УДК 004.056:004.031.4

Д. М. Самойленко
Національний університет кораблебудування імені адмірала Макарова

ГНУЧКЕ УПЕРЕДЖЕННЯ МЕРЕЖНИХ АТАК

Для побудови системи захисту інформаційних ресурсів від мережних атак та розвідок необхідно аналізувати технології їх здійснення. Наявні засоби виявлення атак переважно ґрунтуються на моніторингу комплексу показників щодо стану функціонування системи чи вимагають додаткових відомостей про особливості побудови ресурсу чи мережі. Засоби протидії атакам переважно полягають в ігноруванні потенційно небезпечних дій. Це обмежує область використання існуючих засобів виявлення та упередження атак, зокрема для ресурсів з обмеженим доступом до системних показників чи серверних параметрів. Запропоновано методику створення гнучких захисних рішень, головна відмінність яких полягає в імітуванні вразливості інформаційного ресурсу з подальшим моніторингом дій користувача. Впорядковано мережні атаки у співвіднесені з об'єктом, на який вони спрямовані. Наведено приклад реалізації гнучкої захисної системи для упередження атак SQL-ін'єкції. Методи дають змогу краще ідентифікувати дії користувача, передбачити та упередити потенційну мережну небезпеку. Використання методів дозволить покращити інформаційну безпеку мережних ресурсів.

Ключові слова: інформаційна безпека, мережний ресурс, захист даних.

To build the system for information resource protection from network attacks and scans, it is necessary to analyze their implementation technology. As a rule, existing appliances for attacks detection are based on monitoring a set of indicators of the system state or on gathering some information about the resource or network features. Attacks prevention is usually based on ignoring potentially dangerous actions. This limits the scope of usage of existing methods detection and prevention of attacks, particularly for resources with restricted access to system parameters or server settings. The method for creating flexible protective solutions is proposed. The main difference of the flexible method is the simulation of information resource vulnerability with subsequent monitoring of user actions. The sorting of network attacks in correlation with the object to which they are directed was carried out. An example of a flexible protection system to prevent SQL-injection attacks was shown. The method allows for better identification of user actions, prediction and prevention of potential network threats. Use of the described techniques will improve the network resources information security.

Key words: information security, network resource, data protection.

Вступ

Активність розвитку мережних технологій, популярність глобальних мереж та мережних інформаційних ресурсів (МІР) супроводжуються постійним зростанням кількості мережних атак. Традиційно атаки поділяють на ті, що спрямовані на сервер та ті, що спрямовані на конкретний інформаційний ресурс. Загрози щодо безпеки серверів належать відповідальності системних адміністраторів, тоді як загрози МІР має врахувати програміст-розробник МІР. Виділимо предметом подальшого розгляду останні загрози та атаки, що їх реалізують.

Неодмінність атак стає типовим явищем у сучасних мережах. Здійснюється постійне автоматичне сканування мережі на предмет наявності вузлів, на яких не заблоковано доступ до файлової системи, які дають змогу з'єднатися з іншими МІР або містять вразливості, характерні для поширених інтегрованих засобів розроблення кодів, що активуються спеціальними послідовностями.

Протидія зазначеним загрозам засобами змін налагодження сервера переважно не дає бажаних результатів, оскільки сканери загроз використовують кожного разу інші адреси, зокрема адреси реальних легальних МІР, у яких було виявлено вразливість щодо транзиту запитів. Блокування доступу за виявленими адресами може з часом зробити МІР узагалі майже недоступним. Додатково варто зазначити, що не кожен розробник МІР може розраховувати на наявність дозволу щодо змін налагоджень сервера "під себе". Відтак проблема виявлення та упередження атак на МІР постає виключно перед його розробником (чи власником) і має бути врахована, бажано на етапі проектування.

Аналіз останніх досліджень і публікацій

Задача виявлення та упередження мережних атак через свою актуальність неодноразово розглядалась різними дослідниками і знайшла відображення у серії наукових праць. Так само очевидно, що ця задача належить до тих, що постійно зберігатимуть актуальність, оскільки засоби розвідки та здійснення атак постійно оновлюються та еволюціонують із розвитком захисних технологій.

Зазначені в останніх дослідженнях методи ідентифікації атак можна умовно поділити на два напрями. До першого належать досягнення, пов'язані з використанням засобів нечіткої логіки з формалізацією ознак мережних атак [6, 12], а також побудовою нейромережних засобів виявлення атак [7]. У цитованих дослідженнях описано методики виявлення атак моніторингом низки показників інформаційно-комунікаційних систем та виявлення їх критичних (аномальних) станів, що відповідають наслідкам реалізації загроз. Наведені методики хоча і мають універсальний характер, не передбачають розрізнення загроз за спрямуванням (на сервер чи ресурс), передбачаючи можливість збирання довільної інформації про стан системи – таких, як завантаженість процесора чи мережного каналу, розмір темпоральних файлів тощо. За умов, типових для функціонування більшості МІР, доступ до цих параметрів може бути обмежений чи взагалі заборонений, що ускладнює безпосереднє використання досягнутих результатів.

До другого напряму належать дослідження, побудовані на засобах класичного математичного аналізу. Серед останніх варто зазначити атомарну концепцію безпеки [8], яка передбачає протидію загрозам проектуванням захисту з урахуванням індивідуальних наборів атрибутів та перевпорядкуванням інформаційних зв'язків у ресурсі. Кінцевим етапом проектування є модель ресурсу як "молекули", що складається з "атомів", до яких групуються елементи ресурсу за атрибутами захисту. Подібні ідеї, проте з меншою розмірністю, визначаються стандартом [3], за яким ресурс рекомендується будувати у вигляді "шаруватої" структури. До кожного шару належать елементи, що є близькими за функціональним призначенням. Головною ідеєю відносно упередження атак є унеможливлення взаємодії елементів, що належать віддаленим (не сусіднім) шарам. За основними положеннями розглянуті концепції можуть бути запозичені для реалізації захисної системи МІР у контексті структурування та архітектурних принципів.

Також до методів, побудованих на класичному аналізі, належать лінійні моделі атак з варіацією вагових коефіцієнтів [9]. Зазначені методи дають змогу визначати основні види загроз інформаційному ресурсу у конкретних умовах оточення. Зазначені моделі можуть скласти підґрунтя для розподілу ресурсів захисту з виділенням найістотніших загроз.

Метою роботи є розвинення методів виявлення та попередження основних атак на мережні інформаційні ресурси з урахуванням обмеженості доступу до системних показників та налагоджень серверу, на якому ресурс функціонує.

Виявлення та упередження атак

Вважають, що мережну атаку здійснюють у три етапи: підготовчий етап (збирання інформації), виконавчий етап (дії), завершальний етап ("замітання слідів").

На підготовчому етапі здійснюється розвідка щодо того, яке програмне забезпечення встановлене на сервері і забезпечує функціонування МІР, якою мовою складено МІР, які додатки чи плагіни включено до складу МІР, який сервер баз даних використовується тощо – встановлюються

потенційні вразливості MIP. Оскільки думка щодо створення ідеально захищеного MIP має бути відкинута відразу, розробником MIP може бути прийняте одне з двох рішень щодо характеру реакцій захисної системи: жорстке та гнучке.

Під жорстким рішенням будемо розуміти створення комплексу захисних заходів з безпосереднього блокування можливих вразливостей. За такого підходу приховуються усі відомості щодо предмета розвідки. За фактом виявлення спроб підготовки атаки засоби захисту MIP просто ігнорують розвід-запит або обмежують (частково або повністю) подальшу активність користувача, з адреси якого надійшов нелегальний запит. Ідеальним за цим підходом результатом буде той, коли зловмисник не зможе розвідати жодних відомостей про MIP.

Навпаки, гнучке рішення передбачає формування у зловмисника відчуття успіху наданням йому хибної інформації щодо умов функціонування MIP та, можливо, сервера. У разі детектування спроб розвідки захисні елементи MIP сформують відповідь, з якої випливатиме хибний висновок щодо предмета розвідки. Зрозуміло, що гнучкість стосується не всієї захисної системи, а лише окремих її функцій. Немає потреби гнучко реагувати на всі можливі види розвідок.

Остаточне рішення, очевидно, приймає конкретний розробник чи замовник MIP, проте логічно надати перевагу саме гнучкому рішенню. Як аргументи можна невести такі.

Окремі способи розвідки, наприклад, щодо можливості SQL-ін'єкцій, полягають у малих змінах легальних запитів. Такими можуть бути додавання символу ′ (одинарна лапка) до значення змінної. Розміщення цього символу на клавіатурі цілком дозволяє його випадкове натиснення разом з клавішею "Enter". Якщо після такого (випадково спотвореного) запиту користувач буде заблокований, позитивне враження від MIP буде спаплюжене.

З іншого боку, якщо користувач насправді має зловмисні наміри, то негативний результат щодо простих розвідок спонукатиме його до складніших заходів. Оскільки ці заходи постійно оновлюються та еволюціонують, покладатись на абсолютну надійність захисту від них усіх не варто. Доцільно імітувати для зловмисника наявність простих "дірок" у захисті MIP, зосередивши увагу на його подальших діях. У разі переходу до активного етапу атаки за імітованою вразливістю висновок щодо злочинності намірів не викликатиме сумнівів.

Додатково до можливості трактування дій користувача, гнучкість захисної системи допоможе "виграти час", що, своєю чергою, допоможе визначити зловмисника. Одержавши повідомлення про спробу розвідки, адміністратор безпеки матиме час на аналіз походження запитів. Очікування їх повторення в активній фазі атаки також можна використати як відомості про зловмисника. У будь-якому разі краще попередити прості дії зловмисника, ніж пропустити складні.

З метою розвинення принципів побудови гнучкої захисної системи розглянемо типові способи здійснення атак на MIP. Єдиним способом активувати MIP та регулювати його функціональність (як легально, так і нелегально) є запит до нього. З погляду впливу на MIP запит можна умовно поділити на активуючу та керуючу частини. Активуюча частина запускає виконавчий код MIP і, як правило, складається з мережного імені ресурсу, номера комунікаційного порту та методу запиту. Керуюча частина слідує за активуючою і складається з директив, що встановлюють певні змінні і визначають надалі особливості виконання програмних інструкцій.

Відповідно, атаки на MIP також можна поділити за їх спрямуванням. Спрямовані на активуючу частину запиту атаки мають на меті самостійно запустити окремі програмні складові (скрипти) ресурсу або одержати вміст файлів чи каталогів. Наприклад, якщо нормальний запуск MIP передбачає запит у вигляді WWW.MIP.NET, то атака на скрипт (admin.php) може мати вигляд WWW.MIP.NET/admin.php, а спроба одержати файл (logo.gif) виглядатиме як WWW.MIP.NET/ images/logo.gif [напр. 13]. Окремі атаки можуть бути спрямовані на метод запиту, наприклад, замість запиту читання (GET) використовувати запит створення (PUT, POST) чи знищення (DELETE).

Класична протидія таким атакам полягає у створенні файлів та каталогів з неочікуваними іменами [16]. Гнучкість за умови дотримання класичних вимог може бути реалізована

перехопленням помилок “не знайдено” 404, а також “заборонено” 403 (за умови вгадування реального імені), з аналізом характеру та методу запиту у власному обробнику помилок.

Основна маса мережних атак на МІР спрямована на керуючу частину запиту. Різновидів цих атак настільки багато, що лише їх класифікація займає понад 40 сторінок [15]. Відповідно, намагатись виявити їх усіх у процесі оброблення запиту недоцільно. Аналогічно до попередньо запропонованого способу, можна рекомендувати відмовитись від автоматичного оброблення усіх запитів, створивши певну точку входу за формалізмом [1] – диспетчер доступу.

Засоби створення точки входу надаються локальними налагодженнями серверу. Наприклад, для серверу Apache це директиви, що знаходяться у файлі .htaccess, дія якого не вимагає одержання дозволу на керування сервером. Як приклад розглянемо спосіб створення єдиної точки входу МІР, обравши її ім’ям ускладнену для вгадування послідовність “a21e4.php”.

Стандартні обробники помилок змінюють на визначену точку входу командами на зразок “ErrorDocument 403 /a21e4.php”, “ErrorDocument 404 /a21e4.php”. Можна рекомендувати реалізацію перехоплення усіх можливих помилок сервера, принаймні з метою приховування версії сервера, який зазначається у стандартних формах-сторінках для помилок. Переведення на точку входу всіх запитів до МІР (шаблон .*) забезпечується командами:

RewriteEngine on

RewriteRule .* a21e4.php [L]

Організований описаним способом файл .htaccess забезпечить автоматичне скерування усіх звернень до МІР (зокрема помилкових) до обраної точки входу замість стандартної (index.php), яка може використовуватись як свідчення аномальної роботи, що відповідатиме збою точки доступу.

Головною перевагою створення точки доступу є можливість проаналізувати запит у тому вигляді, у якому він надійшов від користувача, без втручання автоматичних засобів виявлення помилок чи розділення змінних. Рядок запиту передається до МІР через серверну змінну REQUEST_URI. Саме це надає змогу виявити ознаки розвідок чи атак та їх попередити.

Зазначимо найтиповіші ознаки, за якими атаки можна ідентифікувати. Однією з найвірніших ознак атаки є спроба додавання до складу запиту активної команди. Оскільки такі команди необхідно відокремлювати одну від однієї, ознакою атак є наявність символів розділення чи переривання команд – “;”, пробіл, “%20”, “+”, “%00”, “|”. Виявлення таких символів є достатнім підґрунтям для повідомлення адміністратора безпеки щодо підозрілого запиту.

Викриті роздільні символи самі по собі не несуть загрози, хоча є їх ознакою. Тілом атаки є відокремлені активні елементи. Типові елементи, що використовуються для ін’єкції кодів, відповідно до об’єкта, на який вони спрямовані, наведено у табл. 1. Наведені дані не претендують на повноту переліку, а лише зазначають найпоширеніші способи здійснення мережних атак. Для практичного використання наведені дані можна суттєво розширити, створивши відповідну базу небезпечних елементів запиту.

Окрім зазначених у табл. 1, небезпеку можуть становити наявні у запиті мережні адреси інших МІР чи електронної пошти. Розмір запиту також може містити відомості щодо спроб атаки. Слід зауважити, що у межах цієї роботи акцент робиться не на повноті огляду засобів здійснення мережних атак, а на співвіднесені атак з об’єктом впливу, що врешті-решт є основою для створення гнучких захисних систем.

Саме встановлення об’єкта, на який спрямовано атаку чи розвідку, допомагає дезінформувати зловмисника [11]. Наприклад, у разі виявлення у запиті спроб виділити команду системи Unix [14], МІР, що насправді працює під управлінням іншої системи, може імітувати результати успішного виконання команди у визначеній невеликій області візуальної частини без спотворення загального інтерфейсу МІР. Оскільки зазначені команди переважно короткі (див. табл. 1), цілком можлива ситуація, за якою мнемоніка команди виявиться простою складовою безпечного запиту. У такому разі користувач не зверне уваги на малопомітні зміни інтерфейсу у відповіді сервера. Якщо ж від користувача надійде наступний запит з ознаками повторної ін’єкції Unix-кодів, то висновок щодо нелегальності його намірів набуде обґрунтованості.

Таблиця 1

Активні елементи, що використовуються для ін’єкції кодів

Об’єкт впливу	Характерні послідовності у запиті
Команди Unix	ls, cat, echo, mv, ln, touch, more, head, tail, ps, top, kill, !!, whoami, tee
Команди Windows	cmd, ftp, telnet, ping, dism, dialer, finger, getmac, ipconfig, locator, logonui, net,
Команди DOS	del, erase, copy, move, exit, dir, md, cd, rd, mkdir, diskcopy, format
Типові каталоги	/usr, /bin, /id, /chgrp, /chown, /etc /home /www /perl /servlet /con
Обхід каталогів	../, %2e%2e%2f, ..%u2216, ..%c0%af, ..%255c
SQL Server	xp_enumdsn, xp_availablemedia, xp_filelist, xp_cmdshell
PHP	<?php, ?>, $_REQUEST, $_GET, $_POST, phpinfo()
HTML та JS	Javascript\://, file\://, window.open, <script>, img src, <!--
Схожі для різних мов	fopen, include, print, getenv, exit
Команди сервера	traceroute, reboot, powerdown, server-info, server-status, mail
SQL запити	union, insert%20into, select, like, drop, or 1=1
Apache	HTTP_PHP, HTTP_USER_AGENT, HTTP_HOST, HTTP/1.
Потік виведення	>>, <<
Спеціальні файли	.system, .conf, .htpasswd, .history inc.php, config.php, cgi-, robot.txt
Імена для доступу	admin, root, user
Розширення файлів	.js, .jsp, .eml, .inc, .exe, .com, .msi, .html, .php

Для ілюстрації принципу побудови гнучкої захисної системи наведемо алгоритм створення модуля, призначеного для виявлення розвідок SQL-ін’єкцій та попередження подальших атак цього типу. Нехай функціями МІР передбачено оброблення цілочислової змінної x, що передається у складі HTTP запиту. Вибірка з бази даних здійснюється запитом на зразок SELECT * FROM tbl WHERE x=1 DESC LIMIT 1.

Очікуємо, що розвідка відбуватиметься через додавання до змінної x символу ‘ (x=1’). Проаналізуємо повідомлення про помилку оброблення такого запиту різними системами управління базами даних (СУБД), популярними в сучасних серверах. Результати наведемо у табл. 2.

Таблиця 2

Повідомлення про помилку оброблення помилкового запиту різними СУБД

СУБД	Повідомлення про помилку
MySQL	You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ‘/?x=1’ DESC LIMIT 1’ at line 1
Oracle	ERROR at line 1: ORA-01756 quoted string not properly terminated
MS SQL Server	Syntax error (missing operator) in query expression ‘/?x=1’ DESC LIMIT 1’
InterBase	Invalid token. Dynamic SQL Error. SQL error code = -104. Token unknown - line 1, char 28.
PostgreSQL	Query failed: ERROR: syntax error at or near ‘/?x=1’ DESC LIMIT 1’

Зберігаємо текст помилки тієї СУБД, що не є робочою для МІР. Оскільки, реалізуючи точку входу, одержуємо усі вихідні параметри запиту, здійснити аналіз щодо наявності у ньому SQL-ін’єкції не становитиме складності. За позитивного результату аналізу слід вивести повідомлення про некоректність даних у місці, функціонування якого визначається вибором змінної x та додати повідомлення про “помилку”, скажімо, у нижній частині сторінки (футері). Водночас необхідно сформувати повідомлення для адміністратора безпеки про небезпеку атаки та розпочати посилену фільтрацію наступних запитів цього користувача з виявленням ознак атак, типових для тієї СУБД, помилку від якої було імітовано [17].

Подібно до описаного способу можна забезпечувати гнучку реакцію захисної системи МІР на розвідки щодо інших об’єктів впливу [11]. З метою економії часу проектування та розгортання МІР може бути реалізовано змішаний підхід: частина захисних функцій на початку реалізується за простішим (і швидшим у реалізації) жорстким принципом, частина – відразу за гнучким. Якщо при

цьому дотримуватись модульного принципу програмування, рекомендованого у [1–2], надалі можна замінити початкові модулі на ті, що забезпечують гнучку захисну функцію.

Висновки

Аналізом типових атак на мережні інформаційні ресурси виявлено основні об'єкти, на які атаки спрямовані, а також співвіднесено ознаки атак із зазначеними об'єктами. Запропоновано розподіл захисних методів на жорсткі, що відкидають небезпечні запити, та гнучкі, які імітують вразливість ресурсу. Останнім надано перевагу в силу якіснішої ідентифікації дій користувача та можливості "виграти час", передбачити та попередити потенційну небезпеку. Наведено приклад організації гнучкого рішення для захисту від атак SQL-ін'єкції. Перспективи подальших досліджень вбачаються у розширені переліку ознак атак у співвіднесені їх з об'єктами впливу, розвиненні та реалізації гнучких захисних модулів для різних мережних загроз.

1. НД ТЗІ 1.1-002-99 Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. [Текст] / НД ТЗІ, ДСТСЗІ СБ України. 2. НД ТЗІ 2.5-010-2003 Вимоги до захисту інформації WEB - сторінки від несанкціонованого доступу. [Текст] / НД ТЗІ, СБ України 3. ISO/IEC 7498-1:1994(E) Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. [Text] / Second edition. 1996. Switzerland: ISO/IEC Copyright Office. – 68 p. 4. Самойленко Д. М. Проектування захищених ресурсів за об'єктно-орієнтованим принципом мовою PHP 5. [Текст] / Самойленко Д. М. // Збірник наукових праць НУК, 2014. – № 3. – с. 83-87. 5. Шевченко Р. Рейтинг языков программирования №5, январь 2014 [Електронний ресурс] / Руслан Шевченко. – Режим доступу http://dou.ua/lenta/articles/language-rating-jan-2014/ (дата звернення: 26.12.14). 6. Азарсков В. Параметры прогнозирования и идентификации атак в информационно-коммуникационных системах. [Текст] / В. Азарсков, А. Гизун, А. Грехов, С. Скворцов // Захист інформації, 2014, т. 16, № 1. – с. 89-95 7. Корченко О. Сучасні нейромережеві методи та моделі оцінки параметрів безпеки ресурсів інформаційних систем. [Текст] / О. Корченко, І. Терейковський, А. Дзюбаненко // Захист інформації, 2014, т. 16, № 3. – с. 223-232 8. Іванченко І.С. Захист інформаційних ресурсів на основі атомарної концепції безпеки. [Текст] / І.С.Іванченко // Інформаційна безпека, 2013, № 3 (11). – с. 22-28 9. Карпінський М. П. Атаки на відмову в обслуговуванні комп'ютерних мереж [Текст] / Карпінський М. П., Яциковська У. О., Балик А. В., Александер М. // Вісник Національного університету "Львівська політехніка": "Комп'ютерні системи та мережі". – 2014. – № 806. – с.94-99 10. Руководство по PHP [Електронний ресурс] / Группа документирования PHP. – Режим доступу http://php.net/manual/ru/ (дата звернення: 26.12.14). 11. Самойленко Д. М. Модель захищеного інформаційного ресурсу з обманними функціями. [Текст] / Самойленко Д. М. // Інформаційна безпека. – 2013. – № 4 (12). – с. 107-111. 12. Корченко А. А. Метод α-рівневої номіналізації нечітких чисел для систем виявлення вторгнень [Текст] / Корченко Анна Александровна // Захист інформації, 2014, т. 16, № 4. – с. 292-304. 13. Разумов М. Следы атак по 80 порту - исследование сигнатур атак [Електронний документ] / Михаил Разумов, Домашняя сеть Veer /режим доступу http://proit.com.ua/article/internet/2007/01/12/170054.html Назва з екрану 14. Barschel C. Unix Toolbox [Електронний документ] / Colin Barschel / режим доступу http://cb.vu/unixtoolbox.xhtml 15. WEB SECURITY Announcement: WASC Threat Classification in Russian [Електронний документ] /Jeremiah Grossman / режим доступу http://www.webappsec.org/projects/threat/v1/WASC-TC-v1_0.rus.doc 16. Низатмудинов М. Ф. Тактика защиты и нападения на WEB-приложения [Текст] / Марсель Низатмудинов. – СПб: БХВ-Петербург, 2005. – 432 с. 17. Учимся на ошибках: методика проведения Error-based SQL-Injection [Електронний документ] /Журнал "Хакер". – 2010 (27.05) / режим доступу https://xakep.ru/2010/05/27/52222/