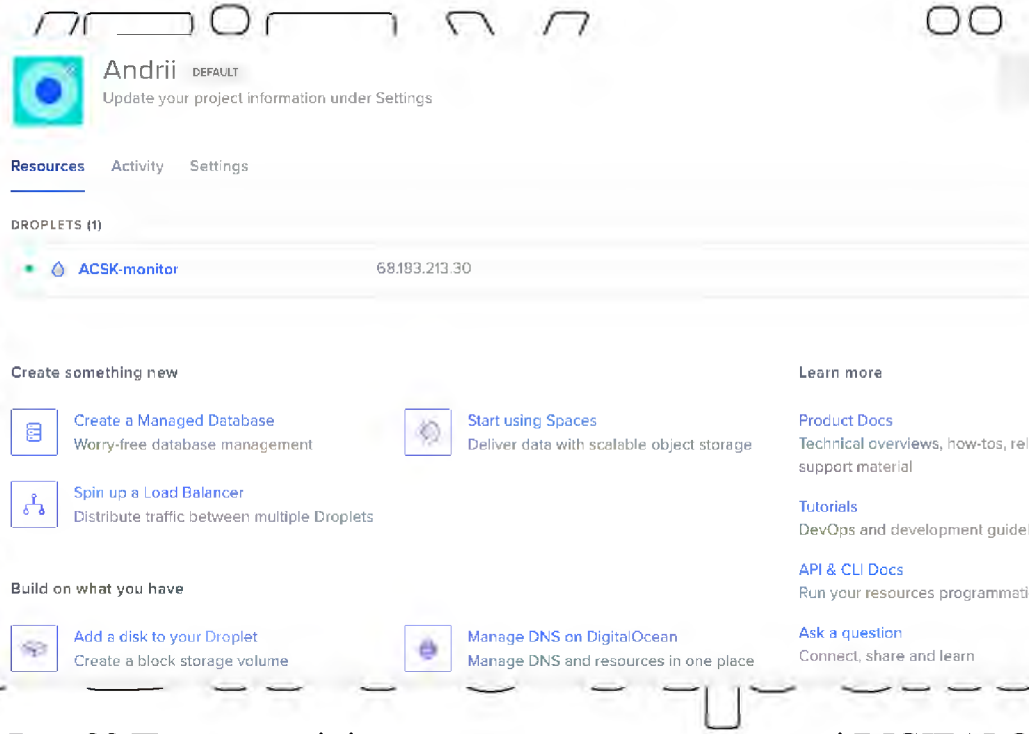


Рис. 22 Панель адміністрування сервером на ресурсі DIGITALOCEAN

Не менше уваги було приділено тому, на якому веб-сервері працюватиме дане програмне забезпечення. Було два варіанти, це NGINX та APACHE.

HTTP-сервер Apache, який у розмовній мові називається Apache, є безкоштовним багатоплатформним програмним забезпеченням веб-сервера з відкритим вихідним кодом, випущеним на умовах ліцензії Apache 2.0. Apache розробляється та підтримується відкритою спільнотою розробників під егідою Apache Software Foundation.

Переважна більшість екземплярів HTTP-сервера Apache працює в дистрибутиві Linux, але поточні версії також працюють на Microsoft Windows, OpenVMS і в широкому спектрі Unix-подібних систем. Попередні версії також працювали на NetWare, OS/2 та інших операційних системах, включаючи порти до мейнфреймів.

Спочатку заснований на сервері NCSA HTTPd, розробка Apache почалася на початку 1995 року після того, як робота над кодом NCSA зупинилася. Apache зіграв ключову роль у початковому зростанні всесвітньої мережі, швидко обігнавши NCSA HTTPd як домінуючий сервер HTTP. У 2009 році він став першим програмним забезпеченням для веб-сервера, яке обслуговує понад 100 мільйонів веб-сайтів.

Станом на січень 2021 року, за оцінками Netcraft, Apache обслуговує 24,63% з мільйона найбільш завантажених веб-сайтів, тоді як Nginx обслуговує 23,21%, а Microsoft займає третє місце з 6,85% (за деякими іншими показниками Netcraft Nginx випереджає Apache), за даними W3Techs, Apache займає перше місце з 35,0% і Nginx - друге з 33,0% і Cloudflare Server третє з 17,3%.

Nginx, є веб-сервером, який також можна використовувати як зворотний проксі, балансувальник навантаження, поштовий проксі та HTTP кеш.

Програмне забезпечення було створено Ігорем Сисоєвим і опубліковано в 2004 році. Nginx — це безкоштовне програмне забезпечення з відкритим вихідним кодом, випущене згідно з умовами ліцензії BSD з двома пунктами. Значна частина веб-серверів використовує NGINX, часто як балансувальник навантаження.

Згідно з опитуванням веб-серверів Netcraft у листопаді 2016 року, Nginx був другим за поширеністю веб-сервером на всіх «активних» сайтах (18 відсотків опитаних сайтів) і серед мільйонів найбільш завантажених сайтів (28 відсотків опитаних сайтів). За даними W3Techs, його використовували 38 відсотків із 1 мільйона найкращих веб-сайтів, 50 відсотків із 100 000 найпопулярніших веб-сайтів і 57 відсотків із 10 000 найкращих веб-сайтів. За даними BuiltWith, він використовується на 38 відсотках із 10 000 найкращих веб-сайтів, і його зростання в сегментах 10 тис., 100 тис. та 1 мільйон збільшилося.

Опитування використання Docker у 2018 році показало, що Nginx був найбільш поширеною технологією в контейнерах Docker. У OpenBSD версії 5.2 (листопад 2012 р.) Nginx став частиною базової системи OpenBSD, надаючи альтернативу системному форку Apache 1.3, який він мав намір замінити, але пізніше у версії 5.6 (листопад 2014 р.) він був видалено на користь власного httpd OpenBSD.

Проаналізувавши вищанаведені дані та статистику з відкритих джерел інформації, було обрано систему APACHE, для веб-серверу системи моніторингу сервісів кваліфікованих надавачів довірчих послуг України

Також з системних модулів які виконують основну частину роботи, наприклад відправлення POST та GET запитів, було використано бібліотеку

«Popen». Дана бібліотека надає змогу інтерпритувати команди з командного рядку системи. Це в свою чергу дало змогу роботи з інструментарієм CURL-у.

cURL (розшифровується як Client URL) - програмне забезпечення, яке надає бібліотеку libcurl та інструмент командного рядка curl. Можливості cURL величезні, багато опцій легко загубитися.

curl - інструмент для передачі даних з сервера або на нього, при цьому використовується один з протоколів, що підтримуються: DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMB, SMBS, SMTP, SMTPS, TELNET та TFTP. Команда призначена для роботи без взаємодії з користувачем.

Команда curl запускається з командного рядка і встановлена в більшості дистрибутивів Linux.

Варіанти застосування:

- доступ без браузера;
- усередині shell-скриптів;
- для тестування API.

В основному curl використовується для тестування API, іноді просто вставляючи команди, які знайшов в Інтернеті. Але розберемо curl і краще зрозуміємо його особливості.

Запит сторінки

Якщо жодні аргументи не вказані, команда curl виконує HTTP-запит get і відображає статичний вміст сторінки. Воно аналогічне тому, що ми бачимо під час перегляду вихідного коду у браузері.

curl www.google.com

Завантаження файлу

Є два варіанти цієї команди.

Завантажити файл та зберегти під оригінальним ім'ям (testfile.tar.gz).

curl -O https://testdomain.com/testfile.tar.gz

Завантажити файл та зберегти під іншим ім'ям.

curl -o custom_file.tar.gz https://testdomain.com/testfile.tar.gz

Ще можна завантажити кілька файлів однією командою, хоча так робити не рекомендують.

curl -O https://testdomain.com/testfile.tar.gz -O https://testdomain.com/testfile2.tar.gz

Отримання заголовків HTTP

Якщо необхідно подивитися, які заголовки віддає сервер, можна використовувати опції -I або -head. Вони дають змогу отримати заголовок без тіла документа.

curl -I https://www.google.com

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="Це не є P3P policy! See g.co/p3phelp for more info."
Date: Thu, 04 Jun 2020 15:07:42 GMT
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
Expires: Thu, 04 Jun 2020 15:07:42 GMT
Cache-Control: private
Set-Cookie: 1P_JAR=2020-06-04-15; expires=Sat, 04-Jul-2020 15:07:42 GMT;
path=/; domain=.google.com; Secure
Set-Cookie: <cookie_info>
```

Ігнорування помилки неправильних або самопідписаних сертифікатів

Коли виконується тестування веб-додатку або API, то у тестовому оточенні можуть бути самопідписані або неправильні сертифікати SSL. За замовчуванням curl верифікує всі сертифікати. Щоб він не видавав помилку про невірні сертифікати та встановлював з'єднання для тестування, необхідно використовувати опцію -k або -insecure.

```
curl -k https://localhost/my_test_endpoint
```

Надсилання POST-запиту

Іноді для тестування API потрібно надіслати будь-які дані, зазвичай це роблять через POST запит. Якщо виконується POST-запит за допомогою curl,

то можливо надіслати дані або у вигляді списку ім'я = значення, або у вигляді JSON.

Запит у вигляді списку им’я=значення.

curl --data "param1=test1&param2=test2" http://test.com

Запит у вигляді JSON.

curl -H 'Content-Type: application/json' --data

'{"param1":"test1","param2":"test2"}' http://www.test.com

Параметр --data еквівалентний -d, обидва вказують curl виконати HTTP POST-запит.

Вказівка типу запиту

Якщо curl не передаються жодних даних, то за замовчуванням він виконує HTTP GET запит. Але якщо, наприклад, потрібно оновити дані, а не перестворити їх заново, curl підтримує опції, що вказують тип запиту. Параметри -x або --request дозволяють вказати тип запиту HTTP, який використовується для повідомлення з сервером.

updating the value of param2 to be test 3 on the record id

curl -X 'PUT' -d '{"param1":"test1","param2":"test3"}' \http://test.com/1

Використання авторизації

Якщо просто передати логін, curl запросить пароль у командному рядку. Використовуючи параметр кілька разів для авторизації, на сервер буде передано лише останнє значення.

curl -u <user:password> https://my-test-api.com/endpoint1

Управління резольвом імен (DNS)

Якщо необхідно протестувати API перед розгортанням і перенаправити запит на тестову машину – це можна зробити, вказавши альтернативний резоль імені ендпоінта для даного запиту. Все працює еквівалентно прописуванню хоста в / etc / hosts.

curl --resolve www.test.com:80:localhost http://www.test.com/

Завантаження файлу

З опцією -F емулюється відправлення заповненої форми, коли користувач натискає кнопку відправки. Опція вказує curl передавати дані у вигляді POST-запиту, використовуючи multipart/form-data Content-Type.

curl -F @field_name=@path/to/local_file

Ви можете завантажити кілька файлів, повторюючи -F.

curl -F @field_name=@path/to/local_file
@field_name=@path/

3.2 Алгоритмізація і програмування програмних модулів

На початку правильно буде розібрати головний модуль, який відповідає саме за відправку запитів. Розберемо детально, що означають ці запити, що відправляється, та що маємо у відповідь. Звідки береться інформація щодо TSP(timestamp, позначка часу) сервісу, та звідки ми отримуємо інформацію щодо OCSP сервісу.

Таблиця у базі данних, яка відповідає за перелік АЦСК, отримує данні з веб-ресурсу ІІТ.

ІІТ – це Приватне акціонерне товариство “Інститут інформаційних технологій” (АТ "ІІТ") яке було засновано в 1995 році і спеціалізується на наданні послуг в області проектування, розробки і впровадження рішень щодо забезпечення безпеки інформації у сучасних інформаційно-телекомунікаційних системах різного призначення та різного рівня складності.

Компанія має ліцензію Держспецзв’язку України від 15.02.2017 р. на провадження господарської діяльності з надання послуг у галузі криптографічного та технічного захисту інформації (у тому числі і інформації, що становить державну таємницю).

У складі компанії створені дві групи експертів, що здійснюють експертні дослідження комплексів і засобів криптографічного захисту інформації, комплексів технічного захисту інформації та засобів захисту від НСД.

Компанія має дозвіл на здійснення діяльності, яка пов'язана з державною таємницею.

Основні напрямки діяльності

Захист інформації у інформаційно-телекомунікаційних системах:

- створення комплексних систем захисту інформації (КСЗІ);
- розробка та впровадження комплексів і засобів криптографічного захисту інформації (КЗІ);
- впровадження засобів захисту від несанкціонованого доступу (НСД);
- створення засобів та комплексів технічного захисту інформації (ТЗІ).

Експертиза:

- спеціальні дослідження комплексів технічного захисту інформації;
- експертиза комплексних систем захисту інформації та аудит інформаційної безпеки.

На сайті Інституту інформаційних технологій можна знайти JSON файл, який знаходиться за наступним посиланням, та містить у собі перелік АЦСК та посилань на їх сервіси(TSP, OCSP). Дане посилання на json файл - https://iit.com.ua/download/productfiles/CAs.json

Має даний файл структуру, яка відображена на рисунку 23.

Рис. 23 структура json файлу з переліком АЦСК

Розпарсивши даний файл, отримуються дані які будут записані у таблицю LinksACSK яка має зв'язок з таблицями TSP та OCSP.

Рис. 24 Структура бази даних

Як вже згадувалось у розділі аналізу вимог до системи, деякі перевірки які виконує система моніторингу сервісів кваліфікованих надавачів електронних довірчих послуг України, можна зробити власноруч маючи навички роботи з командним рядком, та розбираючись у предметній області довірчих послуг, та сертифікатів.

Переглянемо рис , на якому зображено виконання команд, послідовність яких, необхідно автоматизувати, та детально розберемо їх, що вони виконують, як вони це роблять, та навіщо це необхідно.

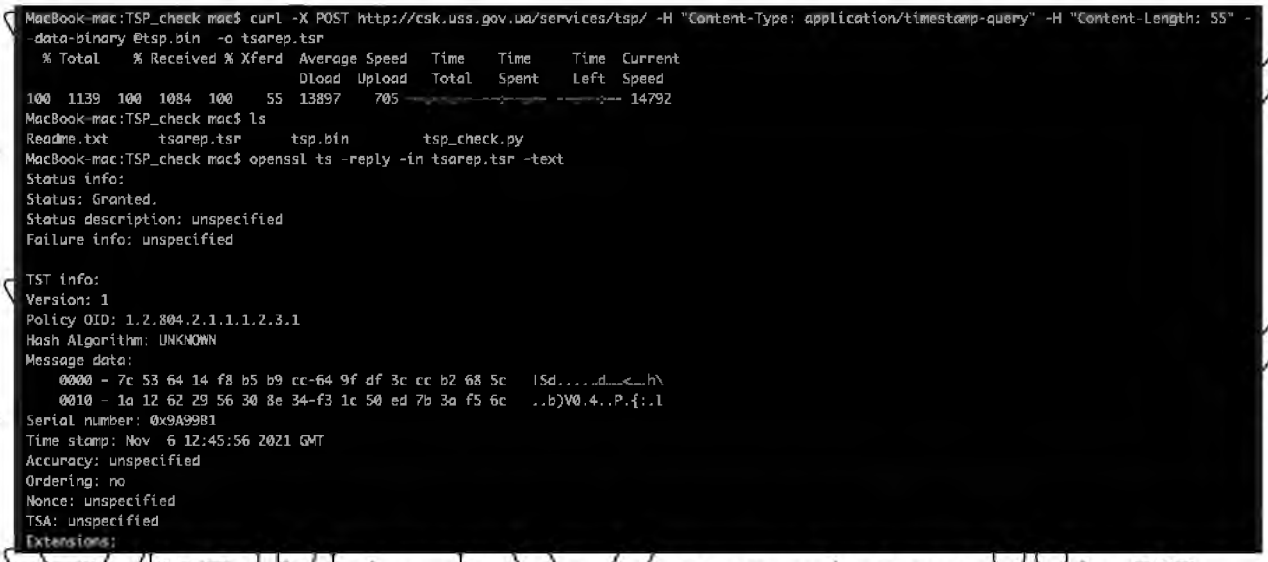


Рис. 25 Послідовність команд для отримання мітки часу з сервісу TSP (timestamp)

Перша команда виконує пост запит, по заданому URL, передаючи у заголовки «Content-type», та «Content-Length». Перший заголовок означає, що відправлені дані у POST запиті мають формат timestamp-query, довжина якого складає 55 байтів, вказано вже у другому заголовку.

Тому загальний вигляд запиту наступний :

«curl -X POST http://csk.uss.gov.ua/services/tsp/ -H "Content-Type: application/timestamp-query" -H "Content-Length: 55" --data-binary @tsp.bin -o tsarep.tsr»

Результатом виконання даної команди є інформація, щодо завантаження відповіді, яку ми зберігали флагом «-o» у запиті, а саме, зберігали до файлу tsarep.tsr.

Далі, за допомогою OPENSSL необхідно відкрити даний файл

Проект OpenSSL розробляє та підтримує програмне забезпечення OpenSSL – надійний повнофункціональний набір інструментів комерційного рівня для криптографії загального призначення та безпечного зв'язку. Прийняття технічних рішень у проекті керується Технічним комітетом OpenSSL (OTC), а управління проектом — Комітетом з управління OpenSSL (OMC). Проект діє відповідно до офіційного Статуту.

Щоб отримати додаткову інформацію про команду та спільноту навколо проекту або почати робити власний внесок, почніть зі сторінки спільноти. Щоб отримувати останні новини, завантажувати джерело тощо, перегляньте бічні панелі або кнопки у верхній частині кожної сторінки.

OpenSSL ліцензується за ліцензією в стилі Apache, що в основному означає, що ви можете вільно отримати та використовувати його в комерційних і некомерційних цілях за умови дотримання деяких простих умов ліцензії.

Виконавши команду

openssl ts -reply -in tsarep.tsr -text

ми отримуємо відповідь сервісу TSP у читаємому вигляді. Результат може бути наступним:

Status info:

Status: Granted.

Status description: unspecified

Failure info: unspecified

TST info:

Version: 1

Policy OID: 1.2.804.2.1.1.1.2.3.1

Hash Algorithm: UNKNOWN

Message data:

0000 - 7e 53 64 14 f8 b5 b9 cc-64 9f df 3e cc b2 68 5e |Sd....d..>..h^

0010 - 1a 12 62 29 56 30 8e 34-f3 1c 50 ed 7b 3a f5 6c ..b)V0.4..P.{:.l

Serial number: 0x9A99B1

Time stamp: Nov 6 12 45:56 2021 GMT

Accuracy: unspecified

Ordering: no

Nonce: unspecified

TSA: unspecified

Extensions:

Нас цікавить в даному випадку лише поле Time stamp.

Далі буде наведено, які функції виконують розбір цієї дати, у программи, та оброблюють її. Обробка позначки часу, означає що буде взята дата та час даної секунди на сервері, а далі вона буде порівнюватись з відповіддю серверу.

Листинг функції для вирахування дельти позначки часу, буде приведений нижче:

```
def delta_tsp(socket, URL_TSP):
    result_rm = Popen(['rm', '/opt/dmsu/tsarep.tsr'])

    cur_time = datetime.datetime.now()
    t_stamp = cur_time
    user_agent = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36'
    try:
        result = Popen(['curl', '-m', '5', '-A', user_agent, '-X', 'POST',
```

```
URL_TSP, '-H',
                        '\"Content-Type: application/timestamp-query\"', '-
H', '\"Content-Length: 55\"',
                        '--data-binary',
                        '@/opt/dmsu/tsp.bin', '-o', '/opt/dmsu/tsarep.tsr'],
stdout=subprocess.PIPE,
                       stderr=subprocess.PIPE,
                       universal newlines=True).communicate()[1]
    except:
        print 'bad curl timestamp-query'
    else:
        print 'good curl timestamp-query'
        result_openssl = \
            Popen(['openssl', 'ts', '-reply', '-in', '/opt/dmsu/tsarep.tsr',
'-text'], stdout=subprocess.PIPE,
                  stderr=subprocess.PIPE,
universal_newlines=True).communicate()[0]
        for line in result_openssl.splitlines():
            if line.find('Time stamp:') >= 0:
                t_stamp = parse_time_tsp(line[12:]) +
timedelta(seconds=globals.utc_offset)

    dlt_tsp = t_stamp - cur_time

    res_lines = result.splitlines()
    res_num = re.sub(r'\s+', ' ', res_lines[len(res_lines) - 1]).split(' ')

    return dlt_tsp
```

А на рисунку 26 буде представлено функцію для обробки формату часу.

```
def parse_time_tsp(str):
    format_str = '%b %d %H:%M:%S %Y GMT'  # The format
    datetime_obj = datetime.datetime.strptime(str, format_str)
    return (datetime_obj)
```

Рис. 26 Функція обробки формату часу

Також, щоб не перевантажувати сервер, було написано скрипт, для програмного звільнення пам’яті на сервері. Лістинг даного скрипта вказаний нижче:

```
#!/bin/bash

free_data="$(free)"
```

```
mem_data="$(echo "$free_data" | grep 'Mem:')"
free_mem="$(echo "$mem_data" | awk '{print $4}')"
buffers="$(echo "$mem_data" | awk '{print $6}')"
cache="$(echo "$mem_data" | awk '{print $7}')"
total_free=$((free_mem + buffers + cache))
used_swap="$(echo "$free_data" | grep 'Swap:' | awk '{print $3}')"

echo -e "Free memory:\t$total_free kB ($((total_free / 1024)) MB)\nUsed
swap:\t$used_swap kB ($((used_swap / 1024)) MB)"
if [[ $used_swap -eq 0 ]]; then
    echo "Congratulations! No swap is in use."
elif [[ $used_swap -lt $total_free ]]; then
    echo "Freeing swap..."
    sudo swapoff -a
    sudo swapon -a
else
    echo "Not enough free memory. Exiting."
    exit 1
fi
```

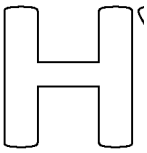
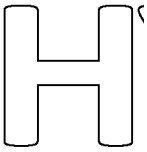
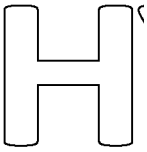
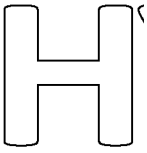
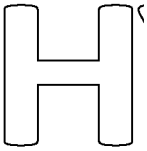
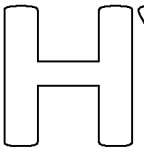
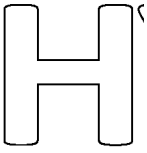
І звичайно ж модуль відображення інформації та даних, які було узято з сховища даних та підраховано в базі даних. Цей модуль був написаний на мові PHP, та виконує селекти до бази даних, за допомогою яких отримує результати та відображає їх на сторінці.

Частковий лістинг даного файлу буде наведено нижче:

```
$conn = mysqli_connect($db_host, $db_user, $db_pass, $db_name);
if (!$conn) {
    die ('Failed to connect to MySQL: ' . mysqli_connect_error());
}
```

```
$sql = 'SELECT actual_crls.name_acsk, actual_crls.name_file, actual_crls.dt
as last_time, actual_crls.last_update, actual_crls.next_update,
actual_crls.link_crl FROM actual_crls, (SELECT name_acsk, name_file, max(dt)
as last_time FROM actual_crls GROUP BY name_acsk, name_file) AS short WHERE
(actual_crls .dt=short.last_time AND actual_crls .name_acsk=short.name_acsk
AND actual_crls .name_file=short.name_file) GROUP BY name_acsk, name_file';
//$sql2 = 'SELECT name_acsk, name_file, max(dt) as last_time FROM links_crls
GROUP BY name_acsk, name_file';
$sql2 = 'SELECT actual_crls.name_acsk, actual_crls.name_file, actual_crls.dt
as last_time, actual_crls.last_update, actual_crls.next_update,
actual_crls.link_crl FROM actual_crls, (SELECT name_acsk, name_file, max(dt)
as last_time FROM actual_crls GROUP BY name_acsk, name_file) AS short WHERE
(actual_crls .dt=short.last_time AND actual_crls .name_acsk=short.name_acsk
AND actual_crls .name_file=short.name_file) GROUP BY name_acsk, name_file';
```

```
$query = mysqli_query($conn, $sql);

if (!$query) {
    die ('SQL Error: ' . mysqli_error($conn));
}
?>
    <html>
    <head>
        <meta http-equiv="refresh" content="30">
        <title>Displaying CRL Data in Table</title>

        <script type="text/javascript" language="javascript"
src="https://code.jquery.com/jquery-3.3.1.js"></script>

        <script>
            function rload($name_acsk, $name_file) {
                $.ajax({
                    url: "updatedb.php",
                    data: {name_acsk: $name_acsk, name_file: $name_file},
                    type: "POST",
                    success: function(data){
                        alert("***********CRL додано до черги на
закачування***************"); //You can remove here
                    },
                    //on error
                    error: function(){
                        alert("***********Error***************"); //You can
remove here
                    }
                });

            }
        </script>
        <script>
            function rload2($name_acsk, $name_file) {
                alert($name_file)
            }
        </script>

    </head>
    <body>
    <a href="tsp_table.php">TSP таблиця</a>

    <?php
    exec("ps aux | grep crl | wc -l", $output);
    if ($output[0] == 3) {echo "<p style=\"text-align: right\">Процес CRL
працює нормально   </p>";}
    else {echo "<p style=\"text-align: right ; color: red\">Процес CRL
зупинено   </p>";}
    ?>


    <h1>СЕРВЕР МОНІТОРИНГУ АЦСК</h1>
    <table class="data-table">
        <caption class="title">Актуальні файли списків відкликаних
сертифікатів акредитованих центрів сертифікації</caption>
```

```
            ключів
            України
        </caption>
        <thead>
        <tr>
            <th>&amp;nbsp;</th>
            <th>АЦСК</th>
            <th>Ім&amp;apos;я CRL файлу</th>
            <th>Час завантаження</th>
            <th>Час формування</th>
            <th>Час наступного</th>
            <th>Затримка</th>
            <th>Наступний СВС</th>
            <th>TSP</th>


        </tr>
        </thead>
        <tbody>
        <?php
        $no = 1;
        $total = 0;
        $tz = new DateTimeZone('Europe/Kiev');
        while ($row = mysqli_fetch_array($query)) {
            $last_update = $row['last_update'];
            $last_date = new DateTime(date('Y-m-d H:i:s',
strtotime($last_update)), $tz);

            $next_update = $row['next_update'];
            $next_date = new DateTime(date('Y-m-d H:i:s',
strtotime($next_update)), $tz);

            $dt = $row['last_time'];
            $dt_date = new DateTime(date('Y-m-d H:i:s', strtotime($dt)),
$tz);

            $interval_1 = $dt_date->diff($last_date);


            $interval = '';
            $invert = $interval_1->invert;
            $isec = $interval_1->s;
            $imin = $interval_1->i;
            $ihours = $interval_1->h;
            $iday = $interval_1->d;
            $imounth = $interval_1->m;
            $iyears = $interval_1->y;

            $invert_str = '';

            if ($invert == 0) {
                $invert_str = '-';
            } else {
                $invert_str = '+';
            }


            $interval = $invert_str;
```

4 ЕКСПЛУАТАЦІЇ СИСТЕМИ

4.1 Результати робочої системи

Розберемо приклад використання системи. На головній сторінці маємо список з переліком АЦСК. Далі обравши Саме яку АЦСК ми хочемо побачити, ми отримуємо два інформативні графіки.

На рисунку 27 представлений графік, щодо сервісу TSP. На ньому чітко видно, що час має бути рівно на позначці 1. Але деякі дані відхиляються, у відсотковому порівнянні, й це означає, що можуть виникнути проблеми з часовою позначкою тих чи інших АЦСК.

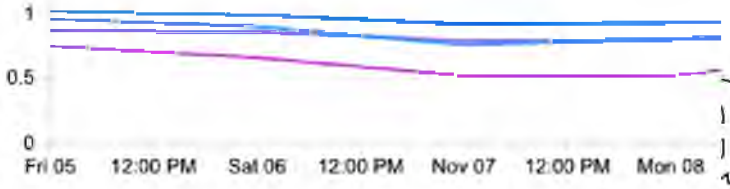


Рис. 27 Моніторинг сервісу TSP

На рисунку 28 показано приклад того, як виглядає графік стану сервісу OCSP. Даний сервіс відповідає за те, чи взагалі дійсний та робочий сертифікат користувача. Тому важливість роботи цього сервісу вище за усе, бо якщо він не відповідатиме, то користувач не зможе скористатися своїм сертифікатом, тобто не зможе скористатися довірчими електронними послугами України.

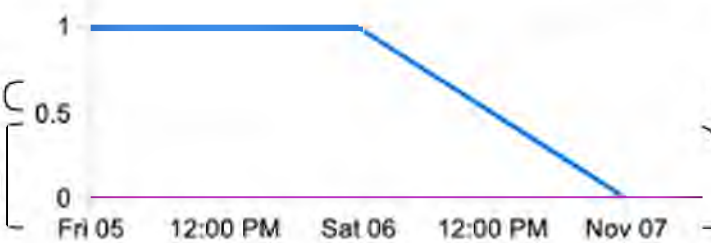


Рис. 28 Моніторинг сервісу OCSP

ВИСНОВКИ

У ході виконання магістерської роботи, було виконано аналіз предметної області, яка стосується сертифікатів, та електронних довірчих послуг України.

Також буда проведена актуалізація даної проблеми. Наведені статистичні дані за 2019 рік та за 2020. Проаналізовано, наскільки цифровізація в Україні набирає оберти, та наскільки важливе питання централізованого вирішення проблеми.

Було розібрано детально приклади вже існуючого програмного забезпечення, яке може вирішувати питання моніторингу сервісів кваліфікованих надавачів електронних довірчих послуг України. Та було визначено які вони мають недоліки, через що й прийнято рішення у розробленні програмного забезпечення моніторингу сервісів кваліфікованих надавачів електронних довірчих послуг України.

Також проаналізувавши програмні інструменти, був обраний стек інструментів які використовувалися для розробки. Це такі інструменти як:

- Python
- PHP
- MySQL
- Apache

Система моніторингу сервісів кваліфікованих надавачів електронних довірчих послуг України допомагає централізовано відстежувати усі подібні погрішності сервісів Акредитованих Центрів Сертифікації Ключів України, та пропонує зручне та гнучке, та лаконічне відображення працездатності сервісів на графіках. Також для адміністраторів АЦСК, буде введена можливість нотифікацій, щодо процаздатності їх сервісів.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Веб-сайт Інституту інформаційних технологій: [Електронний ресурс] / Режим доступу : https://iit.com.ua
2. Веб-сайт Центрального засвідчувального органу Міністерства цифрової трансформації України. [Електронний ресурс] / Режим доступу : https://czo.gov.ua
3. Довідник мови розмітки HTML: [Електронний ресурс] / Режим доступу: http://htmlbook.ru/
4. Довідник мови Python. [Електронний ресурс] / Режим доступу: https://www.python.org
5. Довідник мови PHP: [Електронний ресурс] / Режим доступу: https://www.php.net
6. Законодавство України (нормативно правові норми щодо криптографічного захисту інформації в Україні) [Електронний ресурс] / Режим доступу: https://zakon.rada.gov.ua/cgi-bin/laws/main.cgi?nreg=505%2F98#Text

ДОДАТОК А

Лістинг програми

```
result_rm= Popen(['rm', '/home/khomenko/tsarep.tsr'])

html_time = strftime("%a, %d %b %Y %H:%M:%S +0000", gmtime())

result = Popen(['curl', '-m', '5', '-X', 'POST',
'http://csk.uss.gov.ua/services/tsp/', '-H', '\"Content-Type:
application/timestamp-query\"', '-H', '\"Content-Length: 55\"', '--data-
binary', '@/home/khomenko/tsp.bin',  '-o', '/home/khomenko/tsarep.tsr'],
stdout=subprocess.PIPE, stderr=subprocess.PIPE,
universal_newlines=True).communicate()[1]

print result

result_openssl = Popen(['openssl', 'ts', '-reply','-in',
'/home/khomenko/tsarep.tsr', '-text'], stdout=subprocess.PIPE,
stderr=subprocess.PIPE, universal_newlines=True).communicate()[0]

print result_openssl
print '-----------------------------------------'
print html_time
print '-----------------------------------------'

res_lines = result.splitlines()
res_num = re.sub(r'\s+', ' ', res_lines[len(res_lines)-1]).split(' ')
tsp_len_req = res_num[-6]

print tsp_len_req
```