109 年委託開發報告

109 年度
「IPv6 動態 DNS 服務軟體開發」

IPv6 動態 DNS Server 及 Client 軟體開發報告

計畫委託機關：台灣網路資訊中心

中華民國 109 年 10 月

109 年委託開發報告

109 年度「IPv6 動態 DNS 服務軟體開發」

受委託單位

趣比比股份有限公司

計畫主持人

林韋廷

主要負責人員

張珮馨、馮俊杰、金沅禹、童冠瑜

開發時程：中華民國 109 年 5 月至 109 年 12 月

中華民國 109 年 10 月

目　　次

表　　次

圖　　次

第一章 DDNS Client

DDNS Client 負責將本機的 IP 位址更新給特定的主機，讓主機的在 DNS 內的紀錄可以更新。當使用者拿到 Client 端的原始碼之後，可以在自己的 Windows 環境下安裝好 nodejs 開發環境，之後執行指令重新編譯。

```
npm run package-win
```

由於程式是以 Electron 開發，因此如果需要修改程式，修改者本身需要懂 Electron 的程式語言開發，並同時部署好開發環境，以利於軟體的開發及測試。

第一節 DDNS Client 程式碼說明

表 1 檔案說明

檔案名稱	功能說明
main.js	主要的程式進入點跟控制程式，利用 ipcMain 與 html 頁面的 ipcRender 合作，交換資料。對於所有頁面的控制都放在這支程式內。 async function loginConnect() { 負責連線} async function getSettingsValues() {從 json 讀取設定檔案} async function saveSettingsValues() {將設定以 json 格式存入 function createLoginWindow() {產生登入視窗} async function updateIPCronJob() {更新頻率} ipcMain.on('updateIP', async(event, data) => {更新 IP} async function checkCloseApp() {關閉 app} ipcMain.on("set-secondsToUpdate", async(event, seconds1) => {設定更新頻率} ipcMain.on("quitApp", async(event) => {離開 app}

	ipcMain.on("cron", async(event) => {產生設定 cron 的視窗} ipcMain.on("login", async(event, obj) => {當 login.html 的 submit 按鈕被按下時, 由 ipcRender 觸發 ipcMain function createCronWindow() {設定多久抓一次網卡 ipv4 ipv6 } function createDashboardWindow() {顯示所有主機} function getIP() {取得主機 IP} app.on('window-all-closed', () => {關閉 APP} app.on('ready', async() => {當 app 啟動時} function checkForUpdatesPeriodically() {檢查更新頻率}
api/settings.js	負責儲存設定 async function getSettings() {取出設定} async function saveSettings(data) {儲存設定} module.exports = { settings, getSettings, saveSettings } 將設定跟函數提供給外部呼叫
cron.html	更新頻率設定
dashboard.html	顯示所有主機的列表

login.html	登入

package.json	環境所需套件資訊
	"dependencies": {
	"bootstrap": "^4.5.2",
	"electron-json-storage": "^4.2.0",
	"express": "^4.17.1",
	"express-session": "^1.17.1",
	"jquery": "^3.5.1",
	"popper.js": "^1.16.1"
	}

settings.json	設定檔

第二節 登入

使用者輸入自己的網域，例如 www.ddns.idv.tw 及您在 DDNS Server 上建立帳號跟密碼，Client 就可以正確連到 DDNS Server 並取出 zone 跟 host 清單。

圖 1 Client 登入畫面

圖 2 輸入網域跟帳號與密碼

第三節 取得 zone 及 host 清單

使用正確的網域跟帳號、密碼登入之後，DDNS Server 會回傳 zone 跟 host 清單，使用者可以在這個視窗內勾選要更改的 zone 跟 host，勾選之後，同時抓取本機的 IPv4 跟 IPv6 位址回傳給 DDNS Server，讓 DDNS Server 寫入到資料庫內。之後當使用者以 dig 指令查詢，可以正確的查詢到正確的結果。

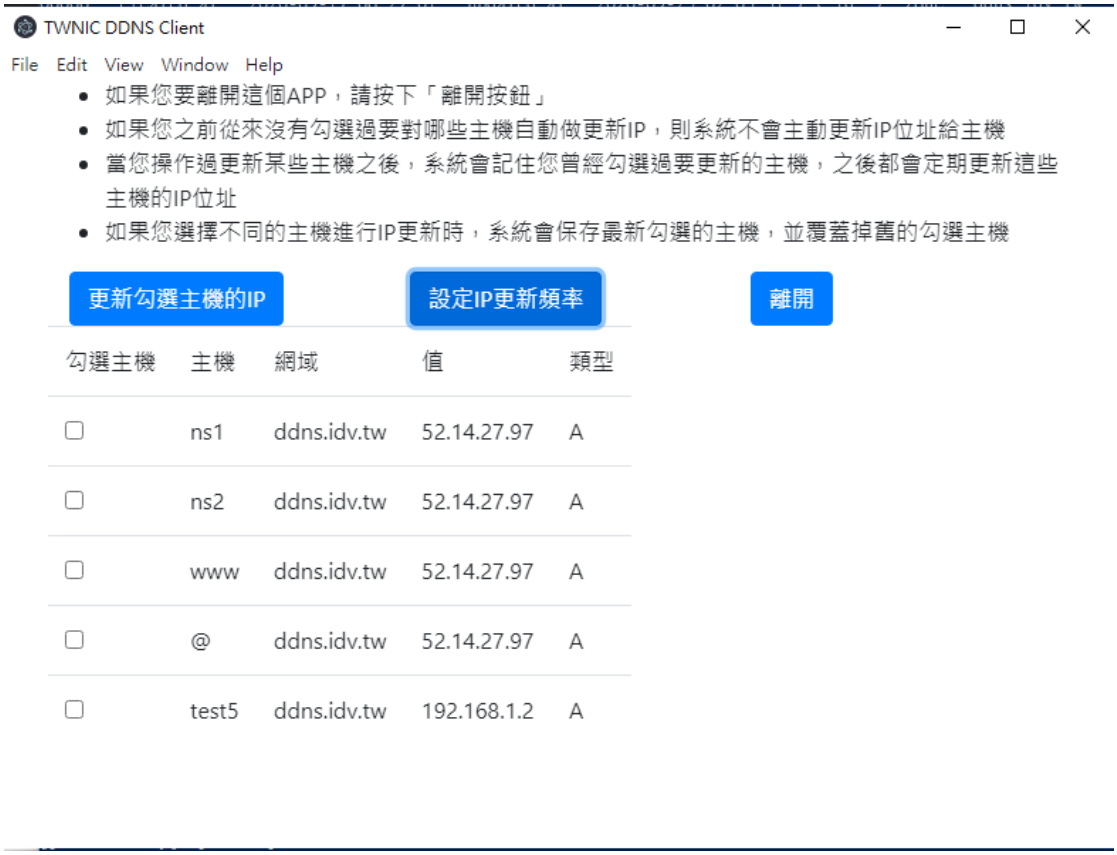

圖 3 zone 跟 host 列表

第四節 更新 zone 及 host

Client 是負責將 IPv4 跟 IPv6 位址傳給 DDNS Server，如果只有一台主機裝 DDNS Client，則會由這一台更新全部的 zone 跟 host，因此可以勾選有 chekbox 的全部主機。當有超過一台以上的 DDNS Client，而每一台 DDNS Client 負責的 zone 跟 host 都不同時，此時 DDNS Client 只需要勾選自己要更新的 zone 跟 host 即可。被勾選的主機，將使用 DDNS Client 抓取到的 IPv4 跟 IPv6 位址。

第五節 設定更新頻率

更新頻率指的是 Client 端軟體要多久抓取本地的主機 IP 位址(包括 IPv4 跟 IPv6)，抓取之後，就需要往 Server 端傳送。至於 Server 要不要更新資料庫，則由 Server 端控管。Server 會檢查 Client 的帳號密碼跟要更新的對象是否為 A 或者 AAAA，只有符合條件的情況下才可以更新，否則該行為會被跳過。系統預設為 300 秒抓一次最新的 IP 位址，使用者可以視自己的環境跟需求，變更這個數值。

圖 4 設定更新頻率

第二章 DDNS Server

DDNS Server 主要是由 PHP、MySQL 開發，並搭配 BIND9 的一套系統，藉由提供 Web 操作介面，使用者可輕鬆的管理 zone 跟 host。傳統的網域主機管理是以文字檔格式管理，現在改為以 Web 介面管理，其優點是自動檢查錯誤，不易發生錯字。另外利用 BIND9 的 DLZ 跟 MySQL 資料庫，可以做到即時的更新，而且可以減輕管理人員的負擔。另外，使用者也可以基於這套 Web 系統上繼續開發額外的功能，例如監控、稽核、帳務等功能，讓擴展功能變得很容易。

DDNS Server 是一個 Web 頁面的管理系統，並同時使用 BIND9 的 DLZ 功能，讓 BIND 使用 MySQL 當作儲存空間，而這樣的結合可以達成更新方便、擴展方便、管理方便。此外，DDNS Server 也需要有自己的帳號系統，登入系統之後可以新增 zone 跟 host，也可以管理帳號跟密碼，讓 Client 軟體可以更新指定的 zone 跟 host 的對應 IP 位址(包括 IPv4 跟 IPv6)。

第一節 DDNS Server 程式碼說明

表 2 DDNS Server 檔案說明

檔案名稱	功能說明
composer.json	環境所需套件資訊

settings.php	設定
	
	$db_host 資料庫主機
	
	$db_name 資料庫名稱
	
	$db_user 資料庫連線帳號
	
	$db_pw 資料庫登入密碼
	
	如果超過 30 分鐘沒動作，則自動登出
	
	if ($phpacl->is_admin()) {
	if (!isset($_SESSION['timestamp'])) {
	// 第一次登入
	$_SESSION['timestamp'] = time();
	} else if (time() - $_SESSION['timestamp'] > 30 * 60) {
	// 30 分鐘沒動作,就自動 logout
	header("Location: logout.php");
	exit;
	} else {
	$_SESSION['timestamp'] = time();
	}
	}

lib/lib.php	共用函式庫

	function check_password($password)檢查密碼強度 function get_ip_info($ip) 取得主機 IP function random_key() 產生隨機數 function create_dir_if_not_exist($tmp) 建立目錄 function get_current_url_base() 取得網域 function is_strong_password($password) 是否為高強度密碼 function get_client_ip()取得 client ip function generateToken()產生 CSRF 的 token 值 function checkToken($csrf_token)檢查 CSRF 的 token 是否有效 function removeBOM($str = '')移除 UTF-8 BOM
lib/phpacl.php	權限控管類別
lib/phplog.php	記錄操作行為類別 function check_ipfail_try($ip) 檢查最近 30 分鐘內的 IP 失敗次數 function check_loginfail_try 檢查最近 30 分鐘內的失敗次數

	function login 紀錄登入行為
lib/phppage.php	頁碼控制類別 function set_total 取得總數 function get_begin() 取出開始位置 function page_number() 產生頁碼
lib/install/mysql.sql	安裝資料庫的 MySQL 資料庫 schema
admin/api-login.php	提供給 Client 軟體登入的 API

admin/api-update.php	提供給 Client 軟體更新主機用的 API
	
	if ($row2['data_type'] == 'ipv4' && !empty($ipv4[0])) {
	  $sql = "update dns_records set data=:data and type='A' where id=:id";
	  $st3 = $db->prepare($sql);
	  $st3->bindParam(':id', $id, PDO::PARAM_INT);
	  $st3->bindParam(':name', $ipv4[0], PDO::PARAM_STR);
	  $st3->execute();
	
	  } else if ($row2['data_type'] == 'ipv6' && !empty($ipv6[0])) {
	  $sql = "update dns_records set data=:data and type='AAAA' where id=:id";
	  $st3 = $db->prepare($sql);
	  $st3->bindParam(':id', $id, PDO::PARAM_INT);
	  $st3->bindParam(':name', $ipv6[0], PDO::PARAM_STR);
	  $st3->execute();
	  }
	
	依據 type 是 A 還是 AAAAap

admin/hostname-add*.php	新增主機的程式
	$sql = "insert into dns_records (zone,host,type,data,ttl,mx_priority,refresh,retry,expire,minimum,serial,resp_person,primary_ns) values (:zone,:host,:type,:data,:ttl,:mx_priority,:refresh,:retry,:expire,:minimum,:serial,:resp_person,:primary_ns)";
	$st = $db->prepare($sql);
	$st->bindParam(':zone', $zone, PDO::PARAM_STR);
	// 域名
	$st->bindParam(':host', $host, PDO::PARAM_STR); //
	記錄名稱
	$st->bindParam(':type', $type, PDO::PARAM_STR); //
	記錄類型
	$st->bindParam(':data', $data, PDO::PARAM_STR); //
	記錄值
	$st->bindParam(':ttl', $ttl, PDO::PARAM_STR); //
	ttl(存活時間)
	$st->bindParam(':mx_priority', $mx_priority,
	PDO::PARAM_STR); // mx 優先級
	$st->bindParam(':refresh', $refresh,
	PDO::PARAM_STR); // 刷新時間間隔
	$st->bindParam(':retry', $retry, PDO::PARAM_STR);
	// 重試時間間隔
	$st->bindParam(':expire', $expire,
	PDO::PARAM_STR); // 過期時間
	$st->bindParam(':minimum', $minimum,
	PDO::PARAM_STR); // 最小時間

	$st->bindParam(':serial', $serial, PDO::PARAM_STR);
	// 序列號,每次更改配置都會在原來的基礎上加 1
	$st->bindParam(':resp_person', $resp_person,
	PDO::PARAM_STR); // 責任人
	$st->bindParam(':primary_ns', $primary_ns,
	PDO::PARAM_STR); // 主域名
	$st->execute();

admin/hostname-list.php	主機清單 $sql = "select * from dns_records order by created_at desc limit :offset,:rowcount"; $st = $db->prepare($sql); $st->bindParam(':offset', $begin, PDO::PARAM_INT); $st->bindParam(':rowcount', $NUMBER_PER_PAGE, PDO::PARAM_INT); $st->execute();
admin/hostname-delete.php	刪除主機的程式 $sql = "delete from dns_records where id=:id"; $st = $db->prepare($sql); $st->bindParam(':id', $id); $st->execute();
admin/hostname-update*.php	更新主機的程式 $sql = "update dns_records set zone=:zone,host=:host,type=:type,data=:data, ttl=:ttl, mx_priority=:mx_priority where id=:id"; $st = $db->prepare($sql); $st->bindParam(':zone', $zone, PDO::PARAM_STR); $st->bindParam(':host', $host, PDO::PARAM_STR); $st->bindParam(':type', $type, PDO::PARAM_STR); $st->bindParam(':data', $data, PDO::PARAM_STR); $st->bindParam(':ttl', $ttl, PDO::PARAM_STR); $st->bindParam(':mx_priority', $mx_priority, PDO::PARAM_STR);

	$st->bindParam(':id', $id, PDO::PARAM_STR); $st->execute();
admin/login*.php	登入 $sql = "select * from users where email=:email"; $st = $db->prepare($sql); $st->bindParam(':email', $email, PDO::PARAM_STR); $st->execute();

admin/logout.php	登出
	
	session_start();
	session_destroy();

admin/profile*.php	變更密碼 $sql = "update users set firstname=:firstname, lastname=:lastname, email=:email where id=:id"; $st = $db->prepare($sql); $st->bindParam(':firstname', $firstname, PDO::PARAM_STR); $st->bindParam(':lastname', $lastname, PDO::PARAM_STR); $st->bindParam(':email', $email, PDO::PARAM_STR); $st->bindParam(':id', $_SESSION['user_id'], PDO::PARAM_STR);

第二節 資料庫 Schema 範本

```
DROP TABLE IF EXISTS `apikey`;

CREATE TABLE `apikey` (
   `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
   `apikey` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
   PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT
CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;



# Dump of table db
# ------------------------------------------------------------

DROP TABLE IF EXISTS `db`;

CREATE TABLE `db` (
   `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
   `ip` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
   `title` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
   `content` text CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci,
   `created_at` timestamp NULL DEFAULT
CURRENT_TIMESTAMP,
   `checkval` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
   PRIMARY KEY (`id`),
   KEY `checkval` (`checkval`),
   KEY `ip` (`ip`),
   KEY `ctime` (`created_at`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```

```
# Dump of table dns_records
# ------------------------------------------------------------

DROP TABLE IF EXISTS `dns_records`;

CREATE TABLE `dns_records` (
   `id` int(32) NOT NULL AUTO_INCREMENT,
   `zone` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
   `host` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
   `type` enum('MX','CNAME','NS','SOA','A','PTR','AAAA','TXT')
CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci
DEFAULT NULL,
   `data` varchar(1000) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci NOT NULL DEFAULT '',
   `ttl` int(11) DEFAULT '600',
   `mx_priority` int(11) DEFAULT NULL,
   `refresh` int(11) DEFAULT '600',
   `retry` int(11) DEFAULT NULL,
   `expire` int(11) DEFAULT '86400',
   `minimum` int(11) DEFAULT '3600',
   `serial` bigint(20) DEFAULT '2020091601',
   `resp_person` varchar(1000) CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci DEFAULT '',
   `primary_ns` varchar(1000) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT '',
   `dynaload` tinyint(1) DEFAULT '0',
   `datestamp` datetime DEFAULT NULL,
   `regnumber` varchar(64) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT '0000000',
   `created_at` timestamp NULL DEFAULT
CURRENT_TIMESTAMP,
   `updated_at` timestamp NULL DEFAULT NULL ON UPDATE
CURRENT_TIMESTAMP,
```

```
  PRIMARY KEY (`id`),
  KEY `host_index` (`host`),
  KEY `zone_index` (`zone`),
  KEY `type_index` (`type`)
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT
CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;



# Dump of table email
# ------------------------------------------------------------

DROP TABLE IF EXISTS `email`;

CREATE TABLE `email` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `ip` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `sender` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `receiver` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `title` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `content` text CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci,
  `created_at` timestamp NULL DEFAULT
CURRENT_TIMESTAMP,
  `checkval` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `checkval` (`checkval`),
  KEY `ip` (`ip`),
  KEY `ctime` (`created_at`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```

```
# Dump of table inactive_zones
# ------------------------------------------------------------

DROP TABLE IF EXISTS `inactive_zones`;

CREATE TABLE `inactive_zones` (
   `zone` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci NOT NULL DEFAULT ''
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;



# Dump of table ipdata
# ------------------------------------------------------------

DROP TABLE IF EXISTS `ipdata`;

CREATE TABLE `ipdata` (
   `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
   `ip` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
   `country_code` varchar(5) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
   `locale` varchar(10) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
   `currency` varchar(10) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
   `latitude` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
   `longtiude` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
   `currency_symbol` varchar(10) CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci DEFAULT NULL,
   `in_eu` int(11) DEFAULT NULL,
   `region` varchar(50) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
```

```
  `regioncode` varchar(50) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `city` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `time_zone` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`),
  KEY `ip` (`ip`),
  KEY `country_code` (`country_code`),
  KEY `locale` (`locale`),
  KEY `currency` (`currency`),
  KEY `updated_at` (`updated_at`),
  KEY `latitude` (`latitude`),
  KEY `longtiude` (`longtiude`),
  KEY `currency_symbol` (`currency_symbol`),
  KEY `in_eu` (`in_eu`),
  KEY `region` (`region`),
  KEY `regioncode` (`regioncode`),
  KEY `city` (`city`),
  KEY `time_zone` (`time_zone`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;



# Dump of table ipfail
# ------------------------------------------------------------

DROP TABLE IF EXISTS `ipfail`;

CREATE TABLE `ipfail` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `ip` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT
CURRENT_TIMESTAMP,
```

```
  PRIMARY KEY (`id`),
  KEY `ip` (`ip`),
  KEY `ctime` (`created_at`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;



# Dump of table login
# ------------------------------------------------------------

DROP TABLE IF EXISTS `login`;

CREATE TABLE `login` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `email` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `ip` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT
CURRENT_TIMESTAMP,
  `checkval` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `checkval` (`checkval`),
  KEY `email` (`email`),
  KEY `ip` (`ip`),
  KEY `ctime` (`created_at`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT
CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;



# Dump of table loginfail
# ------------------------------------------------------------

DROP TABLE IF EXISTS `loginfail`;
```

```
CREATE TABLE `loginfail` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `email` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `ip` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT
CURRENT_TIMESTAMP,
  `checkval` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `checkval` (`checkval`),
  KEY `email` (`email`),
  KEY `ip` (`ip`),
  KEY `ctime` (`created_at`)
) ENGINE=InnoDB AUTO_INCREMENT=1783 DEFAULT
CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;



# Dump of table meta_data
# ------------------------------------------------------------

DROP TABLE IF EXISTS `meta_data`;

CREATE TABLE `meta_data` (
  `next_id` int(8) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;



# Dump of table other
# ------------------------------------------------------------

DROP TABLE IF EXISTS `other`;

CREATE TABLE `other` (
```

```
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `ip` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `title` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `content` text CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci,
  `created_at` timestamp NULL DEFAULT
CURRENT_TIMESTAMP,
  `checkval` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `checkval` (`checkval`),
  KEY `ip` (`ip`),
  KEY `ctime` (`created_at`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;



# Dump of table replication_heartbeat
# ------------------------------------------------------------

DROP TABLE IF EXISTS `replication_heartbeat`;

CREATE TABLE `replication_heartbeat` (
  `timestamp` timestamp NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;



# Dump of table users
# ------------------------------------------------------------

DROP TABLE IF EXISTS `users`;

CREATE TABLE `users` (
```

```
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `email` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `firstname` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `lastname` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `pw` varchar(250) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `avator` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT
CURRENT_TIMESTAMP,
  `temp_pw` varchar(250) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `temp_pw_created` timestamp NULL DEFAULT NULL,
  `temp_pw_expired` timestamp NULL DEFAULT NULL,
  `verify_code` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci DEFAULT NULL,
  `is_verified` tinyint(1) DEFAULT '0',
  `is_deleted` tinyint(3) unsigned DEFAULT '0',
  `is_admin` tinyint(1) unsigned DEFAULT '0',
  PRIMARY KEY (`id`),
  UNIQUE KEY `email` (`email`),
  KEY `firstname` (`firstname`),
  KEY `lastname` (`lastname`),
  KEY `is_verified` (`is_verified`),
  KEY `verify_code` (`verify_code`),
  KEY `is_deleted` (`is_deleted`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT
CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;



# Dump of table xfr_table
# ------------------------------------------------------------

DROP TABLE IF EXISTS `xfr_table`;
```

```
CREATE TABLE `xfr_table` (
   `zone` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci NOT NULL,
   `client` varchar(191) CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci NOT NULL,
   KEY `zone_client_index` (`zone`,`client`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```

第三節 登入

使用者在裝好 DDNS Server 之後，可以使用系統預設的帳號 admin 跟密碼 w123456 登入，登入之後，可以立即更新密碼。修改後的密碼建議使用高強度的密碼，以提高密碼的安全性。所謂的高強度密碼是指長度 12 個字元以上，密碼使用特殊符號、字母、數字等組合而成。

圖 5 登入畫面

第四節 新增主機

在 DDNS Server 內，要新增主機時，需要輸入包括 zone、host、type，如果是 MX，則還需要 mx priority。必填欄位不多，在資料庫內，我們都已經先將預設值填入到資料庫內，對於輕度使用者、對 DDNS 有興趣但是對 BIND 管理不熟悉的人來說，Web 操作介面簡單好用，且有控管機制，避免資料誤填。

圖 6 新增 zone 跟 host

第五節 主機列表

主機列表不僅提供所有 zone 跟 host，也提供修改跟刪除的功能，透過列表可以知道目前有多少的 zone 跟 host，當數量太多時，系統也提供分頁的功能。當需要修改時，只要點擊該 host 旁的修改按鈕，即可進行資料的修改。對於不需要的 host，則可以點擊刪除，將該 host 從資料庫中移除。

圖 7 zone 跟 host 列表

第六節 修改主機

對於已經建立在資料庫內的 zone 跟 host 需要進行調整時，此時可以利用修改的功能，將資料進行調整，調整完畢之後直接儲存到資料庫內，由於 BIND9 會直接查詢資料庫，因此新的異動也會立即生效。

圖 8 修改 zone 跟 host

例外，type 支援以下幾種：

表 3 type 的定義

欄位	說明
PTR	反解，查詢 IP 所對應的主機名稱
NS	查詢管理領域名稱 (zone) 的伺服器主機名
SOA	查詢管理領域名稱的伺服器管理資訊
A	是指 IPv4 的位址

AAAA	是指 IPv6 的位址
CNAME	CNAME 真實名稱記錄(英語:Canonical Name Record),即 CNAME 記錄,是域名系統(DNS)的一種記錄。CNAME 記錄用於將一個域名(同名)映射到另一個域名(真實名稱),域名解析伺服器遇到 CNAME 記錄會以映射到的目標重新開始查詢。這對於需要在同一個 IP 位址上運行多個服務的情況來說非常方便。
TXT	文字記錄,通常存放一些驗證用的資訊。
MX	郵件交換記錄 (MX record)是域名系統(DNS)中的一種資源記錄類型,用於指定負責處理發往收件人域名的郵件伺服器。MX 記錄允許設定一個優先次序,當多個郵件伺服器可用時,會根據該值決定投遞郵件的伺服器。簡單郵件傳輸協定(SMTP)會根據 MX 記錄的值來決定郵件的路由過程。

第七節 刪除主機

要刪除已經存在資料庫內的 zone 跟 host 時，只要進入的主機列表，就可以看到全部的主機，並選擇要刪除的主機，點擊刪除即可。此刪除行為是立即發生，因為會直接刪除資料庫內的值，也會影響 BIND9 的查詢結果。

第八節 新增帳號

管理者帳號 admin 具有最大權限，為方便 Client 連線使用，我們提供新增普通使用者帳號功能。以管理者帳號登入系統後，可以進行包括主機、帳號的變更，但是以一般使用者登入系統後，則不能具備對主機跟帳號進行任何操作，只能修改自己的密碼。因此對於 Client 端，提供一般使用者帳號供登入做更新即可。

圖 9 新增帳號

第九節 帳號列表

當帳號新增太多時，此時帳號列表就很實用，因為可以從列表中對於現存帳號進行管理，可執行的動作包括刪除跟修改。

圖 10 帳號列表

第十節 變更密碼

DDNS Server 預設有一個組帳號跟密碼，但在使用時，建議不定期更換密碼，以確保系統的安全性。

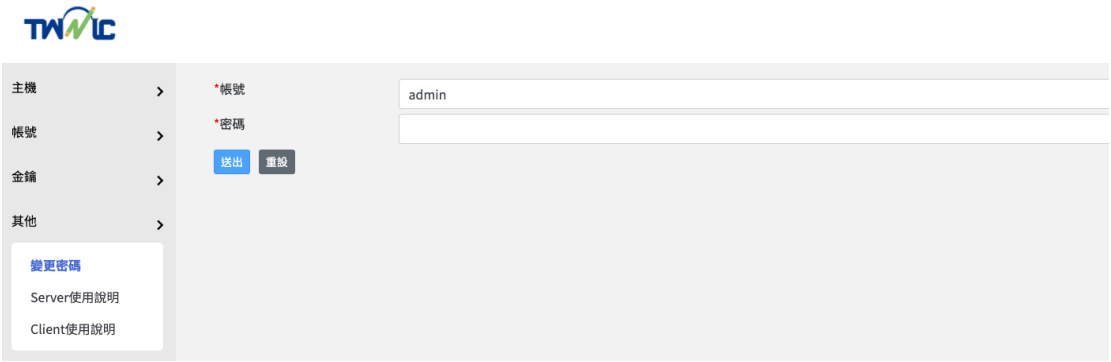

圖 11 變更密碼

第十一節 新增 API 金鑰

金鑰(API KEY)是可以取代以帳號跟密碼登入的另外一個選擇方案，透過 API KEY 登入的優點是 KEY 的長度很長，沒有規則性、無法用暴力攻擊的方式取得 API KEY。另外 API KEY 可以隨意產生，再提供給 DDNS Client 使用，如果要變更，只要把舊的 API KEY 砍掉，再產生一組新的 KEY 即可。

圖 12 新增 API 金鑰

第十二節 API 金鑰列表

由於可以產生多組 API 金鑰，為方便管理者控管，提供 AP 金鑰列表，並在列表中提供刪除金鑰功能。

圖 13 API 金鑰列表

第十三節 Server 使用說明

提供 Server 使用手冊，此手冊為一個可下載的檔案，方便使用者可以一邊閱讀再一邊操作。

第十四節 Client 使用說明

提供 Client 使用手冊，此手冊為一個可下載的檔案，方便使用者可以一邊閱讀再一邊操作。

第三章 軟體安全測試報告

第一節 原始碼掃描報告

執行時間：2020 年 11 月 30 日。使用 phpstan(版本 0.12.58)進行原始碼掃描，掃描原始碼結果如下，無安全漏洞。

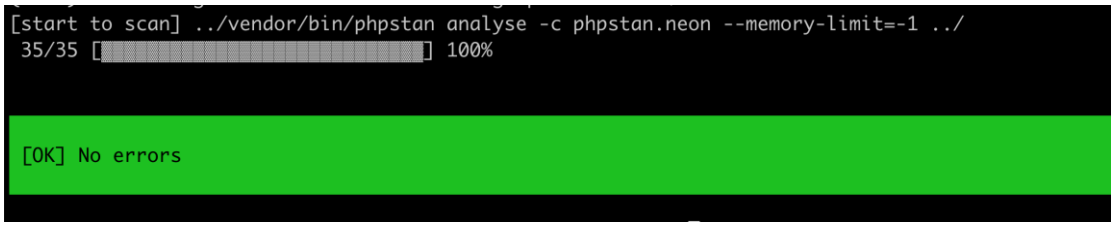

圖 14 原始碼掃描

第二節 滲透測試報告

執行時間：2020 年 11 月 30 日(ZAP 版本為 2.9.0) ，無安全漏洞。

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	0
Low	2
Informational	2

Alert Detail

Low (Medium)	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Description	The web/application server is leaking information via one or more "X-Powered-By" HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to.
URL	http://ipv6ddns.test:8000/admin/index.php
Method	GET
Evidence	X-Powered-By: PHP/7.4.11
URL	http://ipv6ddns.test:8000/admin/login.php
Method	GET
Evidence	X-Powered-By: PHP/7.4.11
URL	http://ipv6ddns.test:8000/admin/css/style.css?t=1606656099
Method	GET

圖 15 滲透測試報告

第三節 弱點掃描

使用 Wapiti 3.0.3 版本於民國 109 年 12 月 1 日進行測試，測試結果沒有發現安全漏洞。

Wapiti vulnerability report

Target: http://ipv6ddns.test:8000/

Date of the scan: Tue, 01 Dec 2020 08:34:33 +0000. Scope of the scan: folder

Summary

Category	Number of vulnerabilities found
SQL注入	0
SQL盲注	0
文件处理	0
跨站脚本	0
CRLF注入	0
命令注入	0
Htaccess绕过	0
备份文件	0
潜在危险文件	0
Server Side Request Forgery	0
Open Redirect	0
XXE	0
内部服务器错误	0
资源消耗	1

资源消耗

Description

Resource consumption description

Anomaly found in /admin/login1.php

Description | HTTP Request | cURL command line

请求超时：尝试往参数csrf_token注释payload

Solutions

涉及的脚本可能以低效的方式使用服务器资源（CPU，内存，网络，文件访问...）

References

- http://www.owasp.org/index.php/Asymmetric_resource_consumption_(amplification)
- CWE-400: Uncontrolled Resource Consumption ('Resource Exhaustion')

Wapiti 3.0.3 © Nicolas SURRIBAS 2006-2020

圖 16 弱點掃描