Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кібербезпка
(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломної роботи

магістр
(освітній рівень)

на тему: **Аналіз проблеми захисту від DDoS-атак на основі ботнетів**

Виконав: студент 6 курсу, групи СНм-61
спеціальності 125 «Кібербезпека»
(шифр і назва спеціальності)

		Амбок В.І.
	(підпис)	(прізвище та ініціали)
Керівник		Александер М.Б.
	(підпис)	(прізвище та ініціали)
Нормоконтроль		Кареліна О.В.
	(підпис)	(прізвище та ініціали)
Рецензент		
	(підпис)	(прізвище та ініціали)

м. Тернопіль – 2019

РЕФЕРАТ

"Аналіз проблеми захисту від DDoS-атак на основі ботнетів" Амбок Володимир Іванович // Тернопільський національний технічний університет ім. І.Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2019 // с. – , рис. – , табл. – , ілюстр. – , джерел – .

Ключові слова: БОТНЕТ, ЗАГРОЗА, ІДЕНТИФІКАЦІЯ БОТНЕТІВ, , ПРОТОКОЛ.

Ботнет — це мережа комп'ютерів, інфікована шкідливим програмним забезпеченням. Кіберзлочинці використовують ботнет-мережі, які складаються з великої кількості комп'ютерів для різних зловмисних дій без відома користувачів. За допомогою ботнетів часто надсилається спам, встановлюються шпигунські програми або здійснюється викрадення облікових даних користувачів. Масштабний ботнет може використовуватися для атак типу DDoS (Distributed Denial of Service) для спрямування додаткового трафіку на сайт та сповільнення роботи або збоїв підключення.

Шкідливі програми виду ботнет розповсюджуються за допомогою вкладень електронної пошти та через завантаження файлів і підроблених програм. Зловмисники також націлюються на такі уразливі місця, як неоновлене програмне забезпечення та відсутність захисту в мережі Інтернет. Все частіше під приціл зловмисників потрапляють камери, смарт-телевізори та навіть автомобілі.

Виявлення, а, відповідно, і захист від ботнетів є важливою і актуальною задачею. Обчислювальна потужність одного ботнету дозволяє здійснювати декілька зловмисних дій швидко та часто без виявлення.

ANNOTATION

" Analysis of security problem caused by botnet-based DDoS-attacks" // Diploma paper of Master degree level // Ambok Volodymyr Ivanovych// Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Cybersecurity Department // Ternopil, 2019 // p. – , Fig. – , Tables – , Refence. – .

Key words: BOTNET, THREAT, BOTNET IDENTIFICATION,, PROTOCOL.

Botnet is a network of computers infected with malware. Cybercriminals use botnets that consist of a large number of computers for various malicious activities without the knowledge of users. Botnets often send spam, install spyware, or steal user credentials. A large-scale botnet can be used for DDoS (Distributed Denial of Service) attacks to direct additional traffic to the site and slow down work or connection failures.

Botnet-type malware is spread through email attachments and through file downloads and fake applications. Attackers also target vulnerabilities such as uninstalled software and a lack of Internet security. Increasingly, the target of intruders is cameras, smart TVs and even cars.

Detecting and, accordingly, protecting against botnets is an important and urgent task. The computing power of a single botnet allows several malicious actions to be performed quickly and often without detection.

ЗМІСТ

РОЗДІЛ 1
ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1.1 Шаблони комунікацій Botnet

Зловмисні ботнети стають загальною загрозою і сьогодні переймають великі частини Інтернету. Існуючі опитування та систематики зосереджуються на топологіях ботнету, протоколах управління та управління (C&C) та цілях ботнету. Спираючись на ці результати досліджень, були запропоновані мережеві методи виявлення, здатні виявляти відомі ботнети. Методи встановлення та експлуатації ботнетів значно розвинулися за останнє десятиліття, що спричинило потребу в методах виявлення, здатних виявляти нові, раніше невідомі типи ботнетів.

У цій роботі ми представляємо глибокий аналіз усіх аспектів мережевої комунікації при створенні та роботі ботнету. Ми розглядаємо топологію, протоколи протоколів ботнетів та аналізуємо великий набір дуже різних і дуже складних існуючих ботнетів з точки зору мережевого зв'язку. На основі нашого аналізу ми впроваджуємо нову систематику узагальнених моделей зв'язку для ботнет-комунікації з використанням стандартизованих діаграм послідовності уніфікованої мови моделювання (UML). Крім того, ми вивчаємо варіанти обміну даними та досліджуємо вплив методів шифрування та приховування. Наші узагальнені схеми комунікації дають корисну основу для розробки складних механізмів детектування ботнетних мереж і можуть запропонувати ключовий компонент для побудови незалежних мережевих детекторів протоколу та топології.

Комп'ютерні мережі дозволяють розподіляти обчислення та обмін ресурсами. Поширення завдань на декілька машин дозволяє виконувати завдання, вимоги яких перевищують ресурси, доступні на одному комп'ютері. Ця методика також може прискорити обробку одного завдання, розділивши його на підзавдання, які можна виконати на декількох комп'ютерах одночасно.

Один з варіантів проектування таких систем - це поєднання ресурсів на мережевих комп'ютерах, які називаються ботами, призначеними для виконання завдання або підзадач.

Для координації ботів потрібен додатковий мережевий комп'ютер, який називається майстер. Звідси і дана назва ботнету, що є комбінацією бота і мережі.

Оскільки Інтернет містить величезну кількість невикористаної потужності обробки та пропускну спроможність мережі, зловмисники виявили способи використовувати ці машини для досягнення своїх цілей. Це досягається шляхом зараження машин шкідливим програмним забезпеченням (зловмисним програмним забезпеченням), а отже, побудовою шкідливого ботнету, який використовує компрометовані машини для обчислення.

Зловмисні ботнети використовуються для різних завдань, включаючи крадіжку інформації [1], [2] або зловживання онлайн-послугами [3]. Інший великий відсоток ботнетів використовується для порушення послуг, який використовується для задушення інших гравців (наприклад, конкуренція, іноземні держави) [4], [5]. Ці зловмисні ботнети представляють реальну загрозу для людей в Інтернеті [2], [6], мережевих інфраструктурах [7] і навіть Інтернеті загалом [4], [8]. Тому були опубліковані різні систематики та рекомендації, що класифікують різні аспекти ботнетів та способи їх виявлення. Відповідно до більшості попередніх публікацій, в решті цієї роботи використовується термін ботнетів для позначення шкідливих ботнетів.

Ботнет складається з і) декількох ботів, ii) сервера Command and Control (C&C) та iii) botmaster. Крім того, у цій роботі використовується термін жертва для цілі нападу або зараження ботом. Діаграма на рис. 1 ілюструє учасників ботнету, ролі та зв'язок у гіпотетичному ботнеті з централізованою структурою команд.

Боти (їх ще називають зомб) - це заражені машини, які виконують виконуваний бот. Ці машини обробляють завдання, надані сервером C&C.

Сервер C&C є центральним об'єднанням ботнету. Оскільки цей сервер здатний керувати кожним ботом, він повинен бути захищений від захоплення сторонніми сторонами, наприклад, правоохоронними органами, дослідниками

або конкуруючими майстрами ботів. Сучасні топології ботнетів (наприклад, Peer to Peer (P2P)) можуть використовувати звичайні боти, що входять до ботнету, як сервер C&C. Таким чином, усувається загроза збиття ботнету через єдину точку відмови.

Botmaster є людиною, керуючим ботнетом через сервер C & C. Ботмейстери намагаються залишатися максимально прихованими, оскільки розкриття особистості ботмайстра може призвести до потенційного судового переслідування. Цього можна досягти, наприклад, використовуючи якомога менше трафіку або не використовуючи пряме з'єднання з рештою ботнету.

1.2 Детектори ботнетів

Ботнет-мережі, які беруть участь у можливо незаконній діяльності, можуть спричинити небажаний мережевий трафік або заважати повсякденному бізнесу. Таким чином, виявлення та видалення ботнетів є важливими завданнями. Оскільки ботнетні мережі, як правило, приховують свою роботу, виявлення ботнетів є активним полем досліджень. Виявлення Botnet може орієнтуватися на одну з трьох частин ботнету (бот, C&C сервер або ботмайстер, як було пояснено вище та на рис. 1.1). Хоча ця робота зосереджена на виявленні ботів, топології, описані в розділі III та протоколи, описані в розділі IV, також стосуються сервера C&C. Крім того, сервер C&C є частиною представлених моделей комунікацій у розділі VI. Шаблон зв'язку представляє собою послідовність обмінюваних повідомлень, необхідних для сценарію конкретного повідомлення. Таким чином, ця робота також може бути використана як основа для побудови серверного детектора C&C.

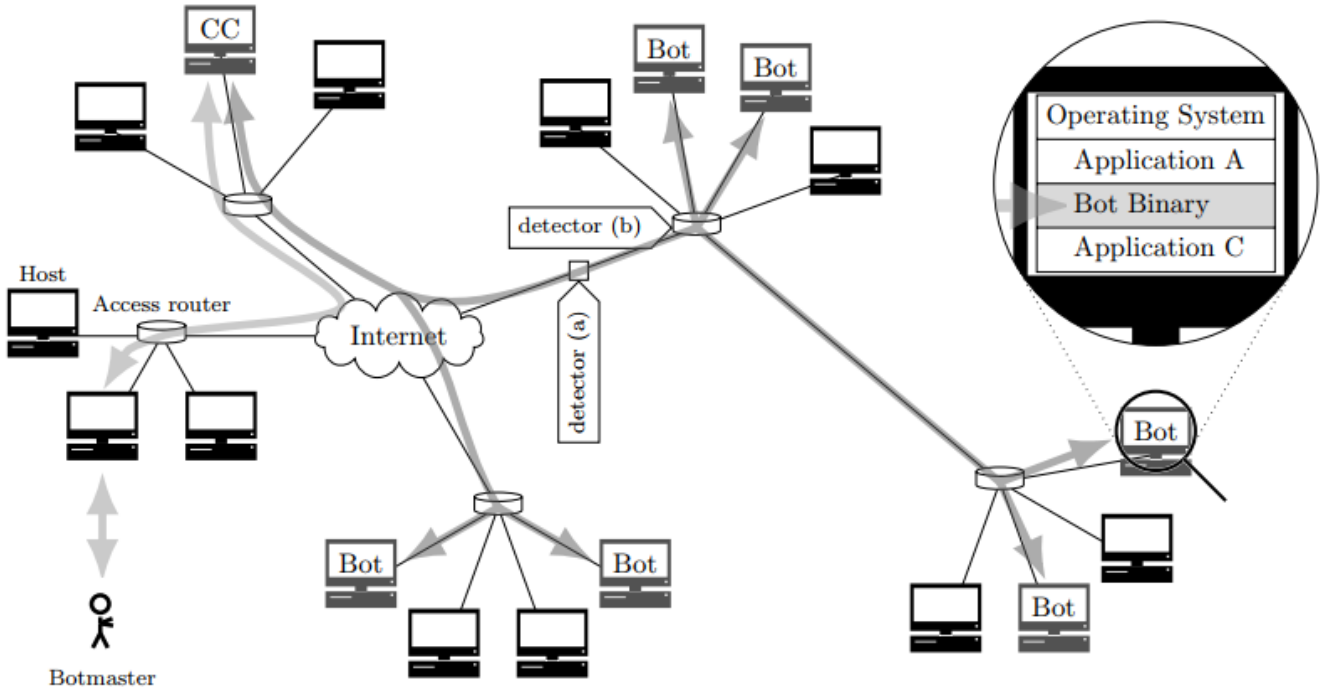

Рисунок 1.1 – Приклад огляду Botnet, що складається з централізованого ботнету, що включає ботмейстер, сервер C&C, канали зв'язку, боти та двійкові файли ботів. Додатково зображені приклади мережевих детекторів (a) та (b). Детектор (a) виконує функцію проміжного вузла, тоді як (b) включений в маршрутизатор.

Оскільки мережевий зв’язок є першорядним для ботнету, він повинен бути присутнім і тому може бути використаний для виявлення бота. Це називається виявленням на основі мережевих ботів. Приклад позицій для мережевих детекторів ботнету можна побачити на рис. 1.1 (Детектори (а) та (б)).

Мері комплексний підхід для реалізації networkbased детектора є сигнатурний виявлення або виявлення синтаксичної [9]. Ця методика працює, спочатку збираючи зв’язок із відомих ботнетів. Далі, з цих зразків можуть бути вилучені репрезентативні байтові послідовності або текстові уривки, які використовуються цільовими ботнетами під час C&C зв'язку. На останньому кроці ці результати порівнюються з невідомим трафіком. Якщо є відповідність, ботів виявлено.

Оскільки вищезазначені послідовності байтів або текстові уривки повинні точно збігатися, прямим контрзаходом для приховування від таких методів виявлення є зміна використаних команд C&C. Узагальнення цієї ідеї полягає в

додаванні випадкових послідовностей на початку або у довільних положеннях команди C&C, що називається padding. Додавання даних на початку може використовуватися для використання обмежень довжини в детекторах, тоді як додавання шаблонів в інших місцях може призвести до поразки самого виявлення підпису. Хоча варіації прокладки та протоколів можна пом'якшити, розділивши захоплену комунікацію на підшаблони, шифрування протоколу може зірвати кожне виявлення на основі підписів.

Для подолання цього обмеження може бути використаний аналіз поведінкової мережі. Ця методика спирається на класифікацію трафіку мережі як звичайний або аномальний трафік [9]. Тому цю методику ще називають виявленням мережевої аномалії.

Один із способів побудови мережевого детектора аномалій - це спостереження за “ненормальною” поведінкою (наприклад, трафіком бот-мережі C&C) та знаходженням характерних зразків або властивостей (наприклад, хронометраж пакету, кількість пакетів, довжина пакету). Ці властивості називаються особливостями. Те ж можна досягти, спостерігаючи за «нормальною» поведінкою (тобто законним трафіком, захопленим під час роботи мережі), вилученням функцій та визначенням мережевого трафіку, який не відповідає цим шаблонам, як аномалія.

Інспекція корисного навантаження не потрібна, і тому такий підхід працює з зашифрованим або затуманеним мережевим трафіком. Оскільки ця методика намагається знайти узагальнені властивості небажаного трафіку, її теоретично можна використовувати для виявлення раніше невідомих ботнетів.

Простий спосіб реалізації цієї процедури - це використання машинного навчання. Машинне навчання - це процес, який змінює параметри алгоритму, поки вихід алгоритму не збігається з очікуваним результатом для заданих входів. Це називається тренуванням. Навчений алгоритм машинного навчання може бути наближенням залежності між тренованими параметрами та очікуваними результатами. Оскільки не обов'язково існує кореляція в основі даних, результати повинні бути перевірені та використані обережно.

Запропоновано різні детектори, які навчають різні методи машинного навчання із спостережуваним трафіком ботнету. Вищезазначений підхід використовувався для вилучення функцій з мережевого трафіку, навчання алгоритму машинного навчання з відомими даними та використання тренованого алгоритму для виявлення ботнетів. Алгоритми були навчені з нормальним та ботнет-трафіком. Для оцінки цю процедуру використовували п’ять різних технік машинного навчання. Результати порівнюють різні алгоритми відповідно до швидкості тренувань та класифікації та ефективності виявлення. За допомогою цього простого підходу можна виявити ботнети з аналізом лише метаданих пакетів.

Однак ці детектори не здатні ні пристосовуватися до змін мережевого трафіку, ні виявляти нові ботнети.

Було розширено цей підхід, додавши крок фільтрування перед машинним навчанням та додавши автоматизований вибір функцій. Це працює, додаючи алгоритм машинного навчання, який використовується для відкидання функцій, які мають лише невеликий вплив на результат. Досягнуто кращих результатів у виявленні відомих ботнетів, ніж у попередніх роботах, але продуктивність на невідомих ботнетах не враховувалася.

Ще один ботнет-детектор також був заснований на згаданому вище простому підході, але замість виявлення ботів він був розроблений для виявлення серверів C&C. Щоб зменшити кількість неправильних класифікацій доброякісних вузлів як серверів C&C, результат поєднувався з результатами різних чорних списків. Чорний список - це список, що містить елементи, яким заборонений доступ. Що стосується ботнетів, ці списки містять адреси або імена відомих серверів C&C.

Згадані вище детектори тренують алгоритми машинного навчання з нормальним трафіком для нормального класу та зі спостережуваним трафіком ботнету для аномального класу для досягнення високої точності. Це означає, що можна виявити лише ботнетів, які зустрічалися під час навчання та близьких родичів. Тому вищезазначеної мети виявлення невідомих ботнетів неможливо досягти.

Є детектори, що намагаються подолати це обмеження, розглядаючи типові завдання ботнету під час аналізу трафіку (наприклад, атаки розподіленого відмови в обслуговуванні (DDoS), розповсюдження небажаних електронних повідомлень (електронних листів), сканування в мережі). Більш детальне пояснення цих робіт можна побачити у наступних розділах роботи. Оскільки ця додаткова інформація використовується для підготовки до конкретних завдань або протоколів, можливість виявлення невідомих ботнетів все ще обмежена.

Ботнети - це дуже досліджена область. Опубліковано багато робіт, які аналізують або класифікують ботнети та їх поведінку. Однак, незалежні від протоколу та топології схеми мережевого зв'язку, необхідні ботнетам, не охоплюються цими. Незважаючи на відсутність досліджень у цій галузі, були запропоновані деякі детектори, які намагаються врахувати поведінку мережі.

На даний момент таксономія ботнетів вводить категорії для кожного аспекту ботнетів і представляє вичерпний список членів для кожної категорії, який підтримується вибраними прикладами ботнетів. Хоча представлений детальний перелік завдань та топологій ботнету та ретельний перелік протоколів C&C, співвідношення між поведінкою бота та мережевою комунікацією не наводиться.

Є детектори ботнетів, які призначений для виявлення ботнетів без необхідності навчання чи підпису. Він заснований на базовому діалоговому вікні зараження, який складається з вхідного сканування, вхідної інфекції, завантаження яєць (виконуваного двійкового файлу) та виїзного сканування для зв'язку C&C. Цей детектор використовує та розширює існуючу систему виявлення вторгнень (IDS) для відповідності мережевих подій правилам, згенерованим із діалогового вікна зараження. Такий підхід обмежений потребою в незашифрованому мережевому спілкуванні, дуже основних моделях комунікацій, потребі в декількох зараженнях ботів у локальній мережі та вимозі активного поширення ботнет-мережі.

Спільним для всіх вищезазначених робіт є те, що поведінка зв'язку ботнетів спостерігається лише з точки зору топології та протоколу. Ця робота надає більш глибокий погляд на зв’язок C&C з ботнетом, надаючи наступні внески:

1) Поглиблений погляд на існуючі топології ботнету з точки зору зв'язку. Різні топології та їх варіації детально пояснюються, включаючи переваги, недоліки та проблеми. Крім того, посилання на існуючі ботнети включаються для кожної зміни.

2) Детальний опис C&C та експлуатаційних протоколів. Описи протоколів включають можливі топології, переваги, недоліки, посилання на існуючі ботнети та історичний контекст.

3) Категоризація існуючих старих і нових ботнетів з точки зору мережевого зв'язку. Ботнетні мережі були спеціально вибрані, щоб забезпечити велику різноманітність різних цілей ботнету, топологій, протоколів та складності.

4) Нова систематика узагальнених моделей комунікації ботнетів. Ці зразки - це послідовності повідомлень, які необхідно обміняти для виконання запланованих завдань ботнету. Ця систематика складається із сценаріїв, що охоплюють можливі завдання ботнету.

Кожен сценарій пояснюється відповідно до відповідних завдань, з детальним описом необхідних моделей комунікацій та прикладів відповідних ботнетів. Описи шаблонів не залежать від використовуваних топологій або протоколів.

5) Структура для побудови мережевих детекторів ботнетів. Таксономія моделей зв'язку ботнету містить діаграми послідовностей UML, що візуалізують послідовності повідомлень стандартизованим способом, забезпечуючи сумісність та майбутню розширюваність. Ці діаграми послідовності не залежать від використовуваного протоколу зв'язку, шифрування та топології, що дозволяє будувати майбутні детектори ботнет-мереж.

1.3 Топологія ботнетів

Як було сказано у вступі, ботнет складається з декількох ботів, які заражені хостами, які використовуються для виконання вказівок, що даються ботмайстром (див. рис. 1.1). Для виконання цих завдань потрібен сервер C&C та боти. Залежно від конкретного ботнету, функціональність бота та сервер C&C

іноді можуть перебувати на одному фізичному хості. Аналогічно, функціональність сервера C&C може бути розподілена на декілька фізичних хостів.

З точки зору топології комунікація між сервером C&C та ботами може бути організована як централізована, децентралізована та комбінація децентралізованих та централізованих, званих гібридними топологіями, за категоріями [9]. У решті цієї роботи ці топології будемо позначати відповідно як централізовані, P2P та гібридні.

Огляд цих топологій можна побачити на рис. 1.2.

A. Централізований

Найпростіша схема компонування ботнетів - це централізована топологія. Ця топологія використовує центральний виділений сервер C&C, кожен бот підключається безпосередньо до цього сервера (див. Рис. 2a). Відповідно до [24] цю топологію легко встановити, має низьку затримку та високу масштабованість. Його легко встановити, оскільки особливих вимог щодо протоколу немає. Тому можна використовувати прості протоколи, такі як Інтернет-ретрансляційний чат (IRC) (див. Розділ IV-A) або HTTP (див. Розділ IV-B). З низькою затримкою і високою масштабованості є викликане простою структурою мережі, в якій команди передаються безпосередньо з сервера C & C до кожного боту.

Переваги централізованої топології полягають у низькій надійності, що спричиняється тим, що сервер C&C є єдиною точкою відмови. Тому для того, щоб зняти всю ботнет, потрібно зняти лише сервер C&C. Це можна зменшити до певного рівня, використовуючи реплікувані сервери C&C замість одного сервера C&C.

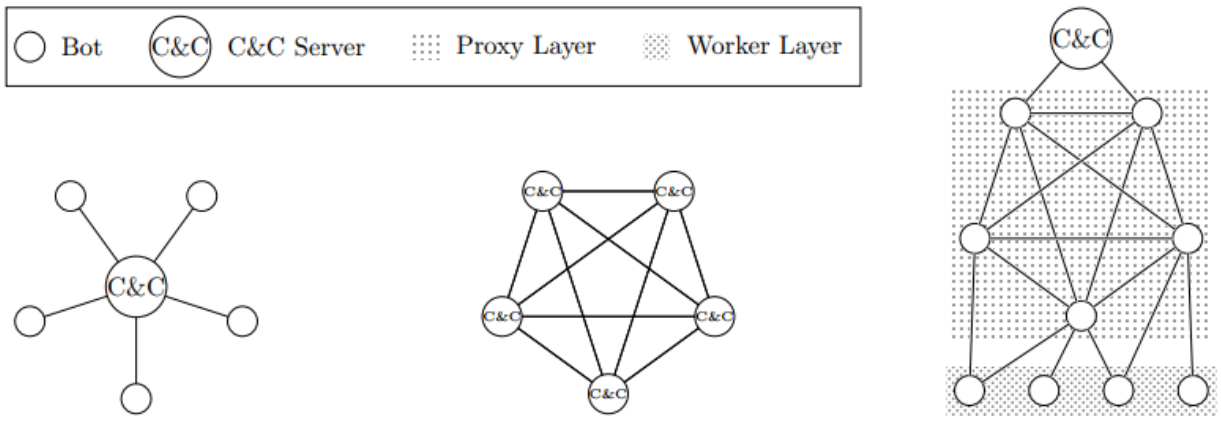

Рисунок 1.2 – Топології ботнетів

(a) Централізований ботнет: Один або кілька C&C

(b) Повністю зв'язаний ботнет P2P: Не виділений

(c) Гібридний ботнет P2P / C і C: Об'єднує P2P і сервер (и), кілька ботів.

Залишилася проблема полягає в тому, що адреси серверів C&C потрібно чітко кодувати в ботнет. Шифрування або оскарження програмного коду має лише тимчасове значення, оскільки дешифрування або знеструмлення займає обмежений час. Крім того, сервер C&C можна виявити, спостерігаючи мережевий трафік бота. Це можна зменшити за допомогою алгоритму генерації домену (DGA).

DGA генерують різні доменні імена, похідні від змінних даних [25]. Більшість DGA використовують поточний час як вхід алгоритму, тому кожен бот в ботнеті повинен мати синхронізований системний час. Залежно від використовуваного алгоритму достатньо бути точним до найближчої години чи дня. Якби це не так, кожен бот генерував би інше доменне ім’я, і, таким чином, надійне спілкування з сервером C&C було б неможливим. Ботні мережі вирішують цю проблему, налаштовуючи комп'ютери, на яких виконується бот, щоб синхронізувати час, або запитуючи популярні веб-сайти.

Наступним недоліком цієї системи є те, що після з'ясування алгоритму кожен може взяти на себе ботнет, резервуючи доменне ім’я, яке DGA буде генерувати в певний час у майбутньому. Проблема реєстрації домену може бути зменшена до певної міри, використовуючи недетермінований вхід у DGA. У минулому спостерігалися ботнети, які використовували актуальні теми щодо

щебетання або обмінного курсу. Замість того, щоб заздалегідь реєструвати доменні імена, той, хто найшвидший в реєстрації нового доменного імені, може контролювати ботнет.

Окрім приховування сервера C&C, ця методика також може використовуватися для посилань у створених спам-повідомленнях. За допомогою цієї методики виявлення спаму на базі Уніфікованого локатора ресурсів (URL) може бути переможене. URL-адреса (веб-адреса) - це посилання на веб-ресурс.

Виявлення спаму на основі URL працює за допомогою створення списку URL-адрес із відомих спам-повідомлень. URL-адреси в електронних листах можна порівняти зі списком, і якщо буде знайдено позитивну відповідність, електронний лист позначається як спам. Якщо DGA використовується для генерування цих URL-адрес, створення списку не може йти в ногу з новоствореними URL-адресами.

Інший спосіб пом'якшити низьку надійність централізованої топології - створити мережу швидкого потоку. У мережі з швидким потоком система доменних імен (DNS) використовується як мультиплексор і балансир завантаження. DNS - це ієрархічна система та супровідний протокол для асоціації інформації з доменними іменами. Ця система зазвичай використовується для асоціації доменних імен з адресами Інтернет-протоколу (IP-адреси) (запис хостів). Тому DNS надає спосіб використання слів, які запам'ятовуються легше замість чисел. Додаткове використання включає асоціацію декількох доменних імен з однаковою IP-адресою, що забезпечує можливість використання одного і того ж хоста для декількох різних веб-сторінок.

Також можливо призначити кілька IP-адрес одному і тому ж доменному імені, що дозволяє використовувати декілька хостів для однієї веб-сторінки. Цей механізм використовується в мережі швидкого потоку, де для одного запису хоста DNS реєструється кілька ботів. Після вирішення запису один або кілька цих ботів повертаються. Замість підключення безпосередньо до сервера C&C ці боти використовуються як проміжні хости, які, в свою чергу, передають дані на

сервер C&C. Тому лише ці так звані проксі-боти знають справжній сервер C&C. Оскільки реєстрація та дереєстрація серверів може бути здійснена в будь-який час, цей метод може бути використаний для швидкого переходу через декілька серверів. Це збільшує зусилля, необхідні для відключення ботнету. Метод fastflux переносить єдину точку відмови на DNS-сервер. Це також можна усунути, додатково використовуючи ботнет як авторитетний сервер імен з тією ж схемою циклічного циклу, яку називають подвійним потоком.

Пізніші варіанти ботнету Zeus використовують централізовану архітектуру, використовуючи DGA. Інший приклад - Conficker, який генерує 50000 доменів у добу у версії C. Ця велика кількість доменів допомагає розподілити навантаження на сервери C&C і робить попередню реєстрацію доменів завданням, що складеться в технічному плані. Прикладами мереж з подвійним потоком є Storm та Waledac. Більш детальне пояснення цих ботнетів можна побачити у розділі V.

B. P2P

Для подальшого підвищення стійкості до збою або збою в мережі можна використовувати топологію P2P. Ця топологія складається лише з ботів, де кожен бот може бути потенційним сервером C&C. Кожен бот з'єднаний щонайменше з одним іншим ботом і команди можуть охопити весь ботнет, лише якщо кожен бот має можливість передавати команди безпосередньо підключеним ботам (див. Рис. 2б).

У повному зв’язаному ботнеті кожен бот підключений до кожного іншого бота. Це забезпечує низьку затримку зв'язку, оскільки повідомлення не потрібно передавати через додаткові боти, щоб транслювати повідомлення кожному боту. Крім того, повністю зв'язані ботнети мають високу надійність, оскільки видалення довільної кількості ботів з ботнету не порушує спілкування. Оскільки для великих ботнетів потрібна дуже велика кількість мережевих підключень, такий підхід не є масштабним. Залежно від типу протоколу, повномережний ботнет може мати максимальну кількість ботів за рахунок операційних систем, що обмежують максимальну кількість розеток протоколу управління

транспортом (TCP). TCP - транспортний протокол, здатний надійно передавати потоки даних між двома хостами.

Крім того, кількість необхідних з'єднань збільшує видимість ботнету. Крім того, додавання та видалення ботів через зміни в мережі вимагає великої кількості повідомлень про координацію. Тому більшість ботнетів P2P не є повністю пов'язаними.

Як зазначається, топології P2P важко реалізувати. Це викликано проблемами пошуку початкових одноліток та надійного розподілу команд кожному боту.

Пошук одноліток можна вирішити, вставивши у виконаний бот жорсткий зашифрований список відомих одноліток. Інший спосіб зберігання цього початкового списку - це використання кеш-серверів існуючих мереж додатків P2P. Оскільки виконуваний бот повинен бути розповсюджений в Інтернеті, щоб сформувати ботнет та кеш-сервери загальнодоступними, дослідники можуть витягнути цей початковий список одноліток. Отже, єдина точка відмови, як було обговорено вище для централізованої топології, переміщується до цього первинного списку рівних. Для того, щоб позбутися загальнодоступних списків одноліток, іншим рішенням для пошуку первинних одноліток є випадкове сканування в Інтернеті ровесників.

Протокол повинен дбати про надійне розподіл команд через ботнет з топологією P2P. Як вже згадувалося раніше, ботнети не використовують тоталогію, що повністю переплетена. Ось чому боти повинні підтримувати ретрансляцію повідомлень, щоб транслювати повідомлення кожному боту. Існуючі протоколи P2P вже надають всю необхідну функціональність. Якщо використовується неотеричний протокол або існуючий не P2P- протокол, то слід додати надійність та функціональність маршрутизації. Це детально буде обговорено у розділі IV.

Не кожен комп'ютер доступний через Інтернет. Це може бути викликано брандмауером або трансляцією мережевих адрес (NAT), які приховують декілька комп'ютерів за одним маршрутизатором. Тому деякі ботнети класифікують ботів на категорії суперноди та NAT-вузли. Після зараження виявляється зв'язок і

пізніше для зловмисної активності використовуються лише NAT-вузли, тоді як надвузли використовуються лише як реле зв'язку C&C ботнету. Цей спосіб розгортання топології P2P також досліджувався в.

Прикладами ботнетів, що використовують топологію P2P з жорстко кодованими списками однолітків, є ботнет Zeus, який повертається до централізованого, якщо жоден одноранговий доступ недоступний, і Sality (версії 3 та 4), який також має резерв до централізованої топології [37]. Phatbot вирішив проблему списку однолітків, використовуючи кешові сервери платформи обміну файлами для початкового списку однолітків. Прикладами випадкових скануючих ботнетів є Sinit та Conficker. Zeroaccess розділив мережу на супервузли та NATnodes. Більш детальне пояснення цих ботнетів див. У розділі V.

C. Гібрид

Як видно вище, топології P2P демонструють слабкі місця (наприклад, можливість повного зриву шляхом видалення частин мережі, ненадійної передачі команд, пошуку однолітків, повного перейому через одного однорангового), які необхідно подолати за допомогою ретельного проектування. Інший спосіб усунення цих слабких місць - поєднання P2P та централізованих топологій для отримання переваг обох світів. Замість ботів, що підключаються безпосередньо до серверів C&C, додається додатковий проксі-шар, що складається з ботів, підключених до топології P2P. Для зниження видимості цього проксі-шару додається третій шар, що складається з ботів, які виконують завдання ботнету. Принциповий огляд цієї реалізації можна побачити на рис. 2в.

Визначити, чи бот стає робочим ботом чи проксі-ботом, можна зробити, як було сказано раніше для топології P2P, виходячи з властивостей інтернету бота. Ціною затримки повідомлення можуть бути додані додаткові шари для подальшого підвищення захисту сервера C&C від виявлення.

Іншим способом реалізації гібридного ботнету є використання централізованої топології для однієї частини зв'язку C&C та P2P для іншої частини. Наприклад, P2P може використовуватися для обходу DNS для інакше централізованої топології.

Спадщина може бути ще однією причиною використання частини топології P2P та частини централізованої топології. Один із способів продовжити використання існуючих веб-серверів - це використовувати P2P лише для обміну командами та централізованої топології для передачі даних.

Прикладами ботнетів, що використовують гібридну топологію, є Майнер ботнет та пізніші версії ботнету Zeus. Шторм замінив DNS, використовуючи існуючу мережу P2P для пошуку серверів C&C. Версія 3 ботнету Sality використовує гібридну мережу, де частина P2P відповідає за обмін командами, тоді як централізовані веб-сервери використовуються для завантаження даних. Оскільки цей централізований компонент представляв єдину точку відмови, версія 4 перейшла до мережі лише P2P. Більш детальне пояснення ботнетів можна побачити у розділі V.

1.4 Протоколи роботи ботнетів

Центральним для ботнету є його зв'язок, який необхідний для віддаленого управління ботами. Як було обговорено в розділі III, макет топології ботнету можна відобразити на одній із трьох альтернатив: централізованій, P2P та гібридній. Різні топології створюють різні обмеження на протоколи зв'язку, які можна використовувати. Крім того, протоколи можуть бути адаптовані до особливих потреб, такі як прихованому я Ness, які можуть бути досягнуті шляхом створення протоколу, який імітує законним трафік або шляхом повторного використання існуючого протоколу. Ще одним важливим фактором є простота реалізації. Ботнети часто використовують повторно існуючі реалізації, тому спрощуючи розробку. Іншим рушійним фактором можуть бути обмеження мережі. Наприклад, використання протоколу IRC для зв'язку з роботою C&C зменшується, оскільки більшість корпоративних мереж не дозволяють IRC.

Відповідно до існують дві основні категорії протоколів C&C ботнету: 1) існуючий протокол, що означає (повторне) використання існуючого протоколу програми, який був розроблений з іншою метою, або 2) неотеричний протокол,

реалізований спеціально для відповідної ботнету. На додаток до протоколу C&C, ботнету потрібно використовувати додаткові протоколи для виконання бажаних цілей та завдань.

A. IRC

Протокол IRC був спочатку розроблений для додатків в Інтернеті. Це текстовий протокол, який дозволяє клієнтам спілкуватися з сервером, який відповідає за передачу повідомлень чату іншим клієнтам і серверам. Повідомлення можуть обмінюватися або між клієнтом та групою клієнтів, які проживають у так званому каналі, або між двома клієнтами. Канали можуть бути захищені паролем, що дозволяє ботмейстерам запобігати захопленню інших ботнетів. Крім того, підтримується передача файлів, що дозволяє розповсюджувати додаткові бінарні файли, файли конфігурації чи оновлення. Протокол IRC використовує централізовану топологію, простий у здійсненні, має низьку затримку та широко використовується в Інтернеті.

Недоліком IRC є те, що його можна легко заблокувати пристроями безпеки, такими як брандмауери. Крім того, це не часто зустрічається в ділових мережах. Пізніше ботнети з централізованою топологією використовують різні протоколи (наприклад, HTTP).

Один з найбільш ранніх ботнетів під назвою PrettyPark, який з'явився в 1999 році, використовує грубу реалізацію існуючого протоколу IRC для C&C і базується на ідеї ботів IRC.

Пізніші ботнети, такі як GTBot, використовують утиліти сценаріїв клієнта IRC під назвою mIRC, тим самим обмежуючи необхідний час розробки. Навіть пізні ботнети, включаючи Agobot, попередника Phatbot (див. Розділ V), все ще покладаються на IRC.

B. HTTP

HTTP був розроблений для доставки веб-сторінок з веб-серверів до веб-браузерів. Це протокол-відповідь на запит, що означає, що клієнт може зробити запит, на який відповідає відповідь із сервера.

Як головний наслідок, груповий зв’язок із цим протоколом неможливий. Крім того, HTTP має більш високу затримку, ніж IRC, оскільки кожен бот

повинен спеціально вимагати команди з сервера. Як і IRC, HTTP також легко здійснити, оскільки декілька реалізацій сервера та клієнтів доступні безкоштовно або як відкритий код. Приклади включають nginx та Apache HTTP Server, які є реалізацією веб-сервера з відкритим кодом.

Якщо кожен бот в ботнеті реалізує HTTP-сервер і клієнт, HTTP також може використовуватися в топології P2P. З моменту HTTP був розроблений для централізованої топології, слід дотримуватися додаткової обережності, щоб не допустити циклів. Такі петлі шару додатків можуть заливати ботів реплікаційними повідомленнями. Наприклад, це може бути спричинене тим, що кожен бот пересилає повідомлення іншим ботам, що, в свою чергу, може знову пересилати повідомлення назад (див. Рис. 1.3). Додаткові теми, які потребують вирішення, - це надійне спілкування через ботнет та пошук інших ботів. Приклад ненадійного спілкування можна побачити на рис. 1.4.

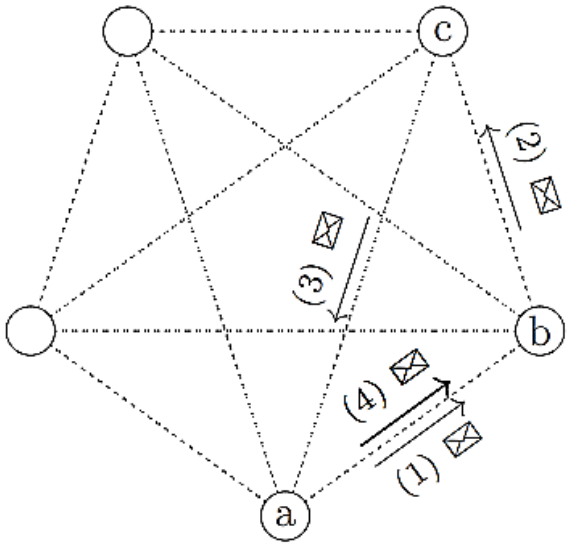

Рисунок 1.3 – Ботнет P2P: Повторне повідомлення. Доставка повідомлень: (1) від вузла a до вузла b, (2) пересилається до c, (3) назад до a, (4) знову до b.

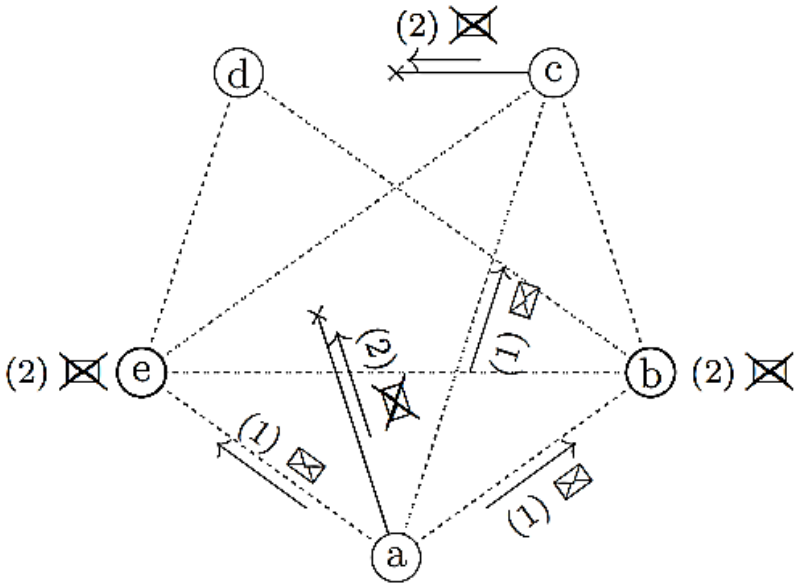

Рисунок 1.4 – Ботнет P2P: повідомлення про втрату. Доставка повідомлення: (1) вузол відправляє повідомлення для кожного іншого вузла; вузол a не пов'язаний з d, (2) вузол c і d не з'єднані; e і b пропустити пересилання повідомлення на d.

У цьому прикладі ботнет не повністю пов'язаний, а додатково боти пропускають переадресацію повідомлення. Ці особливості P2P повинні вирішуватися в додатковому мережевому шарі поверх HTTP.

Приклад рішення можна побачити на рис. 5. У цій моделі додаткові шари вставляються між шаром програми (HTTP) та корисним навантаженням (фактична команда ботнету).

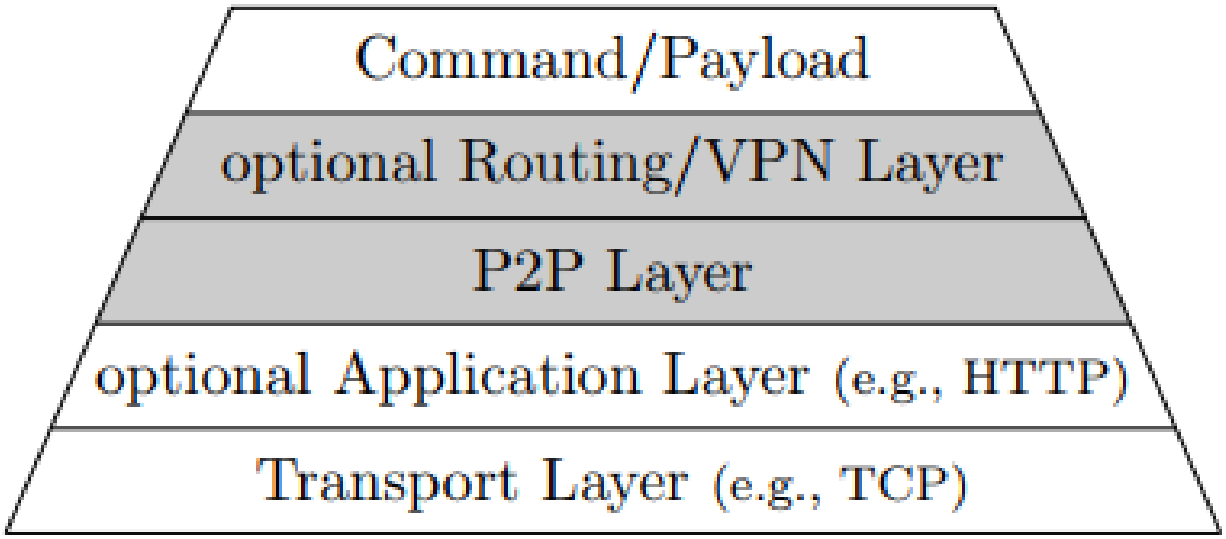

Рисунок 1.5 – Стек протоколів P2P ботнету з протоколом програми, який не підтримує P2P, та додатковим маршрутизатором / рівнем VPN. У разі неотеричного протоколу, прикладний шар не використовується.

Рівень P2P може використовуватися для додавання інформації, необхідної для надійної передачі повідомлень на кожен вузол та запобігання вищезазначеній проблемі із затопленням повідомлень. Необов'язково, додатковий рівень маршрутизації може бути використаний для отримання додаткової інформації, яка дозволяє надсилати команду певному боту, не знаючи необхідного мережевого шляху в мережі P2P. Крім того, повідомлення можуть бути спрямовані через довільну кількість ботів, підвищуючи рівень складності виявлення походження повідомлення. Проблема надійності також може бути вирішена за допомогою цього шару маршрутизації.

Основна перевага використання HTTP полягає в тому, що він переважно використовується в Інтернеті, а тому зв’язок через HTTP вписується в звичайні схеми трафіку.

C. SMB

Ще один протокол, який приховує спілкування за типовими схемами руху в домашній та діловій мережі, - це протокол серверного блоку сервера (SMB). Як і HTTP, SMB є протоколом відповіді на запит і працює, обмінюючись повідомленнями між клієнтом і сервером. Спочатку клієнт повинен автентифікувати себе на сервері за допомогою облікових даних користувачів або як анонімний користувач. Після аутентифікації клієнт може запитати список доступних спільних ресурсів, які є пулами файлів чи служб, або отримати доступ до файлів чи служб (наприклад, спільних принтерів).

Протокол SMB використовується в основному для забезпечення доступу до файлів та принтерів через мережу. Крім того, протокол SMB реалізує іменовані канали, які є механізмом взаємодії InterProcess (IPC), навіть по всій мережі. IPC може використовуватися різними процесами для обміну повідомленнями або виклику певних функцій.

Ще один спосіб використання протоколу SMB - це зараження інших хостів. Якщо комп'ютер обмінюється файлами через протокол SMB, і ці файли підлягають запису, то бот може вводити в ці файли шкідливий код. Щоб заразити комп'ютер цими файлами, користувач повинен їх відкрити. Тому це називається пасивною інфекцією (див. Також розділ VI-C).

Протокол SMB часто блокується на шлюзі, що забезпечує доступ до Інтернету. Отже, цей механізм використовується здебільшого для зв'язку в локальних мережах.

Прикладами ботнетів, що використовують протокол SMB, є Regin та Duqu 2.0, який також здатний емулювати сервер SMB для збору паролів від спроб аутентифікації. Крім того, Phatbot може розповсюджуватись шляхом інфікування файлів, що передаються через протокол SMB,. Більш детально про приклади ботнету можна побачити у розділі V.

D. Протоколи P2P

Існуючі протоколи P2P, які спочатку були розроблені для спільного використання файлів та спільної роботи, можуть використовуватися для ботнетів з топологією P2P. Вони можуть використовуватися для формування окремої мережі P2P або для приховування як частини однієї з існуючих мереж (див. Рис. 6).

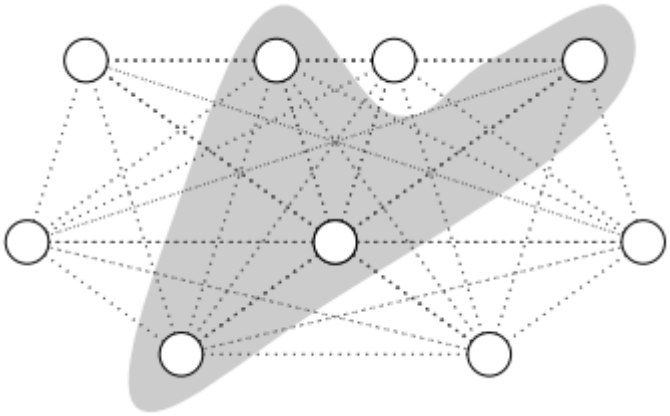

Рисунок 1.6 – Ботнет P2P (сірий), що використовує частини існуючої мережі P2P для C&C.

Прикладом існуючого протоколу є WASTE - протокол, спочатку призначений для співпраці. Ще один популярний клас протоколів P2P заснований на Kademlia. Ці протоколи можуть використовуватися для зберігання значень або хешів розподіленим способом та стійким до помилок.

Ботнет з частинами існуючої мережі - це Phatbot, який використовує кеш-сервери Gnutella для пошуку своїх однолітків. C&C частина Phatbot використовує протокол WASTE, але без шифрування, що спрощує спілкування. Прикладами протоколів на основі Kademlia є варіанти P2P Zeus, які використовують ту саму метрику відстані, що і Kademlia, і Storm, яка використовує протокол Overnet замість DNS. Overnet - це мережа та протокол P2P на базі Kademlia, що включає власний рівень маршрутизації та адреси. В основному використовується для обміну файлами.

E. Комунікація через приховані канали

1) Приховані канали використовуються для приховування існування зв'язку. Це досягається використанням засобів зв’язку, які не призначені для використання в спілкуванні. Для екземпляра біт НЕ використовується заголовок, наприклад, TCP або IP, може бути використано для передачі інформації.

Мережа Tribe Flood використовує поле ідентифікатора пакетів відповідей exo-сигналів ICMP як команду для C&C,. Незважаючи на зусилля, щоб приховати комунікацію, ці пакети можна легко виявити та маніпулювати, шукаючи відповіді на exo-відповідь ICMP з відсутніми запитами exo в ICMP.

2) Шифрування: Шифрування дозволяє приховати передані команди та навіть використаний протокол від детекторів. Це перемагає кожен тип мережевого детектора, якому потрібно перевірити корисне навантаження. Спільним для кожного алгоритму шифрування є те, що за допомогою однієї клавіші або декількох клавіш чіткий текст може бути перетворений в текст шифру і навпаки. Алгоритми шифрування можуть бути потоковими або блоковими. За допомогою алгоритмів шифрування на основі потоку можливо шифрування одиночних байтів, тоді як алгоритми шифрування на основі блоку можуть шифрувати лише блоки фіксованого розміру.

Крім того, алгоритми можуть бути або симетричними, тобто той самий ключ використовується для шифрування та дешифрування, або асиметричний, де використовуються два доповнюючих ключі.

Найпростіший спосіб шифрування даних - це ексклюзивна операція або (XOR), де обчислюється ексклюзивна диз'юнкція між кожним бітом чіткого тексту та кожним бітом ключа. Повторення цієї операції на отриманому тексті шифру призводить до отримання оригінального чіткого тексту. Це простий алгоритм шифрування на основі симетричного потоку. Одним з недоліків є те, що використання тієї самої клавіші з тим самим чітким текстом призводить до того ж тексту шифру. Це означає, що якщо ботнет використовує заздалегідь визначений ключ та фіксовані команди, отримані зашифровані команди знову можуть бути ідентифіковані. Навіть якщо один і той же ключ використовується для шифрування декількох зразків, можна обчислити оригінальний ключ лише з текстів шифру.

Ці проблеми можна вирішити за допомогою нескінченного ключа, який є ідеєю потокових шифрів, які створюють нескінченну псевдо випадкову послідовність, яка визначається ключем. Потім ця послідовність використовується при операції XOR над прозорим текстом. Прикладом такої шифри є Rivest Cipher 4 (RC4).

Блокові шифри працюють у кілька раундів над чітким текстом, які складаються з різних комбінацій операцій XOR, підстановок та перестановок. Приклади симетричних блокових шифрів є покращений стандарт шифрування (AES), (RC5), (TEA), і (XTEA).

Асиметричні шифри ґрунтуються на числових задачах, які важко вирішити (наприклад, проста факторизація). Одним з прикладів є RSA. Оскільки ці алгоритми обчислювально дорогі, вони використовуються в основному для обміну ключами для симетричного зашифрованого з'єднання та для підписання.

Без знання ключа не можна отримати чіткий текст із тексту шифру. Винятки можуть виникнути, якщо виявлені помилки в алгоритмі або помилки в реалізації або порушені вимоги алгоритму. Тому доцільно використовувати існуючі реалізації, такі як безпека транспортного рівня (TLS). TLS обробляє шифрування

та необов'язково аутентифікацію. Один з часто використовуваних протоколів - це захищений протокол передачі тексту гіпертонів (HTTPS), який є HTTP шарувато поверх TLS,. Існує кілька бібліотек, які реалізують усі необхідні функції.

Прикладами ботнетів, що використовують хибні реалізації алгоритмів шифрування, є мережа Tribe Flood Network 2000, яка спричинила закінчення кожного мережевого пакета, створеного ботнетом, символами ASCII 'A', і Blackenergy 2, які повторно використовували однаковий ключ для кожного повідомлення. Версія 3 Blackenergy змінила цю реалізацію на використання HTTPS. Інший приклад, в якому використовується HTTPS, є Adwind.

Версії 3 і 4 версії Sality використовують шифрування RC4 для повідомлень протоколу P2P. Оскільки ключ для шифрування завжди передається в одному і тому ж повідомленні, він використовується виключно для придушення зв'язку та перемоги підходів виявлення на основі підпису.

3) Кілька протоколів: для подальшого підвищення вторгнення ботнету замість одного C&C протоколу можна використовувати різні протоколи всередині ботнету. Цього можна досягти, побудувавши змінний рівень зв'язку, здатний використовувати різні протоколи зв'язку.

Duqu 2.0 - це ботнет, який здатний використовувати протоколи HTTP, SMB з назвою, HTTPS або неотеричний протокол TCP для трафіку C&C. Ці протоколи можуть поєднуватися з декількома методами стиснення та шифрування. За допомогою цього величезного вибору протоколів ботнет може використовувати найбільш переважний протокол цільової мережі.

4) Стиснення: Ще одним способом придушити спілкування є використання стиснення. Алгоритм стиснення зменшує введення згідно алгоритму (наприклад, замініть повторювані послідовності послідовністю та кількістю повторень, використовуйте індекс у словнику замість самої послідовності). На відміну від шифрування, жоден ключ не використовується, і тому кожен, хто знає алгоритм стиснення, може розпакувати дані. Отже, стиснення може лише придушити дані.

Прикладами ботнетів, що використовують стиснення, є Zeus, який використовує zlib, і Duqu 2.0, який може використовувати кілька різних алгоритмів.

5) Стеганографія: Додатковою технікою обфускування є стеганографія, що означає приховування інформації в контейнерах, які не призначені для спілкування. Ця методика подібна до прихованих каналів, але замість мережевих протоколів файли використовуються як носії (наприклад, зображення, відео, документи). Ботнет Duqu 2.0 використовує цю техніку, якщо незашифрований HTTP використовується як протокол зв'язку C&C. Nagaraja та ін. запропонували ботнет, який використовує існуючу соціальну мережу як канал C&C і приховує спілкування у зображеннях Joint Photographic Experts Group (JPEG). Оскільки цей метод покладається на завантаження зображення користувачем, кількість можливих переданих повідомлень обмежена.

Більш новий підхід Compagno та співавт. приховує зв'язок C&C у текстових повідомленнях, які обмінюються користувачами через соціальні мережі. Цей підхід кодує повідомлення C&C з кодами управління, які не відображаються веб-браузерами. Крім того, інформація кодується за допомогою перестановки кодових комбінацій, які призводять до того ж візуального зображення. Обидва підходи підсилюють зв'язок C&C на законний трафік користувача, перемагаючи більшість мережевих механізмів виявлення ботнетів.

6) Оскільки ботнети в основному використовуються для шкідливих дій, ботмайстру потрібно залишатися максимально прихованим. Одним із способів підвищення секретності є використання крокових каменів, які є хостами, які використовуються як проміжний хміль для зв'язку між ботмейстером та сервером C&C (див. рис. 1.7).

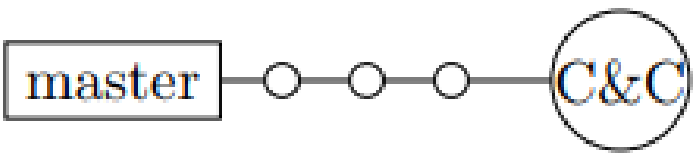

Рисунок 1.7 – Ланки між Botmaster і сервером C&C,
які покращують прихованість Botmaster.

У топології P2P це також можна досягти за допомогою ботів у мережі. Для імітації поведінки проміжних хостів між ботмейстером та джерелом команд управління може бути використаний додатковий шар маршрутизації (див. Розділ IV-B). Цей шар маршрутизації додає можливість задати кількість проміжних хостів, через які команда повинна бути ретрансльована, поки не буде дозволено досягти свого цільового призначення.

Slapper ботнет використовує цей механізм, за допомогою вибору випадкового наступного кроку на кожному кроці.

F. Оперативні протоколи.

Крім C&C трафіку, ботнетам потрібно також спілкуватися з іншими службами для досягнення своїх цілей. Це досягається за допомогою оперативних протоколів, які можна відрізняти від використовуваного протоколу C&C та топології.

Для того, щоб надсилати спам, який є небажаним трафіком електронної пошти, ботнетам потрібно мати можливість спілкуватися за допомогою простого протоколу передачі пошти (SMTP). Протокол SMTP - це дуже простий протокол відповіді на запит, де кожен може підключитися до поштового сервера та надсилати електронні листи. Більшість серверів потребують автентифікації лише для надсилання електронних листів, призначених для іншого сервера, що є основною причиною спаму. Тому багато постачальників електронної пошти намагаються заблокувати електронні листи, що походять від домашніх IP-адрес.

HTTP та HTTPS також необхідні для впровадження шахрайства з натисканням. Шахрайство при натисканні - це схема, яка використовує ботів для доступу до рекламних URL-адрес, що, в свою чергу, приносить дохід.

Крім шахрайства при натисканні, для виконання DDoS-атак також необхідний протокол HTTP, а також UDP або TCP. DDoS-атака - це атака, яка намагається перевантажити мережевий сервіс. Цього можна досягти, надіславши велику кількість пакетів цільовому хосту або мережі. Оскільки пропускна здатність обмежена, це призведе до відключення послуги, споживаючи всю пропускну здатність при атаці трафіку. Ще один спосіб перевантажити послугу

- це надсилання обчислювально інтенсивних запитів до цільової програми. Ці запити використовуватимуть усю наявну потужність обробки, відводячи послугу безвідповідально.

РОЗДІЛ 2
ВЛАСТИВОСТІ БОТНЕТІВ

2.1 Огляд властивостей відомих ботнетів

Мережі ботнетів використовуються для різних цілей, включаючи інструмент віддаленого адміністрування (RAT), DDoS-атаки, розповсюдження спаму, шахрайство з натисканнями, порушення роботи служби, розповсюдження зловмисного програмного забезпечення або як мережа доставки вмісту загального призначення (CDN). Ці цілі, розповсюдження та підтримка ботнету мають різні комунікаційні потреби. Для повного розуміння моделей комунікацій ботнетів у цій роботі проаналізовано дуже різноманітний перелік ботнетів.

Попередні опитування ботнетів, обмежуються ботнетами з конкретною цільовою метою, тоді як більш загальні опитування, лише наводять або згадують ботнетів як приклади. Навпаки, в цій роботі основна увага приділяється конкретизації загальних моделей зв'язку через ботнет. Як наслідок, ця систематика будується на основі навмисно різноманітного списку ботнетів, щоб витягти спільні комунікації різних ботнетів.

Для складання таксономії моделей зв'язку, залежних від завдань, ботнети були проаналізовані відповідно до:

• Топологія ботнету для визначення того, чи впливає топологія на узагальнені схеми зв'язку.

• Призначення ботнету.

• Протоколи C&C, що використовуються, та відношення до мети ботнету.

• Техніка спілкування, пов'язана з C&C, пов'язана з придушенням та приховуванням техніки та вплив на схеми спілкування.

• Завдання, які можна виконати.

• Зв'язок з C&C, необхідний для виконання цих завдань.

• Мережеве спілкування, необхідне для виконання завдань на ботнет.

Наведений нижче список різних ботнетів (див. Також таблицю II) включає ботнетів з різним ступенем витонченості, різних підходів до спілкування, різних топологій та різних цілей. Ботнети в списку були вибрані як представники свого класу, або тому, що вони були першими, включаючи нові методи, або тому, що вони використовують особливо складний варіант певної техніки. Були обрані ще ботнети, які пройшли радикальну еволюцію протягом свого життя. Ця еволюція представляє загальну тенденцію ботнету. Крім того, старі та нові ботнети були обрані для того, щоб перевірити, чи і як змінювались форми спілкування з часом.

1) Adwind: Особливістю цієї ботнети є те, що вона продається як програмне забезпечення як послуга. Це означає, що програмне забезпечення розміщене постачальником, і купувати підписку можна лише протягом обмеженого часу. Тому Adwind вимагає використання централізованої структури з обмеженою кількістю серверів C&C для перевірки ліцензії. Отже, незважаючи на те, що більшість сучасних ботнетів використовують P2P або гібридні топології, цей ботнет все ще використовує централізовану структуру з обмеженою кількістю серверів C&C. Ця схема ліцензування також може бути досягнута за допомогою гібридної топології, але це збільшить витрати на впровадження та експлуатацію. Використання гібридної топології також накладає обмеження на мінімальний розмір ботнету, оскільки для ретрансляції зв'язку потрібна кількість ботів. Це робить бізнес-модель програмного забезпечення як послуги для ботнету занадто дорогою, щоб використовувати його в цілеспрямованій атаці. Оскільки існують також ребрендовані, скопійовані та взломані версії, існує набагато більше серверів C&C, ніж початкові.

Таблиця 1.2 – Огляд відомих ботнетів і їх властивостей

Botnet	Рік	Протокол C&C	Топологія	Шифрування	Склад.	Призначення	Споріднені / сімейні
Адвінд	2012 рік	HTTPS	центральний	TLS	-	RAT, модульний	AlienSpy, Frutas, jFrutas, Unrecom, Sockrat, JSocket, jRat
Blackenergy	2007 рік	HTTP, google +	центральний	v1: -; v2 RC4 [a]; v3: TLS	-	DDoS, модульний	кілька версій
Конфікер	2008 рік	HTTP, SMB, UDP, TCP	центральний; пізніше P2P, центральний	RC4 (P2P)	-	поширювати зловмисне програмне забезпечення	Скорочення, кілька версій (A – E)
Duqu 2.0	2014 рік	HTTP, SMB названа труба, HTTPS, TCP	центральний	Камелія, AES, XTEA, RC4, XOR, TLS	LZJB, LZF, FastLZ, LZO	RAT, модульний	Duqu, Flame, Gauss, Stuxnet, miniFlame
Шахтар	2010 рік	HTTP	центральний; пізніше гібридний P2P / центральний	-	-	PPI, (Bitcoin), DDoS, крадуть тотожності	кілька версій
Фатбот	2004 рік	ВІДХОДИ [b], IRC	P2P, центральний	-	-	красти дані, DDoS, розгорніть зловмисне програмне забезпечення	Agobot, Gaobot, Forbot, XtrmBot, понад 1000 варіантів
Перехід	2003 рік	UDP, TCP, SMB, HTTP, HTTPS	P2P, VPN	RC5, TLS	-	модульний	QWERTY
Русток	2005 рік	HTTP	центральний	RC4 [c], TEA	-	спам	кілька варіантів

Продовження таблиці 1.2

Сільність	2003 рік	електронна пошта, HTTP, UDP, TCP	центральний; пізніше P2P	-	-	завантажують програмне забезпечення, крадуть дані	кілька версій
Синіт	2003 рік	UDP	P2P	-	-	поширювати зловмисне програмне забезпечення	Calyps.a, Каліпсо
Шляхетник	2002 рік	UDP	P2P, маршрутизація	-	-	DDoS, завантажте програмне забезпечення	Scalper, Cinik, Unlock, кілька версій
Буря	2007 рік	Overnet [d], HTTP	гібридний P2P / центральний	XOR	зліб	спам, DDoS	Peacomm, Nuwar, Tibs, Zhelatin
Stuxnet	2005 рік	HTTP, TCP, SMB труба	P2P, центральний	XOR	-	порушити SCADA	Duqu, Duqu 2.0, Полум'я, Гаус, miniFlame
TFN [e]	1999 рік	TCP, UDP, ICMP	центральний	оригінал: -; 2000 рік: CAST-256	-	DDoS	Стахельдрат
Тріно	1999 рік	TCP, UDP	центральний	-	-	DDoS	Штехельдрат, Trin00
Валедак	2008 рік	HTTP	гібридний P2P / центральний	AES, RSA	bzip2	спам, завантажуйте програмне забезпечення	-
Нульовий доступ	2011 рік	UDP, TCP	P2P	RC4, звичай XOR	-	натисніть шахрайство, завантажте програмне забезпечення	Smiscer, Max ++ руткіт, Сірефеф
Зевс	2006 рік	UDP, TCP, HTTP	центральний; пізніше гібридний P2P / центральний	RC4	зліб	вкрасти повноваження	кілька варіантів, Gameover, Murofet, Лікат

Це одна з причин, чому чорні списки для блокування серверів C&C мають лише обмежене значення. Adwind в основному використовується як RAT в цілеспрямованих атаках. RAT дозволяє повному керуванню робочим комп'ютером, на якому працює виконуваний бот. Залежно від включених плагінів, Adwind можна використовувати для видалення файлів, завантаження файлів, модифікації файлів, спостереження за власником комп'ютера, включаючи доступ до приєднаних пристроїв, таких як веб-камери, розшифрування конфіденційних даних, включаючи облікові дані для входу, і навіть доступ до всіх даних на мобільних телефонах. Ця функціональність в ботнеті Adwind використовувалася для отримання доступу до фінансових установ, контролю за державними чиновниками чи простими схемами вимагання грошей.

Окрім того, що використовується як RAT, Adwind також передбачає інші сценарії завдяки модульній конструкції. Виявлені плагіни включають шахтар криптовалюти та проксі. Крім того, він містить java obfuscater для ухилення від виявлення і може працювати на OSX, Windows, Linux та Android. Комунікація захищена через HTTPS та використовує нестандартні порти, а це означає, що використовується інший порт, ніж 443, що пропонується стандартом для HTTPS.

2) Blackenergy. Цей ботнет існує в декількох версіях. Кожна версія використовує централізовану топологію та HTTP-сервер як C&C. Ранні версії не використовували шифрування і були здатні лише до DDoS-атак. Пізніші версії захищають зв’язок із слабкою реалізацією RC4 та додають кілька модулів, що дозволяють використовувати ботнет для зриву сервісу (наприклад, маршрутизатори, контрольний контроль та придбання даних (SCADA)).

Системи SCADA застосовуються для управління процесами у великих масштабах промислових систем управління. Оскільки ці системи іноді географічно розповсюджені, їх потрібно підключити до Інтернету. Крім того, ці системи часто підключаються до офісних мереж для експорту даних в облік або оновлення процесів дистанційно . Blackenergy також може використовуватися як Advanced Persistent Threat (APT), який є цілеспрямованою атакою, коли

кваліфікований противник отримує доступ до мережі і залишається там непоміченим. Це дозволяє здійснювати подальшу розвідку та доступ до різних рівнів мережі, щоб остаточно досягти основної мети (наприклад, ексфільтрація даних, що перериває нормальну роботу) .

В даний час остання версія використовує HTTPS для зв'язку і був визначений модуль, який може використовувати соціальну мережу для спілкування. Blackenergy є однією з тенденцій еволюції ботнетів. Він розпочався як простий ботнет з DDoS з незашифрованим C&C-зв’язком і використовував централізовану топологію. Пізніше Blackenergy був модульований, що дозволило додатково використовувати цілі, крім DDoS. Під час цього процесу було додано шифрування з різним рівнем складності. Незважаючи на простий запуск, в даний час він використовується навіть у високопрофільних атаках на електромережу .

3) Цей ботнет розвивався дуже швидко у 2008 та 2009 роках, зазнаючи п'яти різних версій (названих A – E). Усі попередні версії отримали оновлення, оновивши ботів до найновішої версії. Крім того, версія E встановила зловмисне програмне забезпечення на заражених комп'ютерах та видалила себе 3 травня 2009 року.

Ранні версії використовували централізовану топологію ботнету, використовуючи DGA для генерації доменних імен. Оскільки DGA використовував поточний час як основний, боти надсилали HTTP-запити на популярні веб-сайти для отримання поточного часу. Єдиною метою центрального сервера був розподіл бінарних файлів. Оскільки ці двійкові файли завантажувались безпосередньо в ботнет, можна було додати додаткову функціональність. Після того, як DGA була інженерна, і результати були використані для зриву ботнету, DGA було оновлено. Крім того, пізніші версії Conficker також містили компонент P2P для розповсюдження бінарних файлів. За конкретних обставин централізовану частину було вимкнено та використано лише компонент P2P.

Щоб загартувати ботнет від противників, Conficker використовував підписи RSA для кожного бінарного оновлення та використовував шифрування RC4 для протоколу P2P. Крім того, компонент P2P використовував випадкове сканування для однолітків. Крім того, порти, що використовуються компонентом P2P, були сформовані з номера поточного тижня та IP-адреси.

4) Duqu 2.0: Цей витончений ботнет був вперше виявлений Лабораторією Касперського під час ревізії безпеки у власній мережі. Він високомодульний, може використовуватися як RAT і має декілька прийомів, які роблять його особливо придатним як APT. Щоб уникнути виявлення, цим ботнетом можна використовувати кілька методів шифрування, включаючи камелію, AES, XTEA, RC4 та чіткий XOR з фіксованим ключем.

Крім того, доступні декілька проксі-модулів, які дозволяють переводити з будь-якого використовуваного протоколу протокол HTTP, HTTPS, SMB з назвою і неотеріальний протокол TCP через різні порти. Більше того, мережевий трафік через HTTP приховано за допомогою стеганографії (див. Розділ IV-F5).

Подальше управління мережевим трафіком можна досягти, використовуючи модулі стиснення, які підтримують LZJB, LZF, FastLZ та LZO. Цей APT має унікальну поведінку, що модуль збереження використовується лише на серверах, а це означає, що інші хости потребують повторного зараження після кожного перезавантаження, викликаючи додатковий трафік.

5) Майнер: Ранні версії бонет- мережі Miner використовують централізований підхід з декількома серверами C&C. Пізніші версії замінюють цю топологію шаруватою структурою, що складається з статичних головних серверів C&C, шару P2P / CDN та робочого рівня. Ясний текст HTTP через порт 8080 використовується для зв'язку. Передані бінарні файли підписуються хешем повідомлень Digest 5 (MD5) зашифрованим RSA.

Функції хешування – це односторонній алгоритм, який перетворює введення довільного розміру в більш короткі числа, що представляють вихідний вміст. Хеш двох однакових копій призводить до того ж числа. Це може бути використано як ярлик для перевірки вмісту файлів .

Miner ботнет, як сервіс Pay-Per-Install (PPI), використовується як CDN для додаткових шкідливих програм і для запуску DDoS – атак. До найбільш використовуваних додаткових зловмисних програм належать шахтарі біткойн та програмне забезпечення, здатне викрадати особистість соцмереж.

Miner ботнет є прикладом ботнету, який приборкує обчислювальну потужність своїх жертв для отримання фінансової вигоди (наприклад, видобуток Bitcoins).

6) Phatbot: Цей ботнет є спадкоємцем Agobot, який використовував централізовану топологію над IRC. У цій еволюції додається додатковий механізм зв'язку з використанням топології P2P через протокол WASTE (див. Розділ IV-D для опису ВІДХОДІВ). Частина шифрування протоколу видаляється, щоб уникнути проблеми розподілу ключів, яка не є частиною ВІДХОДУ. Ботнет використовує кеш-сервери Gnutella для пошуку своїх однолітків. Бінарні файли, необхідні під час стадії зараження та оновлення, передаються через звичайний текстовий протокол передачі файлів (FTP). FTP – це простий текстовий протокол відповіді на запит, який використовується для передачі файлів між сервером і клієнтом .

Ботнет використовує модульну архітектуру для свого корисного навантаження, що дозволяє проводити різні шкідливі дії. Основними цілями цього ботнету є крадіжка даних, виконання DDoS-атак та розгортання додаткового зловмисного програмного забезпечення.

Phatbot – одна з найдавніших спроб замінити централізовану топологію, яка домінувала на той час, P2P. Ця спроба, здається, не вдалася, оскільки, незважаючи на великі модифікації решти бота, у більш пізній версії ботнету протокол WASTE ніколи не оновлювався. Крім того, Phatbot все ще містить протокол IRC, а протокол WASTE був розроблений для невеликих мереж з 10-50 вузлами.

Оскільки вихідний код оригіналів Agobot та Phatbot був опублікований в Інтернеті, існує кілька тисяч варіантів цього ботнету.

7) Regin: Хоча перший відомий зразок Regin датується 2003 роком, цей ботнет був вперше виявлений у 2012 році. Це спричинено вдосконаленою сервісно-орієнтованою архітектурою (SOA), яка є використовується для впровадження високомодульної ботнети, де кожен модуль спілкується через віртуальну приватну мережу (VPN). Завдяки такому дизайну функціональність може бути розповсюджена на декілька ботів та динамічно змінена під час роботи. VPN може переноситися через неотеричний протокол UDP, неотеричний протокол TCP, SMB, HTTP та HTTPS.

Щоб ініціалізувати з'єднання між двома ботами, один бот надсилає послідовність стукання другому боту. Потім два боти домовляються про те, який транспортний протокол слід вибрати для зв'язку C&C. Ця послідовність збиття може використовувати будь-який із згаданих вище протоколів та додатково ICMP. За допомогою цієї методики спілкування адаптується для відповідності нешкідливому трафіку, що відбувається в цільовій мережі. Крім того, ця методика може використовуватися для ексфільтрації даних через брандмауері, оскільки може бути обраний протокол, дозволений брандмауером. Комунікація захищена TLS, включаючи сертифікати для аутентифікації. Весь механізм використовується для побудови мережі P2P і в основному використовується як APT.

Хоча Duqu 2.0 також може використовувати різні протоколи між ботами, Regin здатний обробляти це автоматично. Крім того, накладений VPN приховує цю складність від оператора. Duqu 2.0 – це єдиний відомий на сьогодні ботнет, який використовує SOA для своїх модулів.

8) Rustock: Ботнет Rustock використовується для розповсюдження спаму. Більш ранні версії використовують HTTP для зв'язку та зашифровують передані дані за допомогою RC4, включаючи користувацький обмін ключами. Пізніші версії замінюють HTTP зашифрованим TEA зашифрованим неотеричним протоколом, замаскованим під трафік HTTPS, ключами, отриманими з системної інформації зараженого хоста.

Кожна версія використовує центральний сервер C&C для координації. Якщо налаштований C&C-сервер неможливо отримати, Rustock намагається зв’язатися з адресами серверів C&C, генерованими DGA.

Крім надсилання спаму через SMTP, цей ботнет також може використовувати веб-сервіси пошти (наприклад, Windows Live Hotmail) з викраденими обліковими даними, що збільшує труднощі виявлення ботнету за допомогою спостереження за мережевим трафіком.

Rustock добре помітний в Інтернеті, оскільки намагається надіслати якомога більше спаму. Незважаючи на цей факт, потрібні спільні зусилля правоохоронних органів, кількох охоронних фірм та фармацевтичної групи, щоб знищити ботнет через захоплення декількох серверів C&C по всьому світу одночасно.

9) Sality: Ранні версії Sality використовуються для крадіжки інформації, включаючи паролі та дані, отримані за допомогою журналу реєстрації даних. Зібрані дані надсилаються зловмиснику електронною поштою, і контроль над зараженими хостами неможливий. Пізніші версії розбивають зловмисне програмне забезпечення на ботнет, здатний встановлювати додаткове програмне забезпечення та корисний вантаж, який забезпечує можливість крадіжки інформації. Цей простий ботнет використовує чіткий текстовий HTTP для зв'язку з центральним сервером C&C. Оскільки URL-адреси кодуються жорстко, зняття ботнету легко досягти, відключивши центральний сервер. Щоб запобігти цьому, топологію Sality було змінено на P2P із зашифрованою RC4 комунікацією через псевдовипадкові породні порти UDP. Боти розбиваються на робочих роботів, що виконують завдання ботнету, і супер-однолітків, що використовуються виключно для цілей C&C зв'язку.

Основна мета останньої версії Sality – завантажити додаткове програмне забезпечення для крадіжки інформації та сканування мережі.

Sality розроблений так, щоб він був стійким до зняття. Код P2P зберігає список однолітків, який оновлюється кожні 40 хвилин. Це унеможливлює ін’єкцію поганих однолітків. З метою безпеки ці списки однолітків обмежені до

1000 однолітків на бота, тому заважають одному об'єкту знати великі частини мережі. Додатково підписуються двійкові завантаження.

10) Sinit: Основна мета цього ботнету – слугувати CDN, який використовується для розповсюдження додаткового зловмисного програмного забезпечення. Sinit використовує неотеричний протокол поверх UDP. Топологія ботнету – P2P, і боти знаходять однолітків шляхом випадкового сканування в Інтернеті. Порт 53 UDP використовується для зв'язку, який використовує той самий порт і протокол, який використовує DNS. Це веде до однозначного виявлення помилково тракту.

Розподілені бінарні файли цифрово підписуються для запобігання іноземного коду. Окрім протоколу P2P, ботнет включає невеликий веб-сервер, який забезпечує лише завантаження двійкового файлу для завантаження, таким чином поширюючи ботнет.

Синіт – одна з ранніх спроб побудови топології P2P за допомогою неотеричного протоколу.

11) Slapper: Ботнет Slapper є спадкоємцем ботнету Scalper, який використовує топологію P2P і здатний поширювати спам, виконувати довільні команди на зараженому хості та виконувати DDoS-атаки. Протокол P2P зроблений більш надійним та ефективним порівняно з попередником. Це обробляється додатковим рівнем надійності в протоколі, який використовується для підтвердження. Додатковий шар маршрутизації здатний направляти повідомлення через декілька ботів, перш ніж воно досягне цільового бота. Це дозволяє приховати походження повідомлень. Шлях маршрутизації також може перешкоджати спробам виявлення на основі кореляції, оскільки він дозволяє ботам в одній мережі отримувати одне і те ж повідомлення від різних ботів.

12) Буря: Цей ботнет використовує гібридну P2P / централізовану архітектуру, що складається з центрального сервера C&C, прихованого за проксі-серверами. Ці проксі-сервери підключені до існуючої мережі P2P Overnet (див. Також розділ IV-D). Мережа P2P використовується як заміна DNS, таким чином переховуючись від детекторів на основі DNS. Інформація, що зберігається

в Overnet, має лише 16 байт і шифрується статичним 16-байтним ключем за допомогою XOR.

Для виконання DDoS-атак та розповсюдження спаму використовуються лише робочі боти, тоді як інші боти відповідають за створення подвійної потокової мережі для ботнету в декількох шарах. Файли поширюються за допомогою HTTP та команд через неотерічний текстовий протокол TCP. Storm має можливість виявляти спроби сканування та запускати DDoS-атаки, щоб запобігти детальному дослідженню ботнету.

13) Stuxnet: Основна мета цього ботнету – порушити конкретну інфраструктуру SCADA. Цей ботнет має лише обмежені можливості управління та використовує мережу P2P для оновлень у мережах, що не мають підключення до Інтернету, HTTP у мережах з підключенням до Інтернету, а також може поширюватись та оновлюватись через інфіковані накопичувачі Universal Serial Bus (USB). Поєднання накопичених USB-накопичувачів та мережевого зв’язку дозволяє цій ботнеті навіть заражати повітряні системи (див. Також розділ VI-C2) та пропонує можливість керувати цими мережами ціною дуже високої ненадійності та дуже високої затримки. Зв'язок P2P шифрується XOR статичним ключем і працює поверх неотеріального протоколу TCP або труби з назвою SMB. Stuxnet дуже складний, вважається, що він орієнтований на єдиний промисловий об'єкт, і його нібито замовляє або є автором розвідувальної служби національної держави.

14) Tribe Flood Network: Мережа затоплення Tribe була створена наприкінці 1999 року як показовий випадок, щоб продемонструвати можливість складного інструменту DDoS-атаки. У ранніх версіях використовуються пакети відповідей Echo ICMP

для зв'язку, з командою, закодованою в полі ідентифікатора, щоб уникнути виявлення. Пізніша версія Tribe Flood Network 2000 змінює C&C-зв’язок на неотеричний зашифрований протокол CAST-256, що транспортується через UDP, а також може працювати у Windows, Linux та Solaris. Стійкість ще більше посилюється, опускаючи визнання. Для усунення потенційної ненадійності

каналу зв'язку команди надсилаються ботам кілька разів. Ботнет можна додатково замаскувати, відправивши підроблені пакети команд у підроблені пункти призначення. Хоча цей ботнет є одним із самих ранніх, він вже містить сучасні приховувальні методи, такі як приховані канали та шифрування.

15) Trinoo: Trinoo був розроблений на початку 1999 року для запуску координованих DDoS-атак. Він містить компонент сервера C&C під назвою обробник і боти, що називаються агентами. Команди можна надсилати на сервер C&C через простий текстовий протокол TCP на основі тексту. Сервер C&C захищений паролем із жорстким кодуванням пароля, і він намагається повідомити майстра про спроби обгону ботнету. Неотеричний протокол UDP, що містить прості текстові команди та власні підтвердження, використовується як протокол C&C. Як і Tribe Flood Network, Trinoo також дуже рано намагається створити ботнет.

16) Waledac: Цей ботнет використовується в основному для розповсюдження спаму та виконання додаткового зловмисного програмного забезпечення на цільовому хості. Топологія, що використовується для зв'язку, – це гібридний підхід C & C / P2P, що складається з головного сервера C&C, шару релейних вузлів і шару підлеглого вузла. Як і Storm botnet, ця інфраструктура використовується як мережа швидкого потоку. Для спілкування використовуються шифровані HTTP-запити AES та RSA, які додатково стискаються bzip2. Одна особливість, що відрізняє Waledac від інших ботнетів, полягає в тому, що кожен бот генерує ключі та сертифікати RSA. Вони використовуються для шифрування сеансових ключів, які, в свою чергу, використовуються для шифрування AES зв'язку C&C. Крім того, зв'язок C&C сильно затухає за допомогою декількох етапів, таких як стиснення, шифрування, кодування та керування повідомленнями.

17) Zeroaccess: Zeroaccess ботнет використовує мережу P2P для зв'язку. Ця мережа P2P ділиться на супервузли, що використовуються для розподілу файлів, і звичайні вузли. Супервузли повинні бути доступні через Інтернет, і тому вузли оголошуються нормальними вузлами, якщо вони стоять за NAT. В якості

протоколу зв'язку використовується неотеричний протокол UDP і TCP, які зашифровані RC4 або користувацьким шифруванням на основі XOR. Однією з головних цілей цієї ботнети є шахрайство з натисканням. Крім того, цей ботнет використовується для отруєння пошуковими системами – це розміщення шкідливих веб-сайтів, які містять цільові ключові слова. Тому ці сторінки відображаються в результатах пошуку, заманюючи користувачів, які не підозрюють у зловмисному веб-вмісті.

Zeroaccess має дуже обмежений набір команд C&C, які можна використовувати лише для оновлення біт-файлів бота або завантаження додаткових зловмисних програм. Додаткове спілкування, необхідне модулям, повинно здійснюватися самими модулями. Отже, цій ботнеті потрібно кілька каналів C&C, кожен з яких діє по-різному.

18) Зевс: Цей ботнет використовує декілька центральних серверів C&C у ранніх версіях. Для того, щоб зробити ботнет стійким до вилазок сервера C&C, ботнет має DGA, який довільно генерує 1000 імен хостів DNS на день для того, щоб знайти сервери C&C. Оскільки цей алгоритм був виявлений, ботнет Зевса перейшов до топології P2P. Якщо бот не може отримати доступ до будь-яких однолітків через P2P, він повертається до пошуку сервера C&C через DGA. Протокол використовує UDP для повідомлень і TCP для передачі файлів. Зв'язок шифрується за допомогою RC4 та стискається зліб. Під час запуску виконуваного бота однолітки виявляються шляхом підключення до хостів, взятих із жорстко кодованого списку однолітків. Існує кілька тисяч варіантів ботнету, оскільки вихідний код ботнету був опублікований в Інтернеті. Різні варіанти також містять зміни в протоколі, що робить його придатним для визначення подібності в поведінці, хоча використовуються різні протоколи.

2.2 Дослідження комунікації ботнетів

Центральним компонентом ботнетів є спілкування між ботмейстером та ботами. Як пояснено у розділах III та IV, для ботмейстера першорядним

залишається приховане. Тому ботмейстери не спілкуються безпосередньо з ботами, а використовують сервер C&C. Оскільки виявлення ботмайстра потенційно може призвести до притягнення до відповідальності, ботмайстер намагається залишатися прихованим, використовуючи якомога менше трафіку . Боти зазвичай генерують додатковий мережевий трафік під час виконання покладених на них завдань, що збільшує їхню видимість у мережі. Таким чином, виявлення ботнетів на основі мережевого трафіку має найвищі шанси на успіх при спостереженні за трафіком, близьким до ботів. Тому ця робота зосереджена на моделях комунікацій, які можна спостерігати у ботів або біля них.

А. Позначення

Шаблони зв'язку, визначені для ботнетів, представлені узагальненими моделями зв'язку. Шаблон зв'язку – це послідовність обмінних повідомлень, необхідних для досягнення конкретного завдання. Наприклад, для відображення веб-сторінки веб-браузеру потрібно надіслати запит на веб-сервер. Сервер повертає відповідь, що містить вміст запитуваної сторінки. Наприклад, якщо ця веб-сторінка містить зображення, то для кожного зображення слід повторити той самий процес відповіді на запит. Веб-браузери використовують HTTP як мережевий протокол, який був розроблений для виконання цієї схеми зв'язку. Форма візуального подання таких шаблонів надається діаграмами послідовності UML. Приклад діаграми послідовностей, що представляє схему зв'язку веб-браузера, зображений на рис. 2.1.

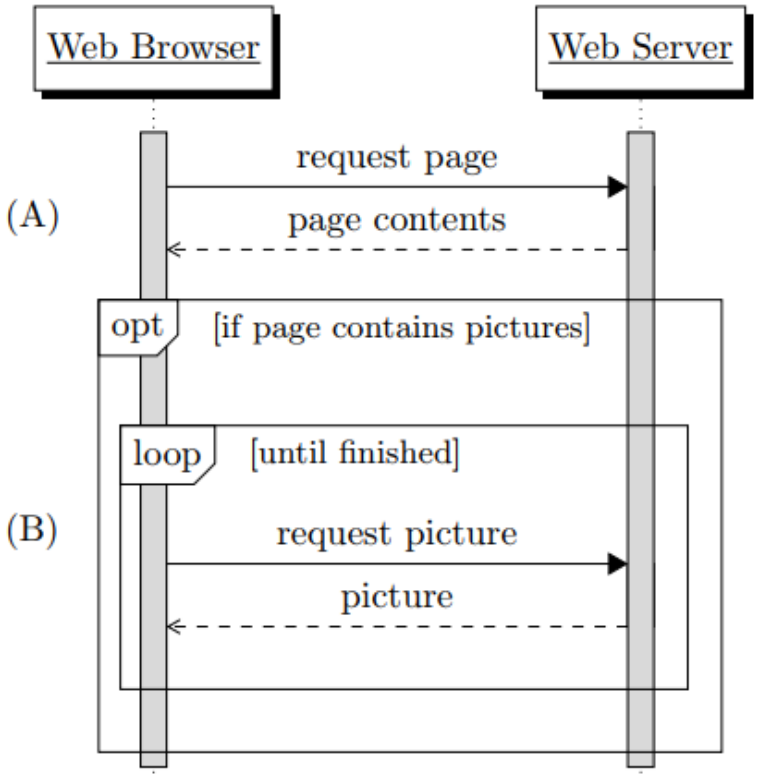

Рисунок 2.1– Приклад схеми зв'язку: Веб-браузер запитує веб-сторінку, яка може містити зображення.

Шаблон зв'язку називається взаємодією в діаграмах послідовностей UML. Основними компонентами такої взаємодії є життєві лінії та повідомлення . Учасники взаємодії представлені рятувальними принципами. У прикладі, наведеному на рис. 2.1, це веб-браузер та веб-сервер. Ці учасники обмінюються повідомленнями протягом усього життя схеми спілкування.

Повідомлення можуть бути синхронними (також називаються дзвінками) або асинхронними. Виклик складається з двох запитів і відповіді на повідомлення. Приклад синхронного виклику, що складається із запиту запиту сторінки (заповнюється стрілка) і відповідь вмісту сторінки (незаповнення стрілки і пунктирна лінія) можна бачити на рис. 2.1 крок (A). Асинхронні повідомлення складаються з запиту без будь-якої відповіді. Вони намальовані незаповненою стрілкою.

Додатковим компонентом є комбінований фрагмент, який складається з операторського роду, обмеження та операнда. Комбіновані фрагменти

намальовані коробкою, що додає операнду, що є довільною комбінацією повідомлень або подальших фрагментів. У лівій верхній частині вікна міститься вид оператора, який вказує тип фрагмента і, можливо, обмеження, укладене квадратними дужками. Приклад комбінованого фрагмента можна побачити на рис. 8 крок (B).

Комбіновані фрагменти, що використовуються в цій роботі, є opt і loop. Фрагменти з оператором вибору opt позначають необов'язкові частини схеми зв'язку . Вміст цього фрагмента виникає лише за умови дотримання обмеження. Наприклад, рис. 2.1 крок (B) застосовується, лише якщо на сторінці є зображення. Цикл фрагмента використовується для відображення повторюваної зв'язку . Для кожного зображення малюнок запиту на запит виконується на кроці (B). Неавтоматичного фрагмент на рис. 8 містить лише фрагмент циклу. Отже, крок (B) також міг бути зображений лише фрагментом циклу, оскільки він виконується для кожної картини, що становить нуль разів у випадку, якщо немає зображень.

Інша частина діаграм послідовностей UML, що використовується в цій роботі, – це використання взаємодії. Ця частина є посиланням на іншу послідовну діаграму. Використання взаємодії малюється як комбінований фрагмент із ref як вид оператора та без обмежень . У цій роботі вони використовуються для позначення частин, які неможливо узагальнити, як частина інфекції. Якщо під час такого посилання повідомлення обмінюються повідомленнями по мережі, це додатково зображено стрілкою в цій роботі (наприклад, рис. 9a). Це не є частиною стандарту UML і використовується як наочний посібник.

На наступних моделях спілкування використовується узагальнений список загальних повідомлень. Ці повідомлення та дзвінки пояснюються у наступному списку:

– координата: повідомлення про координацію. Якщо конкретна задача не автоматизована, це потрібно, щоб доручити боту, що робити.

– scan: сканування мережі. Це може бути запит відлуння ICMP, сканування TCP або UDP або цільове сканування для вразливих служб.

– дані: Загальні дані. Це може бути бот двійкових, довільних файлів або мережевих даних, які передаються з ботнетом або іншими хостами.

– реєстр: Повідомлення, необхідне для згуртування. У випадку топології P2P це потрібно для налаштування зв'язку однорангових. виконувати, збирати / обчислювати, встановлювати / зберігати, заражати: Узагальнені завдання, які виконуються бот-мережею. Вони або не помітні в мережі, або характерні для ботнету.

B. Огляд.

Залежно від топології, трафік C&C, що надходить до бота, може бути відправлений іншим ботом або сервером C&C. Щоб діаграми послідовностей були простими, загальним терміном C&C використовується як партнер зв'язку бота. У централізованій топології C&C є одним із серверів C&C, у P2P топології інший бот, і якщо для проксі використовується проксі. Метою даної роботи є представлення узагальнених моделей комунікації, пов’язаних із виявленням ботів, саме тому розмежування сервера C&C та бота не має значення. Команди також можуть бути передані через інший бот, а не безпосередньо з сервера C&C.

Однією важливою частиною комунікації ботнету є координація. За винятком автоматизованих завдань, кожне завдання виконується лише після отримання команди по каналу C&C. Без цієї координації третіми сторонами можуть відслідковуватися лише ефективні дії ботнету. Отже, чим більше завдань автоматизовано, тим стійкішим є ботнет. Приклади автоматизованих завдань у ботнетах включають збирання інформації для входу в Sality та Storm, а також самооновлення в Stuxnet та модифікацію мережевого трафіку в Майнер та Zeroaccess . Обмеження координації відбувається за рахунок гнучкості. Отже, існує лише обмежена кількість завдань, які автоматизовані в поточних ботнетах, за винятком вузькоспеціалізованих, які були побудовані за спеціальною метою (наприклад, Stuxnet). Одним з винятків цього є ботнет Conficker, який був

здатний лише розповсюджувати та виконувати нові бінарні файли . Оскільки цей процес ініціювали самі боти, кожне завдання в ботнеті було автоматизовано.

Спілкування в ботнетах може відбуватися через метод потягу або метод push. Перетягування означає, що бот вимагає неповторних команд від C&C, тоді як метод push означає, що C&C активно надсилає команди ботам. Використовуваний метод в основному залежить від використовуваного протоколу. Наприклад, HTTP побудований за моделлю відповіді на запит, що означає, що боти можуть використовувати лише метод витягування (див. Розділ IV-B для більш детального пояснення HTTP). Крім того, транспортний протокол може потребувати підтвердження, яке можна розглядати як дані, що перетікають в інший бік. Щоб зробити наступні діаграми взаємодії простими, відмінність між натисканням та перетягуванням та можливими підтвердженнями не залишається. У детекторі, що базується на мережі, це відмінність також може бути виключено, включивши крок фільтра, який перетворює комунікацію на основі тяги в комунікацію на основі поштовху. Включення цього кроку необхідно для незалежності протоколу.

Зв'язок Botnet можна розділити на два етапи розповсюдження та експлуатації. З точки зору зараженого хазяїна та двійкових ботів, ці два етапи можуть відбуватися лише послідовно, оскільки зараження (поширення) має відбутися до того, як бот може вступити в дію.

Перший етап – це розмноження, яке використовується для набору нових ботів. Поширення може відбуватися активно, коли ботнет намагається заразити додаткових жертв, або пасивно, коли бот-двійковий файл розповсюджується за допомогою інших засобів (наприклад, завантаження електронною поштою або driveby). Після розмноження новостворені боти можуть зареєструватися в ботнеті. Якщо розповсюдження відбувається пасивно, а реєстрація пропущена, то цей крок можна спостерігати лише за допомогою глибокої перевірки пакетів (DPI). За допомогою DPI можна витягнути бінарні файли з мережевого трафіку, які потім можуть бути класифіковані як ботові бінарні файли. Цей підхід працює

лише в тому випадку, якщо шифрування не використовується. У розділі VI-C більш детально розглядаються методи поширення.

Другий етап – етап операції. Під час цього етапу ботнет виконує фактичну роботу. Мережевий зв'язок на етапі роботи залежить від призначення ботнету. Розділ VI-D детально пояснює різні режими роботи ботнету.

C. Розмноження

Розмноження – це перший крок у житті бота. Оскільки потенційна потужність ботнету збільшується зі своїм розміром, ботнети постійно намагаються набирати нових ботів. Наприклад, ботнети, здатні DDoS-атаки, можуть спричинити більший трафік із більшою кількістю ботів, ботнети, які використовуються для крадіжки даних, можуть накопичувати більше даних, поширюючись на додаткові хости, а обсяг спаму ботнетів, що надсилають електронний спам, безпосередньо залежить від кількості ботів. З іншого боку, APT повинні залишатися максимально прихованими. Крім того, APT – це цілеспрямовані атаки, тому потрібна лише певна кількість заражених машин. Ці ботнети мають рідкісний крок розповсюдження.

Поширення може відбуватися як активно, так і пасивно, після чого слід етап реєстрації.

Обидві методи розповсюдження заражають хост-жертву виконавчим ботнетом і запускають бота. Ця інфекція може статися на декількох етапах, коли потерпілий спочатку запускає лише невеликий бінарний файл завантаження, який, у свою чергу, отримує фактичний бот, який виконується з ботнету, або навіть зовсім інший ботнет (див. Рис. 9a, крок (D) та рис. 9b крок (B)). Наприклад, ботнет Rustock використовував багатоступеневий інсталятор, де останній етап був зашифрований сервером C&C ключем, сформованим із системної інформації цільового комп'ютера . Ця методика збільшує труднощі для дослідників з отриманням бота, який виконується для зворотної інженерії.

1) Active: Поширення через ботів, які використовують наявні вразливості для зараження додаткових хостів, називається активним поширенням. Це може статися за командою або автоматично. Залежно від реалізації використовуваного

експлуату, може знадобитися крок сканування, перш ніж може відбутися власне зараження. Це може бути використано для мінімізації необхідних обчислювальних ресурсів, оскільки, залежно від атаки, сканування може бути більш легким. Іншим варіантом є використання сканування для зменшення видимості атаки, якщо фактична інфекція використовує протоколи, які рідко використовуються в цільовій мережі.

Повний шаблон спілкування можна побачити на рис. 9а. Перший крок (A) – це етап координації. Координація використовується для налаштування параметрів сканування та експлуатації. Можливі параметри включають, яку використовувати або сканувати використовувати, обмеження на цільових хостах або координацію бот-мережі, щоб запобігти наближенню кількох ботів до однієї цілі. Нарешті, крок координації вказує боту розпочати зараження. Деякі ботнети мають можливість зберігати згадані параметри у файлі конфігурації, який розміщений поряд із ботом або жорстким кодом параметрів у двійковому файлі бота. Цим ботнетам не потрібен крок координації, і тому крок (A) на рис. 9а необов'язково.

Наступним кроком (B) є сканування. Цей крок є необов’язковим, оскільки не кожен ботнет сканує мережу до того, як спробується реально використовувати, покладаючись на попереднє знання жертв. Сканування може складатися з виявлення доступних хостів за допомогою ехо-запитів ICMP або безпосередньо пошуку вразливих служб за допомогою сканування портів UDP або TCP. Ботнети можуть вирішити безпосередньо використовувати етап зараження для сканування, де корисний навантаження буде виконано у списку хостів.

Основним етапом (C) є власне зараження. Під час цього кроку використовуються вразливості програмного забезпечення, що працює на хості жертви. Оскільки цей крок залежить від фактично використаного експлуатації, він зображується як еталонний. Цей крок може бути, наприклад, запитом, що містить зловмисну корисну навантаження, або спробою отримати доступ до послуги шляхом відгадування паролів. Оскільки максимальний розмір біт-

файлів ботів, які можуть передаватися через мережу під час цього кроку, може бути обмежений, може знадобитися додаткове завантаження даних (етап (D)).

Етап завантаження даних, який є частиною так званої багатоступеневої інфекції, також може бути використаний для ускладнення зворотної інженерії бінарного біта. У цьому випадку на ходу потерпілого виконується лише невелика крапельниця. Потім крапельник завантажує фактичний бінарний бот, додатково розшифровує його, встановлює необхідне середовище та виконує його.

Останній крок (E) – це етап реєстрації. Цей крок може бути використаний для відстеження інфекцій ботнету або може знадобитися для підтримки топології ботнету. Більш детальний опис можна знайти в розділі VI-C3.

Прикладами ботнетів, які автоматично заражають додаткові комп'ютери, є Conficker, Sality, Slapper та Stuxnet . Phatbot може використовувати жорстко кодовану конфігурацію або може бути доручений сканувати певні діапазони мережі . Оскільки сканування спричиняє додатковий мережевий трафік, цей крок опускається в ботнетах, таких як Sality .

2) Пасивні: Ботнети можуть також поширюватися через інші вектори, які не контролюють ботнет, як це детально показано на рис. 9б. Це називається пасивним розповсюдженням і включає розповсюдження через електронні листи, веб-сайти чи носії інформації. Спільним для цих пасивних механізмів є те, що користувачі заражають самого жертви самим клацанням або дією.

Розповсюдження електронною поштою може здійснюватися за допомогою так званих фішинг-електронних листів. Фішинговий електронний лист – це електронний лист, який виглядає як законний, але використовується для того, щоб обманути користувача, що отримує, або відкрити вкладений файл, або порушену URL-адресу. Аналогічно зараженням електронною поштою, є також фішинг-сайти, схожі на законні. Один із способів заманити на підозрілий веб-сайт, що не підозрює, – це реалізувати копію файлу веб-сайт із поширеними помилками у веб-адресі.

Замість того, щоб залучати користувача, можна також використовувати вразливості у клієнтському браузері або плагінах у ньому. У цьому випадку

ботнет встановлюється автоматично після відвідування. Це називається завантаженням з драйву. Це може бути використане шляхом зараження веб-сайту, який регулярно використовується цільовими жертвами. Націлення на атаку таким способом називається атакою з поливної ями. Інший спосіб – заразити рекламну (рекламну) мережу, яка надає Об'явам легальні веб-сайти.

Таблиця 2.2 – Способи поширення відомих ботнетів

Botnet	Active	Passive	Coordination	Scanning	Registration
Adwind	–	✓	–	–	✓
Blackenergy	–/✓[a]	✓	–/✓[a]	–/✓[a]	✓
Conficker	✓	✓	–	✓	–
Duqu 2.0	–/✓[a]	✓	–/✓[a]	–/✓[a]	–/✓[b]
Miner	–	✓	–	–	✓
Phatbot	✓	✓	✓	✓	✓
Regin	–/✓[a]	[c]	[c]	[c]	[c]
Rustock	–	✓	–	–	✓
Sality	✓	✓	–	–	✓
Sinit	–	✓	–	–	✓
Slapper	✓	–	–	✓	✓
Storm	–	✓	–	–	✓
Stuxnet	✓	✓	–	✓	✓[d]
TFN[e]	–	✓	–	–	–[f]
Trinoo	–	✓	–	–	✓
Waledac	–	✓	–	–	✓
Zeroaccess	–	✓	–	–	✓
Zeus	–	✓	–	–	✓

Детектор, що базується на мережі, може виявити лише базове спілкування, наприклад отримання електронної пошти чи перегляд веб-сайту. Оскільки користувачеві, можливо, доведеться активно відкривати завантажений файл, час зараження може бути не пов’язаним із часом завантаження. Крім того, спілкування таке ж, як і для нешкідливих електронних листів чи веб-сайтів. Отже, мережевий детектор може виявляти зловмисний вміст лише за допомогою DPI, що стає неможливим при використанні шифрування.

Знімні носії зберігання можуть використовуватися для порушення повітряних зазорів. Повітряні проміжки – це комп'ютери або частини мережі, які з безпеки не підключені до решти мережі. Якщо повна двійкова система ботнету знаходиться на знімному носії інформації, а носій інформації використовується для зв'язку, ця форма розповсюдження не може бути виявлена шляхом виявлення на основі мережевих ботнетів, оскільки мережевий трафік не обмінюється.

Схема зв'язку, яка виникає під час пасивного поширення, можна побачити на рис. 9б. Спочатку двійковий код бота виконується на жертві на етапі (A). Як і при активному розповсюдженні, цей перший двійковий файл також може бути лише частиною виконуваного бота. Тому наступним кроком (B) є завантаження додаткових даних у разі багатоступеневої інфекції. Більш детальне пояснення цього кроку можна побачити у розділі VI-C1. Останній крок (C) – це реєстрація в ботнеті, що буде пояснено у розділі VI-C3.

Цільові фішинг-листи, на яких встановлено ботнет Adwind, використовувались при атаках на різні банківські установи . Waledac ботнет поширює себе, посилаючи великий обсяг інфікованого спаму . Окрім електронної пошти, ботнет Zeus також розповсюджується за допомогою завантажувальних файлів . Stuxnet використовує знімні носії інформації, крім активного розповсюдження для порушення повітряних зазорів, . Конфікер ботнет був також здатний інфікувати знімні носії для подальшого збільшення її охоплення .

3) Реєстрація: Останнім етапом процесу поширення є реєстрація (див. Рис. 9b крок (C) та рис. 9a, крок (E)). Цей крок також називають згуртуванням . Реєстрація додає можливість для моніторингу стану ботнету, розміру, а також необхідна для налаштування на основі push, оскільки сервер повинен знати, куди надсилати команди. У топологіях P2P цей крок також може бути використаний для отримання списку ботів, необхідних для завантаження мережі P2P.

Adwind ботнет використовував цей крок, щоб перевірити стан передплати програмного забезпечення в Botmaster . Один з найбільш ранніх ботнетів, мережа Tribe Flood, відмовився від кроку реєстрації, і тому список ботів доводилося

вести вручну. Це було автоматизовано пізнішим втіленням Штахельдрахта, який поєднав вихідний код мережі племені племен та інших ботнетів.

4) Підсумок розповсюдження: Першим кроком у життєвому циклі бота є розповсюдження. У цьому розділі представлені схеми зв'язку, необхідні для двох активних та пасивних типів поширення. Обидва починають із зараження потерпілого. Оскільки під час активного розмноження бот заражає жертву-господаря, цей тип розповсюдження має необов'язковий сканування та необов'язковий етап координації перед зараженням. Після успішного зараження обидва типи розповсюдження закінчуються необов'язковою реєстрацією бота за допомогою ботнету.

Підсумок різних типів розповсюдження та реєстрації, що використовуються в ботнетах, представлений у таблиці III. Ця таблиця перераховує для кожного проаналізованого ботнету, чи відбувається поширення активно, пасивно чи за допомогою обох методів. Крім того, перераховані необов'язкові кроки координації, сканування та реєстрації.

D. Виконання (робота)

Після розповсюдження бот виконує поставлені йому завдання. Під час цього етапу, що називається функціонуванням, виникають різні комунікаційні потреби залежно від конкретних цілей. Прикладом цих комунікаційних потреб є координація DDoS, щоб кожен бот атакував одну і ту ж ціль одночасно. Опис роботи DDoS-атак ви можете побачити у розділі IV-G. Інший приклад – боти для збору даних, яким потрібно повідомляти про зібрані дані назад до C&C або до ботмайстра.

Необхідна контрольна комунікація залежить від кількості переданих даних, напрямку потоку даних та схеми зв'язку. Оскільки точний обсяг та напрямок даних залежать від використовуваного протоколу, ця робота зосереджена на схемі зв'язку. Шаблон комунікації – це послідовність повідомлень, які повинні бути обмінені у правильному порядку, щоб досягти конкретного результату. Більш детальний опис можна знайти на початку цього розділу. Залежно від використовуваної схеми зв'язку, операцію можна класифікувати як і)

завантаження даних, ii) завантаження даних, iii) прямий проксі, iv) зворотний проксі і v) інструкція.

Завантаження даних використовується для завантаження створених або зібраних даних з ботів, тоді як завантаження даних може використовуватися для зберігання даних (наприклад, додаткового зловмисного програмного забезпечення) на ботах. Умови завантаження та завантаження розглядаються з точки зору бота.

Проксі-сервер використовує ботнет, щоб приховати реальне походження зв'язку. Це можна використовувати з міркувань конфіденційності або посилити трафік.

Зворотний проксі – сервер здатний ретрансляції запитів, що надходять з Інтернету до конкретних походження хоста. За допомогою цієї методики можна створити CDN для приховування справжнього походження та додати стійкість проти збою або збою хоста.

Інструкція є шаблон зв'язку, який вимагає найменшу кількість передачі даних для завершення, так як тільки команда повинна бути відправлена по мережі. Він використовується в сценаріях, коли боти виконують конкретні завдання на хост-комп'ютері (наприклад, змінюють конфігурацію або видаляють певний файл).

У нижченаведеному списку детально обговорюються різні варіанти комунікаційних моделей.

Один ботнет може мати більше одного режиму роботи. Є кілька ботнетів, серед яких Blackenergy, Duqu 2.0 та Regin, які побудовані модульно. Ті ботнети може бути попередньо налаштовано для виконання певних завдань. Крім того, вони дозволяють динамічно змінювати завдання, на які здатний ботнет.

1) Завантаження даних: Ботнети, що належать до категорій збору інформації та розподілених обчислень, визначених, використовують схему зв'язку даних для завантаження даних. Одним із прикладів завдання ботнету, що належить до цієї категорії, є завантаження різних типів інформації (наприклад, облікові дані для входу, системна інформація, довільні файли) із заражених комп'ютерів. Ще одне

можливе завдання – використовувати ботнет для обчислення даних, а потім завантажувати результати (наприклад, видобуток біткойна, злом пароля).

Завантаження даних починається з необов'язкової координації (крок (A) на рис. 10). Під час цього кроку C&C надсилає команди ботам із конкретними інструкціями або даними, необхідними для обчислення. Оскільки завдання з завантаження даних може відбуватися автоматично, наприклад, безпосередньо після розповсюдження, крок координації не є обов'язковим.

Далі, запитувані дані збираються з комп'ютерів або запитувані обчислення виконуються на етапі (B). Запитані дані можуть включати, наприклад, адреси електронної пошти, облікові дані для входу, ліцензійні ключі програмного забезпечення або дані журналу клавіатури.

Останній крок (C) схеми зв'язку даних для завантаження даних – це звітування про результати C&C. Оскільки цей крок залежить від тривалості призначеного завдання, він може статися пізніше.

Ботнет, що використовує завантаження даних без узгодження, – Waledac. Після зараження він автоматично здійснює пошук зараженого комп’ютера за адресами електронної пошти та обліковими записами для входу, які після відкриття надсилаються до C&C . Ще один приклад беззгодженого завантаження даних – бонет-мережа Майнера. Після розповсюдження він вимірює продуктивність мережі, збирає системну інформацію та автоматично надсилає ці дані в C&C . Ботнети, які використовують координацію для завантаження даних, включають Blackenergy та Duqu 2.0, які здатні завантажувати різні види інформації з заражених комп'ютерів. Два ботнети, які використовують для обчислення заражені комп'ютери, це Майнер та ранні версії Zeroaccess . І те й інше включають добувач монет, який використовує обчислювальну силу ботів, щоб збирати гроші від імені ботмайстра.

Повний перелік ботнетів, які аналізуються цією роботою, разом із запитуваними завданнями та факультативними координуючими заходами зведений у таблиці IV. Ця таблиця включає в себе зібрані дані, такі як логіни з

файлів на жорсткому диску і мережевий трафік, інформації про систему, пристрої та мережеві та електронні адреси.

Крім того, захоплені дані з мережі, аудіо або відео пристроїв, а також screen- і клавіатур -sniffers перераховані. Крім того, перераховані біткойн-обчислювальні ботнети та ботнети, які дозволяють завантажувати загальні дані. Ботнетів використання координації позначаються C і ботнетів, які автоматично завантажувати дані з N.

2) Завантаження даних: Цей режим роботи потрібен для зберігання даних на ботах. Ці дані можуть бути довільними файлами, додатковим програмним забезпеченням (поширення зловмисного програмного забезпечення в) або бінарним оновленням ботнету. Довільні файли можуть включати незаконний вміст, який може бути використаний для дискредитації жертви. Крім того, цей механізм можна використовувати для монетизації ботнету, пропонуючи ботів як потенційну базу для інсталяції іншим розробникам шкідливих програм.

Таблиця 2.3 – Сумарна інформація по завантаження даних ботами

	Collects logins from		Collects information about			Captured events and data				Comp.		
Botnet	File	Net	Sys	Dev	Net	Traffic	AV[a]	Screen	Keyb.	Bitcoin	EmailA.	Generic
Adwind	C	–	N	–	–	–	C	C	C/N[b]	–	–	C
Blackenergy	C	C	C	C	–	C	–	C	C	–	–	C
Conficker	–	–	–	–	–	–	–	–	–	–	–	–
Duqu 2.0	C	C	C	C	C	–	–	C	C	–	–	C
Miner	–	N	N	–	N	–	–	–	–	C	–	C
Phatbot	C	C	C	–	C	C	–	–	C	–	C	–
Regin	C	C	C	C	C	C	–	C	C	–	–	C
Rustock	–	–	N	–	–	–	–	–	–	–	–	–
Sality	N	–	–	–	–	–	–	–	–	–	–	–
Sinit	–	–	–	–	–	–	–	–	–	–	–	–
Slapper	–	–	–	–	–	–	–	–	–	–	–	–
Storm	N	–	–	–	–	–	–	–	–	–	N	–
Stuxnet	–	–	N	–	–	–	–	–	–	–	–	–
TFN[c]	–	–	–	–	–	–	–	–	–	–	–	–
Trinoo	–	–	–	–	–	–	–	–	–	–	–	–
Waledac	N	–	N	–	N	–	–	–	–	–	N	–
Zeroaccess	N	N	N	–	–	N	–	–	–	C[d]	–	–
Zeus	C/N	C/N	N	–	–	C/N	–	C/N	C/N	–	–	–

C – Координація. N – Немає координації. – Не показано

Додавання можливості для самостійного оновлення дозволяє ботмайстру адаптувати ботнет до нових потреб або захистити від майбутніх загроз. Це

самооновлення додає вектор атаки до ботнету, який можна використовувати для захоплення ботнету або заміни його на інший. Тому цю процедуру потрібно забезпечити.

Схема зв'язку даних для завантаження даних починається з кроку координації (A). Під час координації C&C вказує боту, який файл слід завантажити та де його знайти. Це спрощує протокол C&C, оскільки не потрібні можливості передачі файлів і розширює можливості ботнету, що дозволяє завантажувати довільні файли. Бінарні файли ботів можуть містити додатковий файл конфігурації, який запускає завантаження даних відразу після розповсюдження. Тому крок координації необов’язковий.

Після узгодження дані потрібно завантажити на бот на етапі (B). Ці додаткові дані вже можуть бути включені у біт-бінар, який використовується під час поширення, або бути частиною команди координації.

Останній крок (C) – це збереження завантажених файлів на комп’ютері або встановлення завантаженого програмного забезпечення. Незважаючи на те, що немає необхідності в повідомленні C&C, додатково встановлене програмне забезпечення або оновлення клієнта ботнету можуть спричинити додатковий трафік після цього кроку. Одним із прикладів цього додаткового повідомлення може бути реєстрація під час цього поширення (див. Розділ VI-C).

Встановлення додаткового зловмисного програмного забезпечення було однією з головних цілей ботнетів Conficker, Sality, Sinit та Waledac . Miner ботнет може включати в себе додаткове програмне забезпечення в довічним бота, використовуваного в процесі поширення, і, отже, крок координація не використовується кожен раз .

3) Forward Proxy: Ботнети, які використовуються для розповсюдження спаму або для виконання DDoS-атак, потребують можливості підключення до інших хостів від імені ботмайстра (див. Розділ IV-G). Це імітує ботнет, який використовується як прямий проксі. Проксі-проксі – це проміжний сервер, який може здійснювати з'єднання з певними хостами від імені хоста, що запитує. Таким чином, прямі проксі-сервери можуть використовуватися для збереження

анонімності запитуючого хоста, для тунельних з'єднань через інший протокол або для здійснення з'єднань, які були б інакше заборонені або неможливі.

Ботні мережі можуть реалізовувати або двонаправлений, або однонаправлений проксі-сервер.

Двонаправлена використовується, якщо результату запиту потрібен роботодавець або третя сторона. Наприклад, деякі ботнети включають проксі-сервіс Socket Secure (SOCKS), який дозволяє використовувати ботнет як послугу анонімізації. Проксі-сервер SOCKS – це загальний проксі-сервер, який можна використовувати для пересилання довільних TCP та UDP-з'єднань. Після початкової фази налаштування проксі-сервер SOCKS прозоро передає необроблені дані, що дозволяє використовувати проксі навіть із програмами, які не підтримують проксі-сервери .

Інший приклад для двостороннього наближення – це проходження брандмауерів або NAT. Цього можна досягти, заразивши брандмауер або NAT-пристрій ботом, який, в свою чергу, виступає мостом у локальну мережу.

Для підвищення скритності ботнету, вперед проксінг може також використовуватися для приховування протоколів в одному або декількох інших протоколів, тобто до протоколу наймають інкапсуляції. При такому підході одна схема зв'язку C і C може поширюватися на декілька протоколів на цьому шляху. Ця методика збільшує складність мережевих детекторів ботнетів, оскільки детектори повинні бути здатними розуміти всі реалізовані протоколи та реконструювати вихідну комунікацію.

Однонаправлене проксі-просування може використовуватися для перевантаження однієї послуги запитами. Цього можна досягти, доручивши ботам постійно надсилати запити до мережевих служб. Оскільки C&C не потребує контролю над кожним повідомленням, а також відповідей, що цікавлять, достатньо проінструктувати ботів про ціль та тип зв'язку.

Інший спосіб використання однонаправленого проксі-проксі – це використовувати його для розповсюдженого спаму електронної пошти. Під час координації шаблону спаму і списки адресатів відправлення надсилаються боту.

Список адрес призначення може походити від самого ботнету, де він був отриманий за допомогою завантаження даних. Після кроку координації електронні листи розповсюджуються на сервери електронної пошти.

Варіант того ж методу може бути використаний для здійснення шахрайства з натисканням (див. Також розділ IV-G). Як і спам електронної пошти, цільові URL-адреси поширюються на боти під час кроку координації, які в свою чергу надсилають запити після цього.

Ботнети, що підтримують ці завдання, є частиною категорій, що називаються кібер-шахрайством, небажаним маркетингом та порушенням роботи з мережевими сервісами.

РОЗІДЛ 3
ВИЯВЛЕННЯ БОТНЕТІВ
З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ

Пропонується фреймворк з використанням нейронних мереж в модулі попередньої обробки Байєсової регуляризації. Байєсівська регуляризація допомагає досягти кращого узагальнення набору даних, тим самим дозволяє виявити активність ботнетів навіть тих ботів, які ніколи не використовувались у навчанні нейронної мережі. Це означає, що такий фреймворк підходить для зменшення нових і небачених ботнетів у прямому трафіку мережі, як видно з результатів цього дослідження. Завдяки узагальненню, наданому Байєсовою регуляризацією, автори досягли успіху у виявленні активності нетренованих шкідливих ботів з точністю 99,2%. Потім ця модель була інтегрована в API Java, який може використовуватися як модуль попереднього процесора для будь-якої системи виявлення вторгнень для виявлення в реальному часі будь-якого трафіку ботнету в мережі.

Більшість попередніх науково-дослідних робіт була зосереджена на виявленні конкретної діяльності ботнету. Не було зареєстровано таких методів у виявленні ботів, характеристики трафіку яких не використовуються при навчанні алгоритму машинного навчання. Зрозуміло, що видно, що використання базового підходу до машинного навчання є кращим у виявленні зловмисного трафіку в порівнянні з традиційним підходом, заснованим на підписах, оскільки боти-майстри час від часу переробляють ботів, а функціональність, поведінка тощо бот-мережі змінюються досить істотно з кожною новою версією бота. Крім того, в попередніх роботах не було багато досліджень щодо розгортання модуля виявлення в сценарії реального часу для моніторингу та пом’якшення діяльності ботнету в мережі.

3.1 Концепція методики

Алгоритм машинного навчання вимагає відповідних "функцій" як вхідних даних для навчання моделей. Для цього дослідження зразки деяких ботів P2P були розміщені на тестовому шарі (як описано в наступному розділі) та отримані мережеві файли слідів (pcaps). Ці файли слідів потім використовувались для вилучення функцій за допомогою інструмента з відкритим кодом Netmate (Netmate 2011) . Netmate надає свій вихід у вигляді "потоків" (визначених кортежем <Джерело IP, Порядок джерела, IP-адреса призначення, Порт призначення, протокол>) та витягує більше сорока особливостей, що підлягають збиранню на кожен потік.

Таблиця 3.1 – Опис використовуваних функцій

Властивість	**Опис функції**
Srcip	IP-адреса джерела (рядок)
Srcport	Номер вихідного порту
Dstip	IP адреса призначення (рядок)
Dstport	Номер порту призначення
proto	Протокол (тобто TCP = 6, UDP = 17)
total_fpackets	Всього пакетів у напрямку вперед
total_fvolume	Всього байтів у напрямку вперед
total_bpackets	Загальна кількість пакетів у зворотному напрямку
total_bvolume	Всього байтів у зворотному напрямку
min_fpktl	Розмір найменшого пакета, що надсилається у напрямку вперед (у байтах)
mean_fpktl	Середній розмір пакетів, що надсилаються у напрямку вперед (у байтах)
max_fpktl	Розмір найбільшого пакета, відправленого в напрямку вперед (у байтах)
std_fpktl	Стандартне відхилення від середнього пакету, відправленого у напрямку вперед (у байтах)
min_bpktl	Розмір найменшого пакета, відправленого у зворотному напрямку (у байтах)
mean_bpktl	Середній розмір пакетів, відправлених у зворотному напрямку (у байтах)

Продовження таблиці 3.1

max_bpktl	Розмір пакету найбільшого жесту, відправленого у зворотному напрямку (у байтах)
std_bpktl	Стандартне відхилення від середнього пакету, відправленого у зворотному напрямку (у байтах)
min_fiat	Мінімальна кількість часу між двома пакетами, відправленими в напрямку вперед (в мікросекундах)
mean_fiat	Середня кількість часу між двома пакетами, відправленими в напрямку вперед (в мікросекундах)
max_fiat	Максимальна кількість часу між двома пакетами, відправленими в напрямку вперед (в мікросекундах)
std_fiat	Стандартне відхилення від середньої кількості часу між двома пакетами, що надсилаються у напрямку вперед (у мікросекундах)
min_biat	Мінімальна кількість часу між двома пакетами, відправленими у зворотному напрямку (у мікросекундах)
mean_biat	Середня кількість часу між двома пакетами, відправленими у зворотному напрямку (у мікросекундах)
max_biat	Максимальна кількість часу між двома пакетами, відправленими у зворотному напрямку (у мікросекундах)
std_biat	Стандартне відхилення від середньої кількості часу між двома пакетами, що надсилаються у зворотному напрямку (у мікросекундах)
Тривалість	Тривалість потоку (в мікросекундах)
min_active	Мінімальна кількість часу, що цей потік був активний до виходу з режиму очікування (у мікросекундах)
mean_active	Середня кількість часу, що цей потік був активним до простою (в мікросекундах)
max_active	Максимальна кількість часу, що цей потік був активний до простою (у мікросекундах)
std_active	Стандартне відхилення від середньої кількості часу, який протікав активний перед тимчасовим холостим режимом (у мікросекундах)
min_idle	Мінімальний час, коли потік простоював, перш ніж активуватися (у мікросекундах)
mean_idle	Середній час, коли потік простоював, перш ніж стати активним (у мікросекундах)
max_idle	Максимальний час, коли потік простояв, перш ніж активуватися (у мікросекундах)
std_idle	Стандартне відхилення від середнього часу потік простоював, перш ніж стати активним (у мікросекундах)
sflow_fpackets	Середня кількість пакетів в подачі потоку в напрямку вперед
sflow_fbytes	Середня кількість байтів у підтоці в прямому напрямку

Продовження таблиці 3.1

sflow_bpackets	Середня кількість пакетів в подачі потоку в зворотному напрямку
sflow_bbytes	Середня кількість пакетів в подачі потоку в зворотному напрямку
fpsh_cnt	Кількість встановлених прапор PSH у пакетах, що рухаються у напрямку вперед (0 для UDP)
bpsh_cnt	Кількість встановлених прапор PSH у пакетах, що рухаються у зворотному напрямку (0 для UDP)
furg_cnt	Кількість встановлених прапор URG у пакетах, що рухаються у напрямку вперед (0 для UDP)
burg_cnt	Кількість встановлених прапор URG у відкладеннях пакетів у зворотному напрямку (0 для UDP)
total_fhlen	Загальна кількість байтів, що використовуються для заголовків у напрямку вперед.
total_bhlen	Загальна кількість байтів, що використовуються для заголовків у зворотному напрямку.

Для цих експериментів автори видалили перші чотири функції (Source IP, Source source, Destination IP, Destination port), оскільки вони повністю залежать від конфігурації мережі, в якій розгорнуті боти. Оцінка атрибутів набору інформації отримана за допомогою алгоритму Ranke r для того, щоб знайти найвпливовіші характеристики всього набору функцій. Цей метод оцінює значення атрибута, вимірюючи приріст інформації стосовно класу, де посилення інформації описується наступним рівнянням:

Приріст інформації (клас, атрибут) = H (клас) – H (клас | атрибут). (3.1)

Далі були перші 15 у списку, а їхній інформаційний приріст показаний у першій колонці.

3.2 Байєсова регульована нейромережа

Штучна нейронна мережа – дуже корисний інструмент для машинного навчання, який застосовується в декількох сценаріях. Він застосовувався до багатьох випадків, таких як приклад вимови тексту, підготовлений нейронною мережею розповсюдження спини (Franco, et al., 1997) . Він також знайшов багато

застосувань у галузі розпізнавання візерунків (Carpenter and Grossberg 1988) . При прогнозному моделюванні застосування Штучних нейронних мереж має перевагу в тому, щоб вміти фіксувати дуже складні відносини.

Таблиця 3.2 – Особливості, вибрані за допомогою алгоритму ранжирування інформації

Ранг	**Особливість**
0,814	total_bhlen
0,81	std_bpktl
0,7886	fpsh_cnt
0,7751	total_fhlen
0,7654	bpsh_cnt
0,7568	min_biat
0,7438	min_fiat
0,7182	max_bpktl
0,7104	sflow_fbytes
0,6802	sflow_fpackets
0,6761	mean_fiat
0,6625	total_bpackets
0,6502	max_fiat
0,6459	mean_biat
0,6427	max_fpktl

Нейромережева архітектура, A, містить специфікацію кількості шарів, кількості одиниць у кожному шарі, типу функції активації, що виконується кожним блоком, та наявних з'єднань між одиницями. Значення для ваги, w, присвоюється з'єднанням в мережі, зважений вхід, *x*, сума відображається на *y* (3.2) (x; w, A), передбачуване значення виходу. Відстань передбачуваного значення до навчального набору вимірюється деякою функцією помилок. Зазвичай прийнято вважати помилку для всього набору даних

$$E(D, w, A) = \sum_{m=0}^{N} \frac{[y1(x^m; w; A) - y^m]^2}{2} \quad (3.2)$$

Тут E – функція помилок, яку часто називають середнім квадратом помилок (MSE), *y*=1 – прогнозований вихід, x – вхід, m – екземпляр даних. Тут y представляє вихідні дані, а N – загальний набір даних.

У літературі зауважено, що нейронні мережі, що розповсюджуються назад, використовувані для тренування моделі нейронної мережі, дають високу надійність. Алгоритм навчання зворотного розповсюдження використовує методику градієнтного пошуку, щоб мінімізувати середню квадратичну помилку виходу мережі.

Параметри мереж зворотного поширення зазвичай встановлюються методом проб і помилок. Зарезервовані дані тесту використовуються для оцінки його здатності до узагальнення (або перехресної перевірки). Ці параметри змінюють ефективну модель навчання, наприклад, кількість прихованих одиниць, терміни зменшення ваги тощо.

Тренування мережі полягає у пошуку набору ваг, w, який дає оптимальну карту між навчальним набором та передбачуваним набором. Очікується, що вивчені ваги добре підходять до нових прикладів. Відтворення на зворотному розмноженні вчиться шляхом градієнтного спуску для оптимізації функції помилок. Також можуть бути використані більш ефективні методи оптимізації, такі як сполучені градієнти або змінні метричні методи. На малюнку 2 показаний метод вибору оптимального числа о прихованого нейрона з використанням коефіцієнтів кореляції на кожному етапі.

Однак існують певні недоліки використання зворотних мереж поширення:

1. Він може перевищувати задані дані.

2. Для правильного відображення взаємозв'язку між входом і виходом потрібні великі набори даних .

Як було пояснено вище, видно, що хоча нейронні мережі зворотного поширення є дуже надійними, вони часто страждають від проблеми над пристосуванням експериментальних даних. Проблема надмірної підгонки або погане узагальнення даного набору даних виникають, коли мережа перенавчається протягом навчального періоду. Результатом є «занадто добре навчена» модель, яка може не працювати на невидимих даних. Ця проблема відома як крайня проблема Оккама . Принцип полягає в тому, що надмірно складні моделі не повинні віддавати перевагу більш простим. Для вирішення

цього питання використовується метод "байєсівського висновку", який автоматично вирішує проблему бритви Оккама.

Відповідно до основ байєсівського аналізу, правдоподібність літернативної гіпотези представлена ймовірностями, а висновок проводиться шляхом оцінки цих ймовірностей. Таким чином, використовуючи байєсівську теорію ймовірностей, можна автоматично зробити висновок про гнучкість моделі, обґрунтованої даними. Модель с а оцінюється за допомогою простого правила Байєса, заданого, як

$$P\left(H_i/D\right) = \frac{P(H_i)P\left(D/H_i\right)}{P(D)} \tag{3.3}$$

Де знаменник P (D) – це нормалізуюча константа, яка змушує остаточне переконання скласти до 1.

Байєсівський підхід до регуляризації мінімізує проблему надмірного пристосування, враховуючи « корисність придатності», а також мережеву архітектуру. Отже, в цій роботі застосовується байєсівський підхід до регуляризації для кращої придатності даних. Пакет інструментів нейронної мережі програмного пакету MATLAB використовується для навчання та тестування даних. Алгоритм Левенберга-Маркарда з байєсівською функцією регуляризації був використаний для навчання мережі. Отримана архітектура мережі, для якої мережа має мінімальну суму квадратів помилок (SSE), а також має кращу узагальнюючу здатність до іонів.

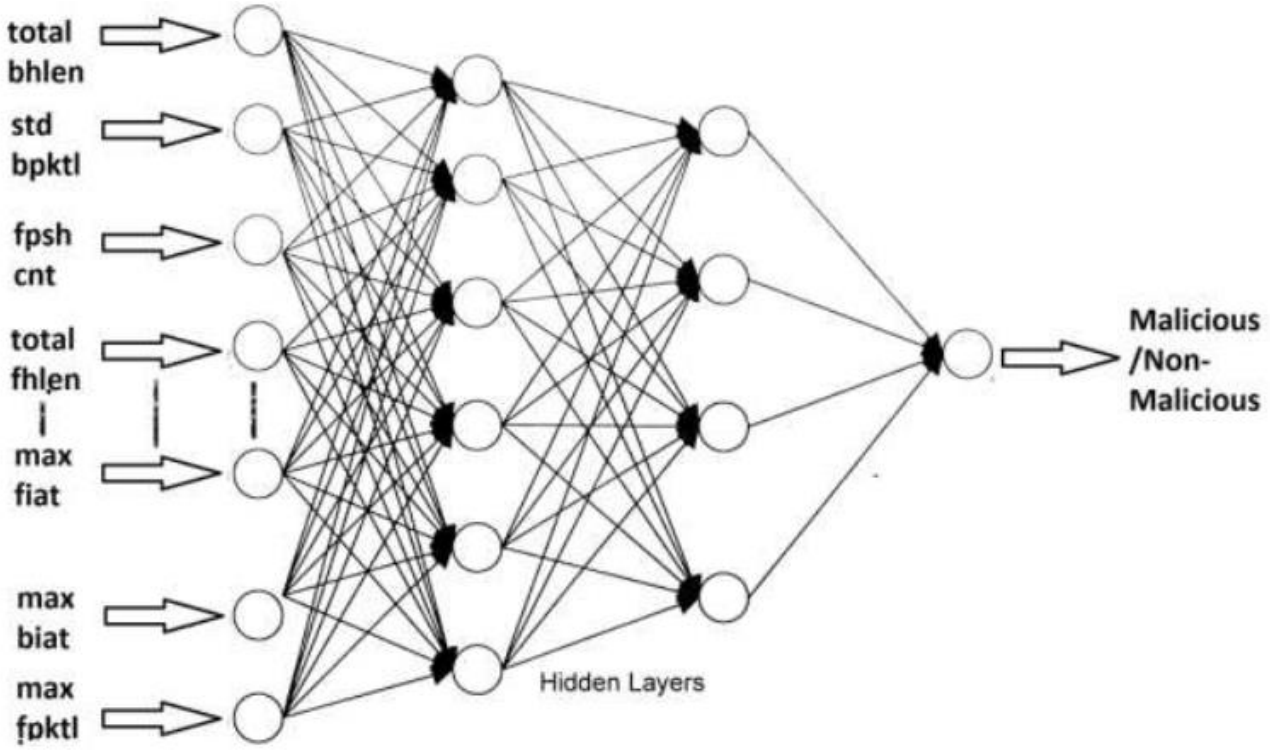

Рисунок 3.1 – Архітектура нейронної мережі для виявлення ботнету

П'ятнадцять функцій – це вхід, а вихід – це мітка класу, що вказує на характер потоку. Спочатку вхід нормалізується від 0,05 до 0,95, використовуючи наступне рівняння для функції передачі для активації, оскільки сигмоподібна функція сірого кольору використовується для відображення зважених входів на вихід.

$$x\,n = 0{,}05 + 0{,}9 * (\,x - x\,min\,) / (\,x\,max - x\,min\,) \qquad (3.4)$$

Де x_{min} і x_{max} – мінімальні та максимальні значення x і x_n – це нормалізовані дані відповідного x. Після виявлення найкращої навчальної мережі всі перетворені дані повертаються до початкового значення, використовуючи наступне рівняння:

$$x = x_{min} + (\,x_n - 0{,}05) * (\,x_{max} - x_{min}\,) / 0{,}9. \qquad (3.5)$$

3.3 Використовуваний алгоритм та псевдокод

Весь код розділений на два модулі. Один модуль – метод getConversations, а другий – метод testtshark. getConversations читає з вказаних файлів pcap і викликає Netmate для отримання статистики потоку з файлу pcap, який використовується для побудови моделі BR-ANN. Testtshark – це модуль у режимі реального часу, який використовує jnetpcap для витягування тих самих функцій із прямого трафіку та подає потоки в Matlab-Weka-API для класифікації.

```
getConversations()
_____________________________________________________________________
1. BEGIN
2. invoke Netmate
      a. packet properties are extracted and buffered
      b. packets are characterised as flows and their 44 statistics are
         calculated
      c. statistics are written into csv file
3. csv file converted to arff file by adding Attribute and Data headers
4. Instances in the arff file divided into 90% training data and 10%
   testing data; sent to the MATLAB BR-ANN code as input to train the
   model
5. Trained model is imported into Weka using MATLAB-WEKA API
6. END
```

Рисунок 3.2 – Псевдокод функції getConversations()

3.4 Формування набору даних

Тестовий зразок для цієї дослідницької роботи складається з окремої мережі на системі Linux. Системи були підключені до перемикача доступу для формування окремої мережі. На вершині кожної з цих фізичних машин автори запускали віртуальні машини з операційною системою Windows XP. На цих віртуальних машинах були розгорнуті зразки Kelihos-Hlux, Zeus, Waledac. Зразки цих шкідливих програм були отримані з (contagiodump.blogspot.in та openmalware.com). Мережева активність цих зразків шкідливого ПЗ відстежувалася протягом 48 годин кожна, а активність фіксувала Wireshark. Після збирання пакетних слідів мережевої активності кожної з шкідливих

програм, безпосередньо від стадії зараження до стадії атаки, файли pcap зберігалися на сервері баз даних для подальшого аналізу.

testtshark()

```
1. BEGIN
2. packet capture started on the wire using tshark
3. jnetPcap invoked
      a. get packet properties of every packet
      b. send the packets to a buffer
      c. group on <source_ip, destination_ip, source_port,
         destination_port, protocol> to form flows.
      d. Calculate flow statistics for each of the keys in the above
         step
      e. Write the flow statistics to another buffer converting them
         into instances of test data
4. Send the Instances to Weka API to test against the loaded training
   model
      a. If instance is malicious
            i.  Flag the class label to be malicious
           ii.  Make a note of the time stamp
          iii.  Search for the packet containing this time stamp and
                write it to a separate pcap file.
      b. If instance is non-malicious
            i.  Ignore and continue
5. Send the pcaps classified as malicious to an IDS/IPS/Firewall for
   further actions
6. END
```

Рисунок 3.3 – Псевдкод функції testtshark()

"Зловмисний набір даних" містив загалом 55 824 мінімуми. Щоб досягти належної класифікації, автори збирали доброякісний трафік, що складається з трафіку з додатків p2p, передачі ftp, сеансів telnet, потокової передачі відео та мобільних оновлень, а потоки витягувались так само, як і для зловмисного трафіку. Весь набір даних був об'єднаний та наданий алгоритму машинного навчання для генерації моделі.

3.5 Виявлення Botnet в реальному часі

Для виявлення в реальному часі активності ботнету в мережі модуль розгорнули в системі, яка отримувала дзеркальний трафік з усієї мережі, як показано на рисунку 3.4. Цей модуль використовує tshark для зчитування пакетів

і зберігає їх у форматі libpcap шматками по 200 Мб кожен. Збережені пакети захоплення перетворюються в бесіди (або потоки). Кожен потік – це екземпляр, який слід контролювати за шкідливою активністю .

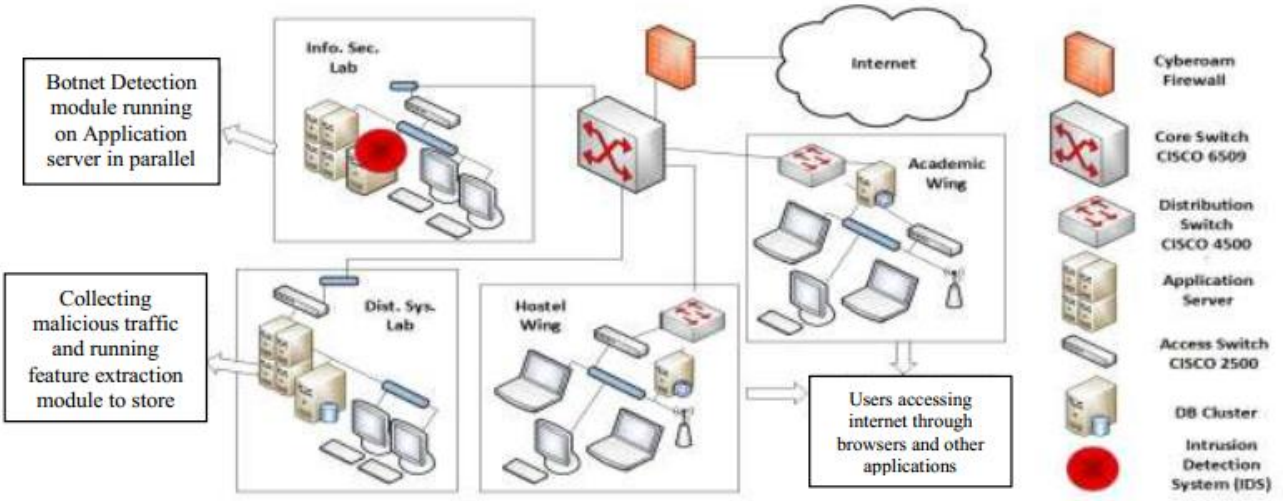

Рисунок 3.4 – Тестовий шар, який використовується для розгортання модуля виявлення ботнет P2P

Модель Байєсової нейронної мережі, навчання якої було описано в попередньому розділі з методології, було успадковано за допомогою API інтерфейсу Matlab-Weka, щоб перевірити потоки проти тренованої моделі в режимі реального часу. Відповідно до класифікації, виконаної навченою моделлю, в окремий файл pcap зберігаються лише ті потоки (і, отже, лише ті пакети), які позначені зловмисним алгоритмом. Все це робиться в режимі реального часу, тим самим вказуючи на активність ботнету в теперішньому стані мережі. Цей модуль, показаний на рисунку 6, може слугувати хорошим модулем попередньої обробки для будь-якого IDS / IPS для виявлення трафіку Botnet і позначає попередження про той самий. Робочий знімок моделі показаний на рисунку 7, що працює на IDE затемнення.

Точна послідовність кроків, що використовуються в цьому модулі виявлення ботнет P2P в реальному часі, показана на рисунку 3.5.

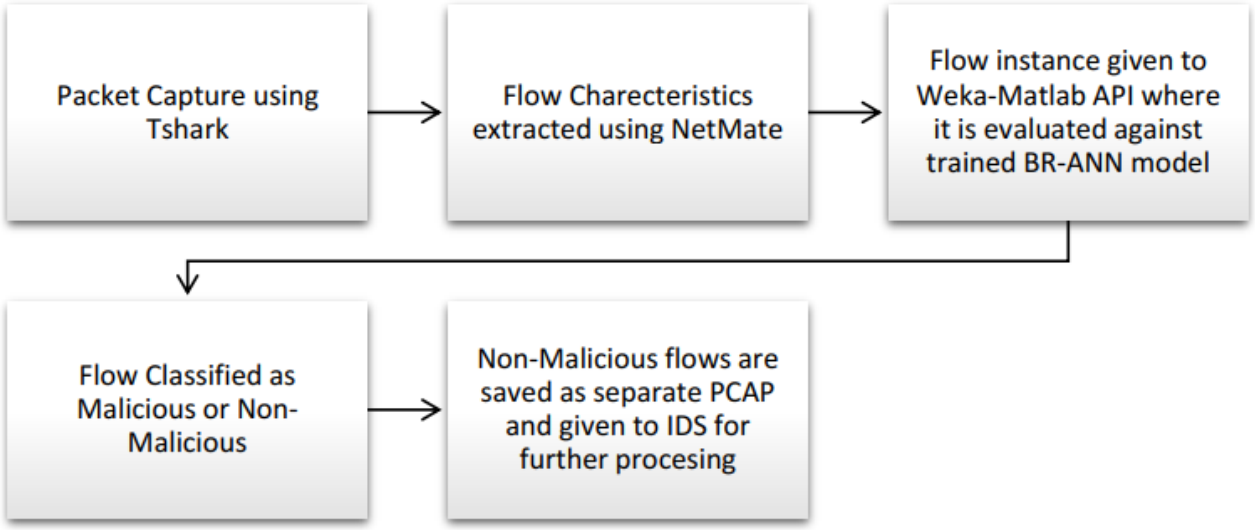

Рисунок 3.5 – Модуль виявлення в режимі реального часу

Кореляція між змінними величинами X та Y вимірюється коефіцієнтом Пірсон, який приймає значення в проміжку +/-1. Він задається формулою

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})\,(Y_i - \bar{\bar{Y}})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

Ідеальне передбачення повинно генерувати пряму лінію, що проходить через початок координат, під кутом 45 градусів, оскільки вісь X і вісь Y представляють експериментальні та прогнозовані значення кожного з використаних методів. Видно, що коефіцієнт кореляції кореляції дуже близький до 1. Це свідчить про те, що корисність моделі, породженої Байєсовим регульованим ANN, близька до точного прогнозу. 90% даних розглядаються як дані тренувань, а решта 10% – дані тестування для перевірки відповідності створеної моделі. Кореляція, отримана тренувальною мережею, дуже висока і дорівнює 0,9931.

Під час тестування на ботах, які не використовувались у навчальному наборі, з зразками, отриманими від CAIDA (мережевий телескоп CAIDA UCSD

"Thre e Days Of Conficker" – 21 грудня 2009 р.) Та ISOT (Saad та ін. 19-21 липня 2011 р.)), модель BR-ANN дала кореляцію 0,9902.

Передбачувана кількість випадків нешкідливих та зловмисних дій дуже близька до експериментальної кількості випадків, і тому ці два встановлюють роботу модуля BR-ANN. Діаграма розсіювання зображує характеристики точності відкликання моделі, де згадати і s на осі x та точність на y. Отже, прогнози Нейронної мережі мають задовільне відображення експериментальних даних. Це показує, що корисність придатності моделі, породженої Байєсовською ANN, є статистично задовільною та точною.

РОЗДІЛ 4
СПЕЦІАЛЬНА ЧАСТИНА

4.1 Кібер-атаки на основі ботнетів

У цьому місяці ми дізнаємося про ще один небезпечний тип атак, який панує навіть у сучасних комп'ютерних системах. Напади, про які йшлося в попередніх статтях, ґрунтувалися на використанні технічних уразливостей для крадіжки даних або знищення даних.

Атака ботнету є вдосконаленим типом, завдяки чому атакована комп'ютерна система стає сама атакою. Це викликає серйозні загрози в світі IT, про які слід знати кожному системному адміністратору та старшому керівництву технологій. Ми також обговоримо, як системи FOSS повинні бути захищені такою атакою.

Ботнет.

Слово ботнет походить від фрази "мережа роботів". По суті це широко поширена колекція великої кількості заражених комп'ютерних систем. Кожна заражена система не містить частину програмної програми, яку називають "Bot".

Як показано на рисунку4.1, існує система Bot-Master, яка веде облік загальної кількості заражених машин та завдань, які вони повинні виконувати. Для ретельно хореографічних систем, яким потрібно оркеструвати між мільйонами таких систем, створюється ще один шар менеджерів ботів. Бот-менеджери виконують завдання приймати команди від ведучого, розповсюджувати ці команди на боти, а також повідомляти про кількість систем, заражених відповідно до її юрисдикції.

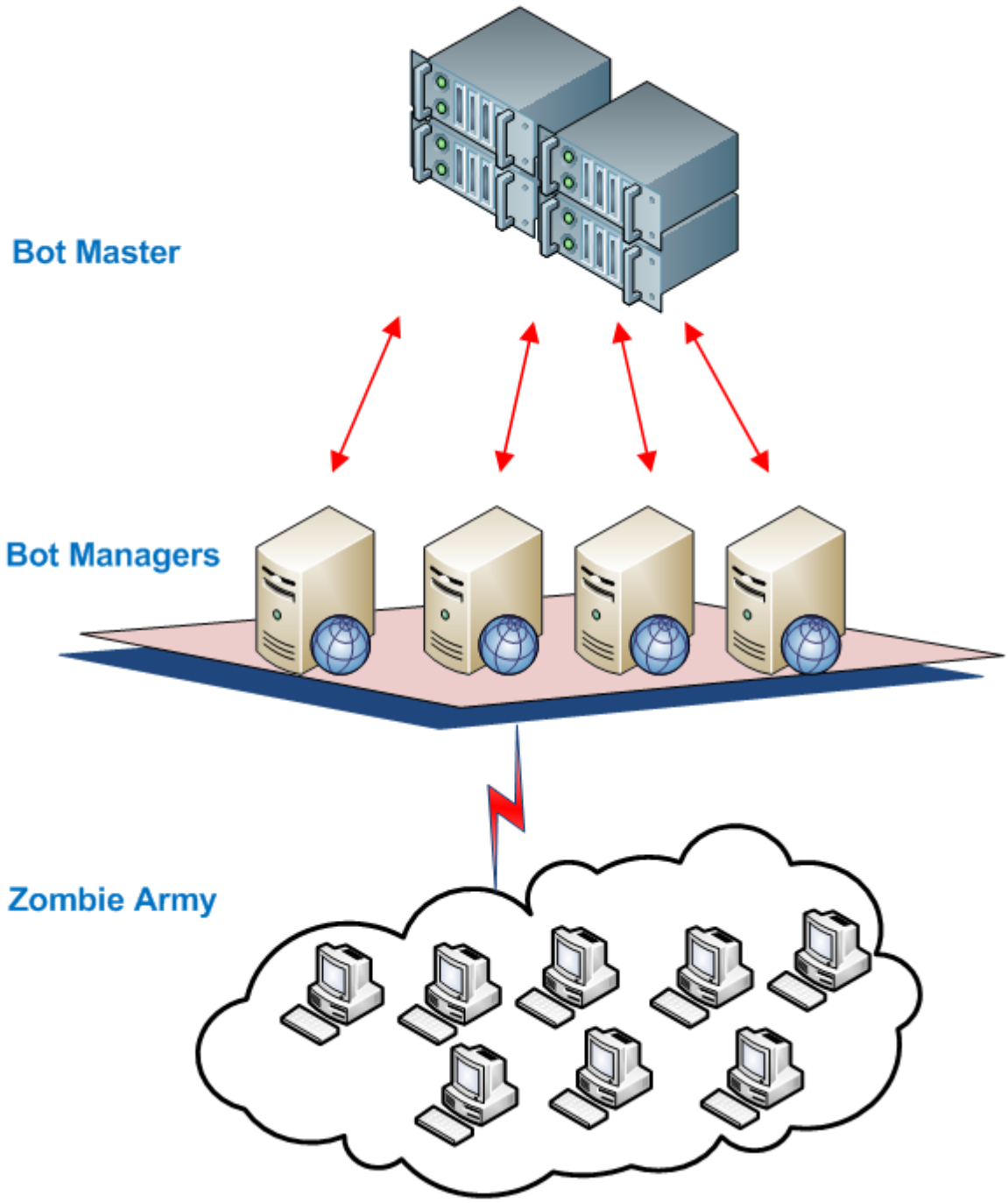

Рисунок 4.1 – Принцип атаки на основі бот-нетів

Також виявлено, що менеджери ботнетів надсилають оновлені програмні виправлення, щоб виправити помилки або покращити функціональність, дуже схожу на систему управління патчем безпеки. Майстер ботів керує хакером, який має злі наміри створити цю армію.

Однак оскільки хакер повинен ховатися від потрапляння, майстер-системи та програмне забезпечення, що працює на ньому, завжди працюють у режимі прихованості. У кількох сучасних атаках на ботнетів було виявлено, що ботмайстри делегують та обертають керівну роль між своїми бот-менеджерами, що робить його надзвичайно важким для виявлення. Крім того, ці зміни ролей виявили, що вони обертають свою власність залежно від країни присутності, щоб забезпечити величезні порушення в усьому світі.

Зазвичай ботнети розроблені для певної операційної системи, і якщо потрібно досягти більш широкого поширення, ботнети віддають перевагу веб-коду чи мові Java, щоб заразити всі можливі платформи операційної системи.

Тепер давайте поглянемо на внутрішні операції типового Бота. Як показано на малюнку 4.2., є 4 основні модулі ботнету. Командний модуль надсилає команди дочірнім ботнетам, тоді як модуль управління керує правами власності, щоб вирішити, хто повинен слухати кого. Модуль зараження несе важливу відповідальність за пошук непатч-серверів у мережі та зараження тих, хто має найновішу копію.

Модуль стелс – це по суті набір програмних програм, який виконує вирішальну роботу, наприклад відключення antiv irus; досягти кореневого доступу або доступу до ядра. Він також гарантує, що його власний слід на зараженій машині невидимий з точки зору запущених процесів та дискового простору, а також стежить за встановленням нового антивірусного програмного забезпечення. У деяких випадках вкрадений модуль і модуль управління працюють разом, щоб отримати найновіший патч себе від майстра або менеджера і плавно модернізувати себе. Деякі модулі прихованості також здатні стерти себе за допомогою механізму саморуйнування або відключення системи, щоб зірвати агресивні методи виявлення.

Спосіб взаємодії ботнетів з майстром або менеджером теж дуже цікавий. Всім ботам надається унікальний ідентифікаційний номер, який зазвичай є продуктом конфігурації та локалізації зараженої системи, але не обов'язково ір-адреси системи. Ведучий завжди має найновіший облік цих ідентифікаційних

номерів, що використовуються, і здатний обмежити поширення або розширити його.

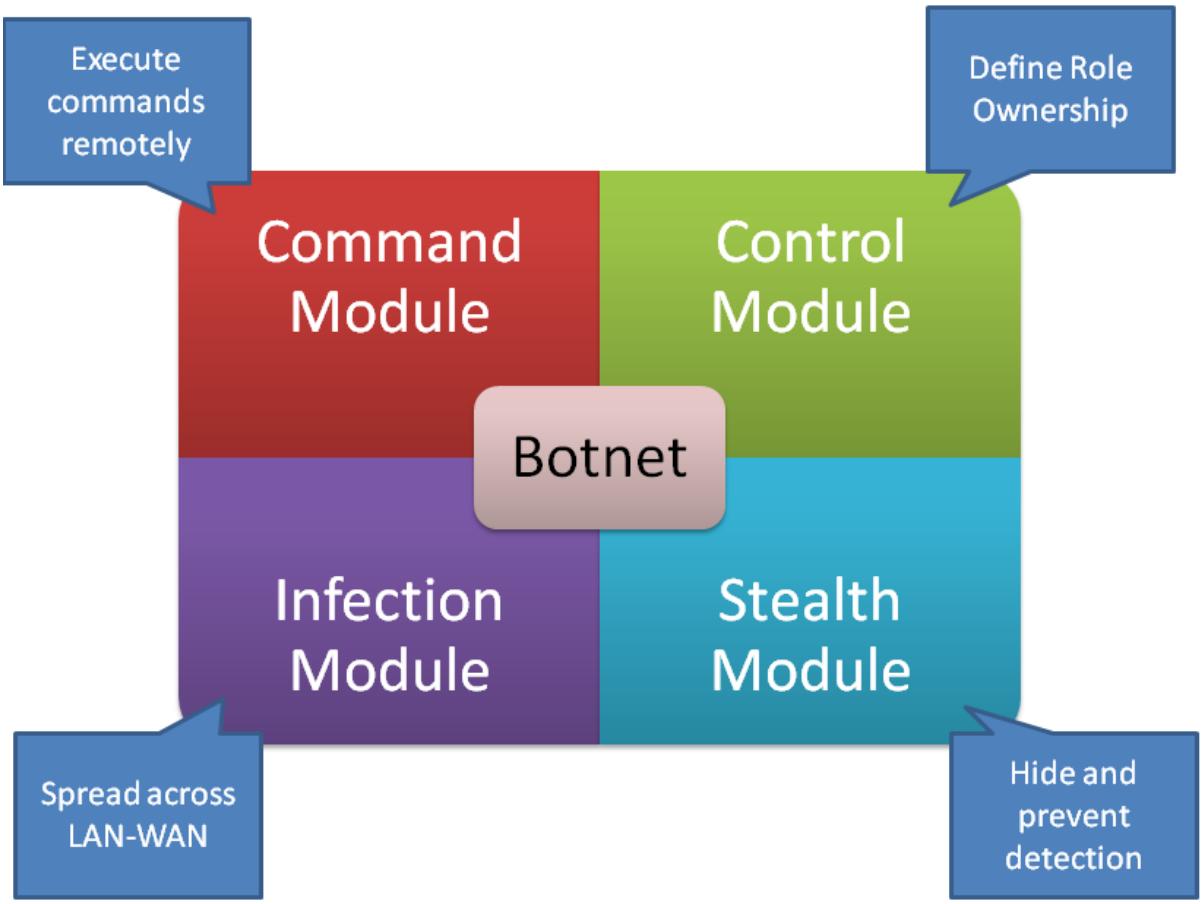

Рисунок 4.2 – Структура бот-нету

Боти використовують специфічний діапазон портів TCP; однак точний порт, що знаходиться у sed, підбирається випадковим чином. Завдання самого ботнету обов'язково повідомляти про майстер або менеджер про номер порту TCP, який він планує використовувати. Ця звітність відбувається при кожному перезавантаженні зараженої системи. У більшості випадків міжботові комунікації ba se-64 або md5 зашифровані, тоді як у деяких випадках використовується цифровий сертифікат, який підписується самостійно.

Основна мета введення ботнету в систему – створити армію заражених систем, яку також називають машинами зомбі. У нижченаведеній таблиці пояснюються різні типи ботнетів та цілі, що стоять за їх введення в мережу.

Загальною метою такої атаки є, в кінцевому рахунку, зірвати комп'ютерні системи або викрасти дані (таблиця 4.1).

Таблиця 4.1 – Основні типи бот-нетів

Тип Botnet	**Призначення**
DoSBot	DoS та розподілена DoS атаки за допомогою протоколу 3 до 7 рівня
SpamBot	Розсилка спаму електронною поштою
BrowseBot	Збір статистики перегляду користувачів та перехід у рекламну мережу
AdSenseBot	Те саме, що BrowseBot, але націлено на Google adsense
ChatBot	Збір стенограми чату, щоб знайти статистику чату користувачів
idBot	Збір паролів користувача
CCBot	Збір інформації про кредитні картки екранах
PollBot	Маніпулювати Інтернет-опитуваннями, присвячених для продуктам та послугам
BruteForceBot	Атака на веб-сайти по протоколу TCP на рівні додатків
NetBot	Атака на мережі за допомогою протоколів рівня 2 і 3

Оскільки ціла армія комп'ютерних зомбі перебуває в дії, але, на жаль, хакер може легко і швидко досягти успіху в своїй злій місії, це тому, що посадка ботнет-атаки – це завжди низький ризик і висока прибуток.

Поширення бот-нетів.

У перші дні Інтернету був розроблений фрагмент коду ботнету для програмного переходу через декілька веб-сайтів та подальшого збору та порівняння вмісту для створення значущих даних. Хоча цей метод формує серце пошукових систем на сьогодні, його в певний момент минулого хакери підробили на свою користь. Перш ніж ми поговоримо про те, як вводять ботнети,

давайте розберемося, чому це робиться. Щоб зробити веб-сайт відомим в пошуковій системі, необхідно обов'язково отримати багато веб-запитів.

Особливо це стосується веб-сайтів, які показують рекламу та заробляють гроші за кожен клік опублікованої реклами. Можна розповсюджувати ботнети по мережах, отримувати доступ до веб-сторінки та програмно клацати по одній або більше рекламних мереж на ній. Якщо така кампанія ретельно організована, важко зрозуміти, який клік ініціюється людиною законно, а який – кодом ботнету.

Веб-сайт, який розміщує людей, зазвичай хакер у такому випадку, може заробити багато грошей. В іншому різновиді, який називається фішботами, для досягнення подібних результатів можна розпочати електронну кампанію. Це говорить нам, що наслідки роботи ботнету виходять за рамки простої репутації чи втрати даних.

Ін'єкція ботнету зазвичай є дуже продуманим і стратегічним підходом, який застосовує хакер. Процес зазвичай починається зараженням однієї або декількох систем, потім ці системи несуть відповідальність за тиражування зловмисного коду на інші машини та в кінцевому підсумку переходять межі мережі, щоб перенести інфекцію на більш широку та глобальну область.

Щоб заразити одну систему, зловмиснику потрібно покластися на кілька способів вторгнення. Дуже часто використовується метод – заманити браузер на веб-сайт, на якому розміщено шкідливий JavaScript-код, або на сторінку, написану на подібній мові сценаріїв низького рівня, такі як python. Цей скрипт – це лише завантажувальна програма, яка виконує та створює незахищений ресурсний простір на машині. Потім сценарій підключається до однієї або декількох веб-сторінок того ж веб-сайту, на яких міститься реальне рекламне оголошення бот-мережі.

Потім завантажуються файли корисного навантаження і зберігаються приховані під прихованим простором. Це корисне навантаження містить усі пояснені вище модулі, які переймають контроль над машиною, і машина може бути заражена в цей момент. Покращені ботнетні мережі не вимагають

перезавантаження машини і здатні перетворити машину на зомбі в момент завантаження.

Ще одним відомим методом ін'єкції є встановлення зловмисного коду ботнету у вигляді інстальованих файлів на USB-накопичувач та введення коду на машину, яка дозволяє легко отримувати фізичний доступ та є вразливою чи незахищеною. Існує небагато вдосконалених методів, таких як змусити користувача запустити скрипт, надісланий як додаток, або приховати код у музичному файлі та поширити його за допомогою peer to peer мережі.

Згаданий вище процес, як правило, можливий тоді, коли недостатньо заходів безпеки. Наприклад, машина, яка не працює з антивірусним програмним забезпеченням або працює з усталеними визначеннями антивірусу, може легко стати здобиччю.

Аналогічно, непаткована або неправомірно виправлена система або мережа може виявити безліч уразливостей, які потрібно використовувати. У випадку захисту периметра мережі, залишення отворів у захисті у конфігурації брандмауера допоможе погіршити ситуацію. Що стосується серверів, то реалізація небезпечних політик або заходів, які не загартовують операційну систему сервера або не залишають експлуатаційні програми програми нефіксованими, може призвести до збитку.

Під час роботи з дистрибутивами Linux зазвичай проникають такі експлуатації, як переповнення буфера та віддалене виконання команд. Для збільшення розповсюдження використовуються типові рудиментарні мене, такі як надсилання фішингових листів, шпигунські додатки тощо. Дуже важливо пам’ятати, що зараження однієї машини в мережі достатньо, тому що ця машина, яка виступає зомбі, може легко копіювати ботнет-ко- де на інші машини в тій же мережі.

Відомі бот-нети.

На цьому етапі важливо обговорити декілька ботнетів, які відомі завдяки своїм горезвісним способам зараження систем та ще важко виявити.

Conficker.

Спочатку вважався вірусом, Conficker мав вбудовані програмні програми, які могли дозволити віддаленому керуванню зараженою машиною, що зробило це загрозою бота. Хоча це було написано для операційних систем Windows, пізніше було створено декілька варіантів, щоб також заразити системи Unix та Linux. Він використовував приховану вразливість файлів для вразливості Windows, щоб потрапити в машину, а потім перетворити її на зомбі, щоб поширити інфекцію далі. Коли кількість заражених інфекцій перевищує 10 мільйонів машин по всьому світу, Conficker все ще знаходиться в системах, які неправильно налаштовані або не захищені сильною системою захисту периметра.

Mariposa

Цей ботнет використовував шпигунські та шкідливі програми як транспортний засіб для введення машин та встановлення корисного набору модулів командного та контрольного центру. Мета Mariposa полягала в тому, щоб запускати в режимі "прихованості" і стежити за паролями, номерами кредитних карт, що вводяться на машині. Він також був запрограмований для перехоплення запитів Інтернет-браузера та заманювання на сторінки, що розміщують оновлені копії самого ботнету, а також рекламні сторінки.

Srizbi.

Цей ботнет був розроблений спеціально для створення мільярдів спам-повідомлень електронної пошти щодня. Він поширився в основному за допомогою піратського та безкоштовно завантаженого програмного забезпечення в Інтернеті, перетворивши кілька машин на зомбі. Він мав дуже маленький слід, що зробило виявлення дуже важким. Він мав інший модуль управління, за допомогою якого інфікований сервер був би власником, який контролює зомбі-армію, а інші заражені сервери просто стежать за ним і переймають його, якщо сервер контролера виходить з ладу або вимикає n. Як відомо, Srizbi створив масштабні спам-атаки по електронній пошті, що спричинили відмову в обслуговуванні на поштових серверах.

BredoLab

Це остання армія ботнетів, яка заразила понад 20 мільйонів машин у всьому світі. Хоча основною метою було створення масового спаму електронної пошти, цей ботнет також включав у свій корисний набір шпигунські програми та віруси. Відомо, що він заражає операційні системи Linux різних дистрибутивів і розгортає на них кореневі набори, щоб запускати в режимі схованості. Він був демонтований правоохоронними органами, але, як вважається, все ще існує у вигляді варіантів.

Ботнетів досить багато, як невеликих, так і масштабних, з мільйонами елементо в в своїй мережі. Як приклад можна привести ті з них, що були найбільш активними в 2018 році (і вони нікуди не поділися і сьогодні).

BetaBot

Кількість атак, скоєних цим Ботнии тому, складає більше 13% загальному обсязі унікальних атак 2018 року. Зловредів працював в 42 країнах, найбільший інтерес його творців викликали фінансові сервіси, соціальні мережі і великі портали.

Trickster (TrickBot)

Майже такий же активний ботнет, як і п редидущій. З його допомогою здійснено 12,85% від загального числа унікальних атак. Працював він у 65 країнах, атакував фінансові та кріптовалютние сервіси.

Panda

Цей ботнет зробив 9,84% від загального числа унікальних атак. Працював він у 33 країнах, атакував фінансів перші і кріптовалютние сервіси.

Серед успішних ботнетів варто вказати також SpyEye і Ramnit.

Окремо стоїть гігантський ботнет Mirai, який свого часу завдав шкоди на сотні мільйонів доларів США. Новий варіант ботнету вже активний і працює, поступово починаючи заражати нові пристрої по всьому світу .

Як ми дізналися, ботнети використовують усі можливі вразливості та створюють власну екосистему для злих причин. Незважаючи на те, що ботнетів важко виявити та знешкодити, є кілька запобіжних механізмів, які кожен

адміністратор мережі повинен прийняти у своїй інфраструктурі. Перший і головний механізм безпеки, який слід розгорнути, – це система оборони периметра. Правильно налаштований маршрутизатор і брандмауер повинні бути встановлені, а брандмауер повинен бути налаштований за допомогою автоматичного оновлення фільтрів проти спаму. Що стосується фізичної безпеки, то відключення USB-накопичувачів, компакт-дисків допоможе значною мірою.

Важливо, щоб шанувальники Linux знали, що Linux-дистрибутиви також не захищені від ботнетів, хоча відсоток зараження дещо менший, ніж Windows-машини. Системи Linux, які зазвичай розміщують веб-сервери та ферми FTP, зазвичай є першими цілями для розгортання корисного навантаження. Також рекомендується зміцнення та блокування файлових систем. Зловмисники, які планують ввести ботнет, можуть використовувати прості способи вторгнення в системи аутентифікації, через протокол SSH або через Інтернет.

Таким чином, використання суворої та складної схеми паролів дуже важливо. Було встановлено, що запуск непотрібних служб на виробничому сервері Linux – це звичайна практика, яку слід відмовляти–, оскільки вона відкриває бродячі порти, які залишаються без нагляду і, таким чином, стають задньою стороною для зловмисників. Отже, розуміння того, як атакують ботнети, є необхідним для системних адміністраторів, щоб розробити стратегію безпеки на основі своїх сценаріїв мережі.

4.2 Актуальні небезпеки і шкоди, що завдається малому і середньому бізнесу

Ботнети – це дійсно значна загроза, яка нікуди не зникла. У цій роботі проаналізуємо, які ботнети актуальні в 2019 році і якої шкоди вони можуть завдати бізнесу - НЕ великим корпораціям, а невеликим і середнім за розміром компаніям.

На кого спрямована загроза?

За даними Securelist , максимальний інтерес для розробників ботнетів представляють соб ой фінансові організації і сервіси. Саме на них припадає понад 70% відсотків всіх атак. Це сервіси онлайн-банкінгу, онлайн-магазини, різного роду платіжні агрегатори і т.п.

Друге місце (тут всього близько 6% всіх атак) займають соціальні мережі і кр упние інформаційні сайти, плюс пошуковики і сервіси соціальної пошти. Тут зловмисники намагаються захопити якомога більше персональних даних користувачів для проведення подальших атак.

Третє місце з приблизно 5% займають ресурси, що пропонують разние продукти і послуги, причому ці сайти не є онлайн-магазинами. У цю категорію потрапляють хостинг-провайдери та деякі інші організації. Зловмисників тут цікавлять, в першу чергу, особисті дані жертви.

Також «ботовод» направляли свої системи на різного роду кріптовалютние сервіси, включаючи біржі, гаманці і т.п.

Зрозуміло, що зловмисників, в першу чергу, цікавлять компанії з багатих країн, так що атакують сервіси та сайти, що базуються в США, Ве нії, Канаді, країнах Європи та Китаї.

Як ботнет шкодить бізнесу?

Розсилка спаму з IP-адрес компанії.

Ботнет може і не бути «заточений» під проведення якихось складних атак, а працювати в якості спам-інструменту. І чим більше елементів в ботнет, тим вище ефективність цього інструменту. Якщо спам розсилається з заражених пристроїв якоїсь компанії, то її IP автоматично потрапляють в спам-фільтри. А це означає, що через якийсь час все e-mail, відправлені співробітниками компанії з локальних ПК будуть потрапляти в спам у клієнтів, партнерів, інших контактів. Виправити цю ситуацію не так і просто, а шкода вона може завдати значної (можна уявити, як зривається домовленість через те, що важливі документи вчасно не потрапили в потрібні руки).

DDoS з IP-адрес компанії.

Приблизно те ж саме, що і в разі вище, тільки на цей раз ботнет задіє зар Ажен комп'ютери компанії для проведення DDoS-атак. В цьому випадку помічені в «темній справі» IP потраплять в чорний список різних провайдерів і будуть заблоковані. Надалі співробітники компанії, чиї ПК постраждали, будуть зазнавати труднощів з доступом м до певних ресурсів – запити будуть блокуватися на рівні провайдерів різного масштабу.

У деяких випадках компанії можуть вимкнути доступ до мережі взагалі, якщо атака була серйозною. А відсутність інтернету навіть протягом декількох годин становить серйозну проблему для бізнесу.

Пряма DDoS-атака на компанію.

Велика кількість ботнетів створені для проведення DDoS-атак. Потужність їх зараз досить велика, так що середній ботнет може повністю «покласти» сервіси і сайти звичайної компанії. А це дуже а дороге задоволення. За оцінкою фахівців пряма така може обходитися бізнесу в суми від $ 20 000 до $ 100 000 на годину .

Якщо навіть сервіси атакований компанії та для тривалого жают працювати, то значно повільніше, ніж зазвичай. Це загрожує прямими і непрямими збитками. І навіть у разі слабкої DDoS-атаки, яка не вплинуло на ефективність роботи компанії, можна зіткнутися з «брудними логами» - коли аналіз роботи сервісів кому панії неможливий через величезну кількості сторонніх IP. Google Analytics в таких випадках стає марним.

Крадіжка важливої інформації.

Ботнети, які існують сьогодні, багатофункціональні і складаються з великої кількості модулів. Оператор ботнету може перетворити «сплячий» ботнет в викрадача корпоративних даних (дані клієнтів, доступи до внутрішніх ресурсів, доступи до клієнт-банку і т.п.). одним помахом пальця. А це вже набагато більш чутлива для бізнесу загроза, ніж спам або DDoS.

Красти данн перші ботнет може різними способами, включаючи такий поширений як кейлоггінг. Кейлоггер може бути «заточений», наприклад, для

роботи з PayPal і активуватися тільки тоді, коли користувач намагається увійти до свого облікового запису.

Проксі для проведення а так.

Ботнет може перетворити корпоративні машини в проксі-сервера, які будуть служити «перевалочним пунктом» для проведення атак. І тут вже все набагато гірше, ніж у випадку зі спамом або DDoS - якщо атака була серйозною і завдала комусь шкоду, компані я може зіткнутися з пильною увагою з боку правоохоронних органів.

Витрачання ресурсів компанії.

Якщо ботнет поводиться активно, то на це можуть знадобитися значні обчислювальні ресурси. Тобто корпоративні машини будуть використ зоваться зловмисниками, з відповідними витратами енергії і процесорного часу. Якщо компанія, чиї комп'ютери заражені, працює з ресурсоємними витратами, це може позначитися на ефективності робочих процесів. Один із прикладів - Майнінг. Ботнет може бути активований в якості Майнера і тоді заражені ПК будуть віддавати значну частину своєї потужності видобутку монет для зловмисників.

Репутаційні втрати. Все це в підсумку може позначитися на репутації компанії, оскільки організація, чиї IP-а дрес помічені в «темних справах», виявиться складній ситуації. Повернути все на круги своя може виявитися не так і просто.

Як підрахувати збитки? Краще, звичайно, якщо збитків немає. Але якщо з'явилася проблема з ботнетом, то прорахувати поточні і майбутні затра ти, які потрібні для ліквідації проблеми, можна за алгоритмом нижче. Підсумовуємо всі витратні статті і отримуємо загальну суму.

Згідно з даними Ponemon Institute, в складних ситуаціях, коли робота компанії зупиняється, збитки можуть бути величезними і складати тисячі доларів за хвилину. У разі великих компаній це вже сотні тисяч. Для малого та середнього бізнесу загальний розмір збитків не так вели до, як у корпорацій, але для самої компанії навіть пара тисяч доларів може бути недозволеною розкішшю, якщо говорити про організацію невеликого розміру.

Як захиститися від ботнету? В принципі, методи захисту нічим не відрізняються від тих, що використовуються для попередження зараження комп'ютерів будь-якими шкідливими програмними продуктами. Це, в першу чергу, особиста «IT-гігієна», тобто потрібно віддавати собі звіт в тому, що перехід по посиланнях в e-mail повідомленнях, відкриття присилаються відомими і невідомими контактами файлів, клікання по банерах на кшталт «ваш ПК заражений і його потрібно лікувати » – все це загрожує зараженням не тільки особистої, а й корпоративної техніки.

У компаніях потрібно регулярно проводити інструктаж співробітників на тему інформаційної безпеки з демонстрацією різних кейсів. Люди повинні розуміти, що перехід по посиланню зі смішними котиками, яку надіслав особистий контакт, може загрожувати бізнесу. Найслабша ланка в ланцюжку інформаційного захисту бізнесу - людина, а не програмне забезпечення або залізо.

Але і програмно-апаратна захист повинен бути. Це програмні антивіруси, фаєрволи або «залізні» мережеві екрани на кшталт ZyWALL ATP500 , які забезпечують багаторівневий захист Multi-Layer Protection. Такі системи допомагають блокувати не тільки відомі, але й невідомі загрози (загрози нульового дня), а також запобігати DDoS середнього рівня. Залежно від величини мережі підприємства і її фінансових можливостей можна використовувати моделі Zyxel ATP200, ATP500 або ATP800.

У будь-якій компанії, чия робота в значній мірі залежить від комп'ютерів і ПО, включаючи хмарні сервіси, повинна бути детально пророблена стратегія інформаційної захисту. І це не просто листок паперу, який висить поряд з планом евакуації. Опрацювання стратегії означає, що запропоновані заходи повинні бути випробувані «в польових умовах», потрібно проводити тренінги та воркшопи з співробітниками. Все це не виключить, але значно знизить інформаційні загрози для бізнесу компанії.

5 ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

Метою цього розділу дипломної роботи є здійснення економічних розрахунків, спрямованих на визначення економічної ефективності від розробки, а також прийняття рішення щодо подальшого розвитку і впровадження або ж недоцільність впровадження відповідної розробки.

Передбачається, що описаний в роботі метод аналізу алгоритмів шифрування може бути імплементовано у вигляді спеціального програмного продукту. Розробка такого продукту вимагатиме певних затрат. Тому розрахуємо ці затрати.

Для здійснення оцінки потрібно зробити розрахунки трудомісткості кожної операції.

5.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

Витрати часу по окремих операціях технологічного процесу відображені в таблиці 5.1.

Таблиця 5.1 – Операції технологічного процесу та час їх виконання

№	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1.	Витрати праці на підготовку опису задачі	інженер	12
2.	Витрати праці на розробку проекту	інженер	20
3.	Витрати праці на розробку структури системи	інженер	15
4.	Витрати праці на створення системи по вибраному проекту та структурі	інженер	77
5.	Витрати праці на підготовку документації	інженер	15
6.	Витрати праці на відлагодження роботи зпроектованої системи при комплексній відладці	інженер	46
Разом			185

Загальні затрати на дипломний проект становить 185 годин.

5.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

Відповідно до Закону України “Про оплату праці” заробітна плата – це “винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу”.

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та додаткової оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов’язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

При розрахунку заробітної плати кількість робочих днів у місяці слід в середньому приймати – 24,5 дні/міс., або ж 196 год./міс. (тривалість робочого дня – 8 год.).

Місячний оклад кожного працівника слід враховувати згідно існуючих на даний час тарифних окладів. Згідно закону України «Про Державний бюджет України на 2018 рік», зокрема Статтею восьмою мінімальна заробітна плата у погодинному розмірі встановлена у розмірі 22,41 грн. Рекомендовані тарифні ставки: керівник дипломної роботи – 30,00...50,00 грн./год., інженер –

22,41...30,00 грн./год., консультант – 22,41...30,00 грн./год., технік – 22,41...30,00 грн./год., лаборант – 22,41...25,00 грн./год.

Основна заробітна плата розраховується за формулою:

$$З_{осн.} = Т_с \cdot К_г \,, \qquad (5.1)$$

де $Т_с$ – тарифна ставка, грн.;

$К_г$ – кількість відпрацьованих годин.

Оскільки всі види робіт в даному проекті виконує інженер, то основна заробітна плата буде розраховуватись тільки за однією формулою

$$З_{осн.} = 30 \cdot 175 = 5550 \text{ грн.}$$

Додаткова заробітна плата становить 10–15 % від суми основної заробітної плати.

$$З_{дод.} = З_{осн.} \cdot К_{допл.}, \qquad (5.2)$$

де $К_{допл.}$ – коефіцієнт додаткових виплат працівникам, 0,1–0,15 (візьмемо його рівним 0,15).

$$З_{дод.} = 5550 \cdot 0,15 = 832,50 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($В_{о.п.}$) визначаються за формулою:

$$В_{о.п.} = З_{осн.} + З_{дод.}. \qquad (5.3)$$

$В_{о.п.}$=5550+832,50 = 6382,50 грн.

Крім того, слід визначити відрахування на соціальні заходи:

1) ЄСВ + ПДФО 22 %;

2) військовий збір – 1,5 %.

У сумі зазначені відрахування становлять 23,5 %.

Отже, сума відрахувань на соціальні заходи буде становити:

$$B_{с.з.} = Ф_{оп} \cdot 0{,}235, \qquad (5.4)$$

де $Ф_{ОП}$ – фонд оплати праці, грн.

$$B_{с.з.} = 6382{,}50 \cdot 0{,}235 = 1499{,}89 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці зведемо у таблицю 5.2.

Таблиця 5.2 – Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівник-ків	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	Нарахув. на $Ф_{ОП}$, грн.	Всього витрати на оплату праці, грн. *6=3+4+5*
		Тарифна ставка, грн.	К-сть відпра-цьов. год.	Фактично нарах. з/пл., грн.			
А	Б	1	2	3	4	5	6
1	інженер	30	185	5550	832,50	1499,89	7882,39

Загальні витрати на оплату праці становить 7882,39 грн.

5.3 Розрахунок матеріальних витрат

Матеріальні витрати визначаються як добуток кількості витрачених матеріалів та їх ціни:

$$M_{в\,i} = q_i \cdot p_i, \qquad (5.5)$$

де: q_i – кількість витраченого матеріалу i-го виду;

p_i – ціна матеріалу i-го виду.

Звідси, загальні матеріальні витрати можна визначити:

$$З_{м.в.} = \sum M_{B\,i}. \qquad (5.6)$$

Проведені розрахунки занесемо у таблицю 5.3. Для розробки ПЗ передбачається покупка Visual Studio Team Foundation Server CAL SNGL LicSAPk OLP NL UsrCAL 2017, вартість якого на сьогодні становить 19400 грн.

Таблиця 5.3 – Зведені розрахунки матеріальних витрат

Найменування матеріальних ресурсів	Одиниця виміру	Норма витрат	Ціна за одиницю, грн	Затрати матеріалів, грн	Транс-портно-заготівель-ні витрати, грн	Загальна сума витрат на матеріа-ли, грн
1. Основні матеріали						
Програмне забезпечення	комп.	1	19400,00	19400,00	–	19400,00
2. Допоміжні матеріали						
Папір формату А4	шт.	200	0,18	36	–	36
Разом:						19436,00

Загальні матеріальні затрати становлять 19436,00 гривень.

5.4 Розрахунок витрат на електроенергію

Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$З_е = W \cdot T \cdot S\ , \qquad (5.7)$$

де W – необхідна потужність, кВт;

T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії.

Вартість кіловат-години електроенергії слід приймати згідно існуючих на даний час тарифів. Отже, 1 кВт з ПДВ коштує 2,42 грн.

Потужність комп'ютера для створення проекту – 550 Вт, кількість годин роботи обладнання згідно таблиці 5.1 – 169 годин.

Тоді, $З_е = 0{,}55 \cdot 185 \cdot 2{,}42 = 246{,}24$ грн.

5.5 Розрахунок суми амортизаційних відрахувань

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

Комп'ютери та оргтехніка належать до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60 % (квартальна – 15 %).

Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{Б_В \cdot Н_А}{100\%}, \qquad (5.8)$$

де A – амортизаційні відрахування за звітний період, грн.;

$Б_В$ – балансова вартість групи основних фондів на початок звітного періоду, грн.;

$Н_А$ – норма амортизації, %.

Для даного проекту засобом розробки є комп'ютер. Його сума становить 20000 грн. Отже, амортизаційні відрахування будуть рівні:

$$А = 20000 \cdot 5\% / 100\% = 1000 \text{ грн.}$$

Оскільки робота виконувалась 175 годин, то амортизаційні відрахування будуть становити:

$$А = 1000 \cdot 175 / 150 = 1166{,}67 \text{ грн.}$$

5.6 Обчислення накладних витрат

Накладні витрати пов’язані з обслуговуванням виробництва, утриманням апарату управління спілкою та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб’єкта, накладні витрати можуть становити 20 – 60 % від суми основної та додаткової заробітної плати працівників.

$$H_в = B_{о.п.} \cdot 0{,}2 \ldots 0{,}6 \,, \qquad (5.9)$$

де $H_В$ – накладні витрати.

Отже, накладні витрати:

$H_В = 6382{,}50 \cdot 0{,}2 = 1276{,}50$ грн.

5.7 Складання кошторису витрат та визначення собівартості НДР

Результати проведених вище розрахунків зведемо у таблицю 5.4.

Таблиця 5.4 – Кошторис витрат на НДР

Зміст витрат	Сума, грн.	В % до загальної суми
Витрати на оплату праці (основну і додаткову заробітну плату)	6382,50	21,3%
Відрахування на соціальні заходи	1499,89	5,0%
Матеріальні витрати	19436,00	64,8%
Витрати на електроенергію	246,24	0,8%
Амортизаційні відрахування	1166,67	3,9%
Накладні витрати	1276,50	4,3%
Собівартість	30007,79	100,0%

Собівартість ($С_В$) проекту розрахуємо за формулою:

$$С_в = В_{о.п.} + В_{с.з.} + З_{м.в.} + З_в + А + Н_в. \quad (5.10)$$

Отже, собівартість проекту дорівнює:

$С_в = 6382,50 + 1499,89 + 19436 + 246,24 + 1166,67 + 1276,50 = 30007,79$ грн.

5.8 Розрахунок ціни проекту

Ціну НДР можна визначити за формулою:

$$Ц = \frac{С_В \cdot (1 + Р_{рен}) + К \cdot В_{н.і.}}{К} \cdot (1 + ПДВ), \quad (5.11)$$

де $Р_{рен.}$ – рівень рентабельності, 30 %;

$К$ – кількість замовлень, од. (встановлюється лише при розробці програмного продукту та мікропроцесорних систем);

$B_{н.і.}$ – вартість носія інформації, грн. (встановлюється лише при розробці програмного продукту);

ПДВ – ставка податку на додану вартість, (20 %).

Оскільки розробка є прикладною, і використовуватиметься тільки для одного підприємства, то для розрахунку ціни не потрібно вказувати коефіцієнти K та $B_{і.н}$, оскільки їх в даному випадку не потрібно.

Тоді, формула для обчислення ціни розробки буде мати вигляд:

$$Ц = C_B \cdot (1 + P_{рен}) \cdot (1 + ПДВ). \quad (5.12)$$

Звідси ціна на проект складе:

Ц = C_B · (1+0,3)(1+0,2) = 46812,16 грн.

5.9 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Економічна ефективність (E_p) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_p = П / C_B, \quad (5.13)$$

де $П$ – прибуток;

C_B – собівартість.

Плановий прибуток ($П_{пл}$) знаходимо за формулою:

$$П_{пл} = Ц - C_B. \quad (5.14)$$

Розраховуємо плановий прибуток:

$П_{пл}$ = 46812,16 – 30007,79 = 16804,36 грн.

Отже, формула для визначення економічної ефективності набуде вигляду:

$$E_р = \frac{П_{пл}}{Св} . \quad (5.15)$$

Тоді, $E_р$ = 16804,36 / 30007,79 = 0,56

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень ($T_р$):

$$T_р = 1/ E_р , \quad (5.16)$$

Термін окупності дорівнює:

$T_р$ = 1/0,56 = 1,8 роки.

В цьому розділі дипломної роботи було розраховано основні техніко-економічні показники проекту (див. таблицю 5.5).

Розраховане значення економічної ефективності становить 0,56 що є високим значенням.

Так само нормальним є термін окупності. Для даного продукту він становить 1,8 роки.

Таблиця 5.5 – Техніко-економічні показники НДР

№ п/п	Показник	Значення
1.	Собівартість, грн.	30007,79
2.	Плановий прибуток, грн.	16804,36
3.	Ціна, грн.	46812,16
4.	Економічна ефективність	0,56
5.	Термін окупності, рік	1,8

Отже, даний проект може бути впроваджений та мати подальший розвиток, оскільки він є економічно вигідним за всіма основними техніко-економічними показниками.

6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

6.1 Питання охорони праці при організації робочого місця розробника програмного забезпечення

Організація робочого місця розробника модуля впливає на його працездатність.

У своїй діяльності розробник використовує комп’ютер, пристрої збереження інформації, а тому є необхідність забезпечення зручного доступу до всіх технічних засобів. Тому в даному розділі докладніше розглянемо відомості про систему ергономічних норм і принципів організації робочого місця, на котрому проводяться роботи зі створення модуля збору статистики.

Під робочим місцем розуміється зона, оснащена необхідними технічними засобами, у якій відбувається трудова діяльність виконавця або групи виконавців, які спільно виконують одну роботу або операцію.

Організація робочого місця полягає у виконанні заходів, які забезпечують безпечний і раціональний трудовий процес і ефективне використання знарядь та предметів праці, що підвищує продуктивність праці і знижує стомлюваність працівника.

Організація робочого місця залежить від характеру розв'язуваних задач і особливостей предметно-просторового оточення, що визначають робоче положення тіла і можливість пауз для відпочинку, типи і способи засобів відображення і керування, необхідність у засобах захисту, спецодягу, простору для налагодження і ремонту устаткування.

Одним з компонентів діяльності на робочому місці є робочі рухи. Їхня раціональна організація створює умови для зниження стомлення, резерви для підвищеної працездатності. Просторові характеристики руху оператора визначаються траєкторіями руху і розмірами моторного поля (зони досяжності).

При організації робочого місця необхідно забезпечити нормальні умови огляду. Зону огляду описує кут, вершина якого знаходиться в центрі ока, а

сторони складають границі, в яких людина при фіксованому положенні голови й ока добре розрізняє їхнє місцезнаходження.

У горизонтальній площині цей кут складає 300 – 400. При організації робочого місця кут огляду можна взяти 500 – 600, включаючи зону менш ясного огляду. Допустимий кут огляду по горизонталі 900. У вертикальній площині оптимальний кут огляду 100 вверх і 300 вниз від лінії погляду, а допустимий 300 вгору і 400 вниз від лінії погляду.

Щоб зберегти нормальну гостроту зору, робочу поверхню розташовують від очей на відстані від 0,3 м до 0,75 м. Робочі меблі повинні бути зручними для виконання робочих операцій. В даному випадку робочий стіл є основним устаткуванням. Особливо важливе значення має висота столу, його конструкція, яка повинна передбачати шухляди для розміщення інструментів, документації.

Важливе значення має конструкція робочих крісел. Погано підібрані крісла можуть бути причиною надмірної стомлюваності.

Нахил і висота крісла повинні регулюватися відповідно до висоти робочої поверхні і росту працюючого. Рекомендована ширина крісла 370 – 400 мм, глибина 370 – 420 мм, висота спинки 370 – 1000 мм від рівня крісла. Для розміщення ніг необхідно передбачити вільний простір під робочою площиною.

Праця людини, що протікає в умовах надмірного нервово-емоційного напруження, довготривалих статичних навантажень, обмеженої рухової активності призводить до неврозів, відхилень у психіці, захворювань опорно-рухового апарату, серцево-судинної системи тощо. Комп'ютери, телебачення, системи зв'язку та інші засоби, що використовують досягнення радіоелектроніки, є генераторами цілої низки електромагнітних випромінювань, вплив яких на організм людини ще не зовсім вивчений.

Сучасний розвиток науки та техніки приносить принципові нововведення у всі сфери матеріального виробництва, докорінно змінюючи знаряддя та предмети праці, технологію, методи обробки інформації. Разом з тим, захопившись вдосконаленням засобів праці залишено поза увагою проблеми людини в рамках своєрідної технічної та комп'ютерної революції. З широким

впровадженням автоматизації та комп'ютеризації виникла потреба врахування психологічних можливостей людини, таких як швидкість реакції, особливості пам'яті та уваги, емоційний стан та ін. Поява операторської діяльності призвела до суттєвих змін у фаховій структурі праці. Зменшились фізична важкість праці, ризик виробничого травматизму, однак разом з тим, на працюючу людину посилюється вплив нових, раніше не відомих чи мало вивчених несприятливих виробничих факторів фізичного, хімічного і особливо психофізіологічного характеру.

Проте, розвиток сучасної обчислювальної техніки відбувається не лише у бік покращення її технічних параметрів, але також звертається увага безпеку використання цієї техніки людиною шляхом зменшення потужності випромінювачів, зменшенням рівня випромінювання з моніторів, зменшення напруг живлення, покращення ергономічних характеристик.

Таким чином, в розділі з охорони праці виконано огляд питань безпечної роботи при створенні модуля інформаційної системи збору статистики та встановлено, що умови такої роботи відповідають вимогам з охорони праці, які застосовуються в галузі інформаційних технологій.

6.2 Охорона праці користувачів ПК

Дослідження, проведені фахівцями Всесвітньої організації охорони здоров'я (ВООЗ) показали, що у професійних операторів та канцелярських службовців, які у своїй діяльності використовують ВДТ, частіше зустрічаються порушення органів зору, опорно-рухового апарату, центральної нервової, серцево-судинної, імунної та статевої систем, захворювання шкіри. Необхідно зазначити, що вже в перші роки впровадження ВДТ в Європі та США була зафіксована значна кількість скарг операторського персоналу на загальне недомагання, передчасне стомлювання, головний біль, порушення функцій органів зору, які здійснювали несприятливий психофізіологічний вплив на самопочуття та працездатність

операторів. Однак, в той час основна увага приділялась розвитку техніки, а людина залишалась без необхідного захисту.

В умовах сучасного виробництва, яке характеризується масовим характером та широким застосуванням комп'ютерної техніки попередні пріоритети зазнали суттєвої трасформації. У центрі уваги вітчизняних та зарубіжних фахівців є питання щодо визначення характеру та умов праці користувачів комп'ютерів, функціональних змін у динаміці виконання трудових завдань, захворюваності та стану здоров'я, розробки засобів захисту.

Дослідження медиків-гігієністів, психологів, світлотехніків та фахівців з охорони праці та ергономіки показали, що сучасна професія користувача ВДТ належить до розумової праці, яка характеризується: високою напруженістю зорових функцій; одноманітною позою; великою кількістю стереотипних висококоординованих рухів, що виконуються лише м'язами кистей рук на фоні малої загальної рухової активності; значним нервово-емоційним компонентом, особливо в умовах дефіциту часу; роботою з великими масивами інформації, що викликає активізацію уваги та інших виших психічних функцій. Крім того, при роботі з дисплеями на електронно-променевих трубках виникає вплив на користувача цілої низки факторів фізичної природи – електростатичні поля, радіочастотне та рентгенівське випромінювання тощо.

Встановлено, що стан організму користувача значно залежить від типу роботи з ВДТ та умов її виконання. В загальному усі користувачі комп'ютерів поділяються на професіоналів та непрофесіоналів. До останніх можна віднести осіб, які використовують комп'ютер епізодично і він є для них не основним, а тільки допоміжним засобом (науково-технічні працівники, бібліотекарі, студенти, школярі, торгівельні працівники та ін.).

Діяльність професіоналів можна поділити на три групи:

1. Діяльність, яка пов'язана з виконанням нескладних багаторазово повторюваних операцій, що не вимагають великого розумового напруження. Наприклад, робота операторів комп'ютерного набору, працівників довідкових служб.

2. Діяльність, яка пов'язана із здійсненням логічних операцій, що постійно повторюються. Це робота інженера-економіста, інженера-проектувальника, оператора автоматизованого виробництва.

3. Діяльність, коли в процесі роботи необхідно приймати рішення за відсутності заздалегідь відомого алгоритму. Наприклад, робота інженера-гірограміста, диспетчерів руху залізничного транспорту, аеропортів тощо.

Необхідно зазначити, що такий поділ досить умовний, оскільки дане питання ще не достатньо розроблене і потребує детального вивчення. Проте, зрозуміло, що для кожної категорії користувачів комп'ютерів характерні свої особливості впливу комплексу несприятливих факторів трудового процесу та умов праці.

В Інституті медицини праці Академії медичних наук України проводились дослідження інтенсивності захворюваності осіб, що використовують у своїй роботі комп'ютер. Була вивчена захворюваність працівників з різною тривалістю використання комп'ютерів та характером діяльності самих користувачів. Розглядалися три групи користувачів: У першу увійшли системні інженери-програмісти (тривалість роботи за комп'ютером більше 6 год. на день), у другу – інженери-економісти, які у своїй роботі використовують уже розроблене програмне забезпечення (тривалість роботи від 4 до б год.), у третю – математики-постановникм завдань, які використовували комп'ютери не більше ніж 2 год. на день. Дані про захворюваність різних груп користувачів комп'ютерів та контрольної групи наведено у табл. 6.1.

Так, здорових серед обстежених користувачів ВДТ виявилось у кілька разів менше, ніж у контрольній групі. З наведених даних видно, що фізіологічні порушення частіше спостерігаються у користувачів, які довше та інтенсивніше використовували ВДТ.

Крім того, за даними ряду авторів у користувачів, які інтенсивно використовують комп'ютер в умовах значних розумових напружень досить часто (40–70%) виникають психологічні та поведінкові порушення (нервозність, роздратування, тривога, нерішучість, замкнутість тощо).

Таблиця 6.1 – Рівень захворюваності (%) осіб, тривалість та інтенсивність використання ВДТ у яких була різною

Стан здоров'я	Користувачі ВДТ			ПІ Контрольна група
	1 група	2 група	3 група	
Функціональні порушення ЦНС (астенопічний синдромів ін.)	15,6	8,2	6,3	2,7
Хвороби системи кровообігу	57,7	60,3	29,2	23,0
Хвороби органів дихання	20,0	21,7	11,2	4,1
Хвороби органів травлення	40,0	38,6	29,8	18,9
Здорові	6,7	20,1	29,8	46,6

Працівниками кафедри охорони праці та екології Української академії друкарства та Українського науково-дослідного інституту поліграфічної промисловості ім. Т. Шевченка проведені дослідження умов праці таї особливостей трудового процесу на комп'ютеризованих робочих місцях низки підприємств, що займаються видавничо-поліграфічною діяльністю (редакції, видавництва, друкарні). Було, зокрема, встановлено, що *зг,* суб'єктивними показниками (скарги) робота з В'ДТ викликає різноманітні симптоми негативного впливу на здоров'я користувачів. В таблицях 6.2 та 6.3 наведені характеристики скарг операторів комп'ютерного набору та редакторів і коректорів, які в процесі своєї роботи використовували комп'ютер. Результати досліджень наглядно ілюструють, що у працівників пізних професійних груп, що працюють за відеотерміналом комп'ютера переважають "очні" симптоми. Часті також головний біль та загальна втома, особливо в кінці робочого дня. Причому у коректорів та редакторів такі симптоми зустрічаються частіше. Досить значний відсоток скарг пов'язаний з опорно-м'язовою системою (біль в області спини та

шиї, втома м'язів рук), особливо в операторів комп'ютерного набору. Характерним також є той факт, що чим більший стаж роботи за комп'ютером, тим очевидніший його несприятливий вплив на здоров'я користувача.

Таблиця 6.2 – Характеристика скарг операторів комп'ютерного набору

№ пп	Симптоми впливу комп'ютера	Кількість працівників, що повідомили про симптоми від загальної кількості опитаних (%)		
		Стаж роботи		
		до 1 року	1 – 3 роки	3 – 5 років
1	Біль та різь в очах	58,8	67,5	88,7
2	Головний біль	17,6	23,3	42,5
3	Біль в області спини та шиї	18,5	21,2	32,2
4	Загальна втома	29,4	25,7	42,6
5	Втома м'язів рук	15,1	22,3	38,7
6	Підвищена роздратованість	11,7	21,6	35,3
7	Порушення нічного сну	8,3	15,5	20,6
8	Погіршення пам'яті	7,2	12,3	17,1

Таблиця 6.3 – Характеристика скарг редакторів і коректорів, які в процесі роботи використовували комп’ютер

пп	Симптоми впливу комп'ютера	Кількість працівників, що повідомили про симптоми від загальної кількості опитаних (%)		
		Стаж роботи		
		до 1 року	1 – 3 роки	3 – 5 років
1	Біль та різь в очах	50,2	64,3	84,6
2	Головний біль	24,7	42,4	55,7
3	Біль в області спини та шиї	12,1	17,7	22,5
4	Загальна втома	32,2	47,3	56,3
5	Втома м'язів рук	11,3	17,5	21,8
6	Підвищена роздратованість	18,4	32,8	44,3
7	Порушення нічного сну	12,7	24,3	32,4
8	Погіршення пам'яті	10,3	21,5	30,5

Під час проведення досліджень на багатьох комп'ютеризованих робочі місцях у видавництвах, редакціях та друкарнях виявлено відхилення гігієнічних та ергономічних вимог, що регламентовані відповідними нормам Так, оператори комп'ютерного набору часто працюють у несприятливих мікрокліматичних умовах, при недостатньому природному та нераціональному штучному освітленні, підвищеному рівні шуму тощо. Окрім того комп'ютеризовані робочі місця, як правило, не оснащені спеціальним] виробничими меблями, які мають необхідні регулювання для забезпеченні оптимальної робочої пози користувача. Сам же трудовий процес характеризується значними психо-емоційними навантаженнями, особливі при правці та коректурі тексту, монотонією, загальною гіподинамією ні фоні значних фізичних навантажень, що припадають лише на кисті руї (оператор комп'ютерного набору набирає за зміну залежно від

кваліфікації та складності тексту 25–40 тисяч знаків). Враховуючи несприятливий вплив цілого комплексу різноманітних виробничих факторів V користувачів можуть розвинутись певні розлади здоров'я, що пов'язані *з* роботою за комп'ютером.

6.3 Питання управління та природно-техногенні небезпеками

Визначені цілі є найвищою метою в системі управління безпекою життєдіяльності. В практиці мають місце багато цілей. В своїй сукупності вони створюють "дерево цілей" і відповідно до них існує "дерево завдань". Основними з "дерева цілей" є:

- ті, що стосуються встановлення, створення передумов для роботи і удосконалення системи;
- методичні цілі;
- загальносистемні цілі;
- ті, що формують можливість функціонування системи управління.

Завданням системи управління безпекою життєдіяльності є визначення природи і змісту упорядкування та адаптації за всіма складовими частинами системи управління, а також – шляхів використання під час їх реалізації в практиці безпеки життєдіяльності.

Кінцевим продуктом системи управління є легкокерований (виробничий чи інший) простір, який завдяки своєму існуванню створює передумови для виконання основних цілей і завдань.

Правову основу безпеки життєдіяльності становить Конституція України як за своїми юридичними особливостями, так і своїми принципами, тобто юридично вираженими об'єктивними закономірностями організації і функції соціально-економічної, політичної, духовної сфер суспільства, правового положення особи.

Конституційні норми, з одного боку, закладають суть безпеки (норми-принципи), а з іншого, – вказують на цілі подальшого розвитку і реалізацію

правового забезпечення безпеки життєдіяльності (норми-програми, норми-завдання, норми-зобов'язання).

Реалізація і розвиток основних конституційних положень, які регламентують суспільні правовідносини, безпосередніми суб'єктами яких є особа і держава, здійснюється за допомогою як чинних фундаментальних нормативно-правових актів (Кодексів України про адміністративні правопорушення, Кримінального), так і спеціальних (Кодексів України про працю, Земельного, Законів "Про охорону навколишнього природного середовища", "Про охорону атмосферного повітря" та ін.)

Поруч з нормативними актами, які прийняті вищим законодавчим органом держави, для встановлення взаємозв'язків, усунення програм, а в ряді випадків і реалізації окремих правових норм або їх елементів, до правової бази безпеки життєдіяльності належать спеціальні акти, розроблені за дорученням виконавчих державних органів усіх рівнів (Кабінет Міністрів, Міністерства, Державні Комітети та ін.).

Так, наприклад, "Положення...", які розвивають Закон України Про охорону праці, діляться на звичайні "Положення" і "Типові положення". Тут держава розподілила питання своєї прерогативи стосовно розробки нормативних актів і прерогативи своїх повноважень стосовно контролю, "правового простору" у вигляді нормативних актів підприємств.

З іншого боку, формуючи систему "Типових положень" держава на сьогоднішній день ліквідує прогалини в чинному законодавстві, узгоджує взаємозв'язки між суб'єктами правовідносин, створює юридичну базу для удосконалення і розвинення "правового поля" підприємств.

Кожний нормативно-правовий документ з безпеки життєдіяльності має свою структуру, яка визначає собою ідею систематизації відповідно зі своїм рівнем, метою та завданнями. Відповідно до цього в кожному нормативному акті є елементи, що відповідальні за зовнішній його зв'язок і створення передумов для відповідного розвинення за рахунок розробки нижчих нормативно-

законодавчих актів. Сама структура нормативного акту формує відповідні внутрішні зв'язки.

Основними систематизуючими ланками нормативних актів безпеки життєдіяльності (які за ієрархією знаходяться нижче законів) є встановлення взаємовідносин в галузі виробництва, в межах дії небезпечного фактора (в тому числі і факторів довкілля), а також відносно управління основних технологій безпеки життєдіяльності (розслідування нещасних випадків, навчання, організації робіт та ін.).

Узагальнюючими ланками систематизації на рівні держави є національна ідея, взаємовідносини в суспільстві, соціально-економічне і політичне становище держави, можливості сприймання і використання законодавчих актів з боку споживачів та ін.

Природно-техногенні небезпеки належать до так званих комбінованих небезпек, тобто таких, які є результатом впливу декількох чинників. Як правило одним з чинників є природа, а іншим – людина.

Небезпечне природне явище – подія природного походження або результат діяльності природних процесів, які за своєю інтенсивністю, масштабом поширення і тривалістю можуть вражати людей, об'єкти економіки та довкілля.

Справжнім лихом є землетруси, повені, зсуви, селеві потоки, бурі, урагани, снігові заноси, лісові пожежі. Тільки за останні 20 років вони забрали життя понад трьох мільйонів чоловік. За даними ООН, за цей період майже один мільярд жителів нашої планети зазнали шкоди від стихійних лих.

На території України можливе виникнення практично всього спектра небезпечних природних явищ і процесів геологічного, гідрогеологічного та метеорологічного походження.

Серед надзвичайних ситуацій природного походження в Україні найчастіше трапляються:

– геологічно небезпечні явища, такі як зсуви, обвали та осипи, просадки земної поверхні різного походження та ін.;

– метеорологічні небезпечні явища, такі як зливи, урагани, сильні снігопади, сильний град, ожеледь;

– гідрологічно небезпечні явища, такі як повені, паводки, підвищення рівня ґрунтових вод та ін.;

– природні пожежі лісових та хлібних масивів;

– масові інфекції та хвороби людей, тварин і рослин.

Виходячи з визначення стихійного лиха як природного явища, що безпосередньо впливає на стан навколишнього середовища і добробут населення і є екстремальним екологічним фактором, територія України характеризується дуже складними умовами, що визначає полігенетичний характер стихійних лих та певні просторові закономірності їх прояву в різних географічних зонах і районах.

Надзвичайні ситуації природного походження в Україні поділяються на: геологічні, географічні, метеорологічні, агрометеорологічні, морські гідрологічні, гідрологічні небезпечні явища, природні пожежі, епідемії, епізоотії, епіфітотії.

Стихійні явища, як правило, виникають в комплексі, що значно посилює їх негативний вплив. Небезпечні природні явища в основному визначаються проявом трьох головних груп факторів – ендогенних, екзогенних та гідрометеорологічних процесів.

Стихійні лиха, що трапляються на території України, можна поділити на прості, що містять один елемент, наприклад, сильний вітер, зсув або землетрус, та складні, що містять декілька одночасно діючих процесів однієї або кількох груп, наприклад, негативних атмосферних та геодинамічних екзогенних процесів, ендогенних, екзогенних та гідрометеорологічних процесів у поєднанні з техногенними.

Аварії природного характеру класифікуються за такими основними ознаками:

– за масштабами наслідків відповідно до територіального поширення;

– за розмірами заподіяних (очікуваних) економічних збитків та людських втрат;

– за кваліфікаційними ознаками надзвичайних ситуації.

6.4 Державна система управління БЖД

Державна система управління безпекою життєдіяльності є складовою частиною загальної системи управління державою. Тому основним (найголовнішим) завданням реалізації такої системи є її повна адаптація до системи управління державою.

Другим за ступенем важливості завданням є розробка передумов її існування (побудови, функціонування, удосконалення та ін.), що здійснюється завдяки:

- розробці «правового поля»;
- розробці складу відповідних управлінських структур та їх повноважень, що реалізують політику держави у сфері безпеки життєдіяльності;
- розробці системи взаємодій (загальних і конкретних) під час виконання функцій і завдань управління безпекою життєдіяльності.

Третій склад завдань державної системи управління — це формування передумов для розгортання нижчих систем управління: регіональних, галузевих, а також виробничих та інших підприємств.

Четвертий склад завдань — це розробка змісту і характеру взаємодій між усіма системами управління.

Визначені завдання є основними. їх склад у практиці управління постійно зростає відповідно до нових завдань кожної з систем управління, що зорієнтовані щодо напрямів удосконалення безпеки життєдіяльності.

Державна система має створювати правову й організаційно-методичну базу існування систем управління нижчого рівня (регіональних, галузевих, виробничих підприємств та ін.). Найбільш прийнятно це буде виглядати як

типове положення про систему управління. Тепер така розробка планується. її поява складе правову основу для забезпечення розбудови і функціонування вищезгаданих нижчих за рівнем систем управління. Це також складе передумови для єдиного розуміння питань управління в безпеці життєдіяльності.

Наведемо основні завдання і функції державної системи управління, що мають загальний і спеціальний характер.

До таких функцій належать:

- планування робіт;
- розробка, прийняття і відміна нормативних актів;
- професійний відбір;
- навчання з питань безпеки;
- регламентація процесу праці;
- атестація робочих місць за умовами праці, паспортизація об'єкта;
- реєстрація й облік;
- експертиза;
- ліцензування та сертифікація;
- забезпечення безпеки обладнання, процесів, будівель, споруд і територій;
- забезпечення санітарно-гігієнічних умов праці, санітарно-побутового обслуговування, лікувально-профілактичного і медичного обслуговування;
- узгодження і видача дозволів;
- попередження про виникнення небезпечних ситуацій;
- розслідування й облік нещасних випадків;
- розслідування й облік хронічних професійних захворювань;
- розслідування й облік аварій;
- управління фондами;
- стимулювання охорони праці;
- пропаганда і виховання безпечної поведінки;
- наукове забезпечення;

- міжнародне співробітництво та ін.

Зміст основних документів, що формують політику держави у сфері безпеки життєдіяльності:

Екологічна безпека. (Основні напрями державної політики України у галузі охорони довкілля, використання природних ресурсів та забезпечення екологічної безпеки. Затверджено Постановою Верховної Ради від 5 березня 1998 р. № 188/98-ВР).

Реалізація «Основних напрямів» передбачає три етапи.

На першому етапі (1997—2000 рр.) необхідно завершити і реалізувати невідкладні заходи щодо обмеження шкідливого впливу на довкілля найбільш небезпечних джерел забруднення.

Основними завданнями цього етапу є:

- вдосконалення законодавчо-правової бази з питань охорони довкілля і раціонального використання природних ресурсів;
- розроблення і впровадження економічного механізму охорони довкілля та раціонального природокористування;
- створення системи досконалого, повного та адекватного контролю за екологічним станом довкілля з одночасним запровадженням елементів комплексного міжвідомчого екологічного моніторингу;
- здійснення першочергових заходів для стабілізації стану довкілля;
- розроблення і впровадження програм екологічної освіти, виховання та екоінформування населення.

На другому етапі (протягом 10...15 років, починаючи з 1998 р.) планується розробити і розпочати реалізацію комплексних програм, орієнтованих на досягнення балансу між рівнями шкідливого впливу на довкілля і його здатністю до відновлення.

Основними завданнями цього етапу є:

- оптимізація структури природокористування;
- екологічно орієнтована структурна перебудова економіки;

− розроблення і впровадження в Україні системи державного моніторингу довкілля, створення системи аналізу екологічної ситуації, прогнозування, планування і здійснення запобіжних заходів щодо ймовірних чинників шкідливого впливу.

На третьому етапі планується створити систему державного управління використанням природних ресурсів, регулюванням техногенного впливу на довкілля як основу управління сталим розвитком суспільства. Фрагментарне здійснення цих заходів розпочалося 1996 р., а більш широке — відповідно до темпів стабілізації економіки країни.

Основними завданнями цього етапу є:

− подальший розвиток системи державного моніторингу навколишнього природного середовища, створення автоматизованої системи оцінки екологічних ситуацій, прогнозування шкідливого впливу на довкілля, планування дій у надзвичайних ситуаціях на основі оцінок і сценаріїв розвитку подій;

− належна координація раціонального використання природного та соціально-економічного потенціалу з урахуванням екологічних чинників на засадах сталого розвитку.

Внаслідок реалізації «Основних напрямів» державної екологічної політики буде створено систему екологічно збалансованого управління розвитком суспільства, яка стимулюватиме відновлення природних властивостей довкілля, компетентного регулювання використання природних ресурсів та розвиток продуктивних сил країни.

До головних складових механізму реалізації державної екологічної політики належать:

− державна інституційна інфраструктура проведення природоохоронної політики;

− законодавчо-правовий механізм регулювання виробничої діяльності юридичних і фізичних осіб щодо охорони, використання природних ресурсів та їх відходів;

− економічний механізм природокористування та природоохоронної діяльності;

− механізм реалізації міжнародних, національних, регіональних, галузевих та місцевих природоохоронних програм.

Охорона праці. («Національна програма поліпшення стану безпеки, гігієни праці та виробничого середовища...». Постанова Кабінету Міністрів України на 1996—2000 роки від 2 листопада 1996 р. № 1345 «Державна програма навчання та підвищення рівня знань працівників населення України з питань охорони праці на 1996—2000 роки». Постанова Кабінету Міністрів України від 18.04.96, Указа Президента України від 18.10.97 р. № 1166/97 «Про основні напрями соціальної політики на 1997—2000 роки» та ін. Більшість документів стосується розвитку лише до 2000 р., тому обговорювати їх в широкому плані немає сенсу). Визначені основні напрями науково-дослідних, проектних та інших робіт в країні у галузі, що формує політику цієї сфери за напрямами: безпосередньо системи управління, наукових досліджень, навчання, проектних рішень та ін.

Надзвичайні ситуації. Політика в цій сфері фактично формується двома документами — «Концепцією захисту населення і територій в разі загрози та виникнення надзвичайних ситуацій», схваленою Указом Президента України від 26 березня 1999 р. № 234/99 і Закону України «Про війська цивільної оборони України», прийнятого Президентом України 24 березня 1999 р. № 556-XIV.

7 ЕКОЛОГІЯ

7.1 Моніторинг поверхневих вод

Поверхневі води – це води суходолу, що постійно або тимчасово перебувають на земній поверхні у формі різних водних об’єктів у рідкому (водотоки, водоймища) і твердому (льодовики, сніговий покрив) станах.

Господарсько-побутові, промислові, сільськогосподарські скиди зумовлюють хімічне, фізичне, біологічне й теплове забруднення гідросфери.

Хімічне забруднення води відбувається внаслідок надходження у водоймища зі стічними водами шкідливих домішок неорганічного й органічного походження: сполук миш’яку, свинцю, ртуті, міді, кадмію, хрому, фтору, а також нафти та нафтопродуктів. Вони поглинаються фітопланктоном і передаються далі трофічним ланцюгом іншим організмам, що супроводжується кумулятивним ефектом. Більшість цих домішок є токсичні для мешканців водоймищ.

Згубно впливають на стан водоймищ стічні та скидні води, що містять розчинені органічні речовини або суспензії органічного походження, оскільки призводять до зниження вмісту кисню у воді.

Вода скидна – вода, що відводиться від зрошуваних сільгоспугідь, присадибних ділянок, а також з територій, на яких застосовується гідромеханізація.

Вода стічна – вода, що утворюється в процесі господарсько-побутової та виробничої діяльності (крім дренажної та скидної води), а також під час відведення із забрудненої території стоку атмосферних опадів.

Кількість хімічних забруднювачів постійно зростає. Про шкідливу дію деяких із них ще мало відомо, оскільки вони мають пролонгований вплив, тобто шкідливі мутації, генетичні розлади тощо виявляються в наступних поколіннях живих істот.

Фізичне забруднення води зумовлює зміни фізичних властивостей – прозорості, вмісту суспензій та інших нерозчинних домішок, радіоактивності й температури тощо.

Біологічне забруднення водного середовища полягає в надходженні зі стічними водами до водоймищ різних видів мікроорганізмів, рослин і тварин (віруси, бактерії, гриби, черв'яки), невластивих водній екосистемі. Більшість із них є хвороботворні. Найшкідливіші є комунально-побутові стоки. Промислові біологічні забруднювачі – це підприємства шкірообробної промисловості, м'ясокомбінати, цукрові заводи.

7.1.1 Основні завдання та організація роботи системи моніторингу поверхневих вод

Моніторинг поверхневих вод – система послідовних спостережень, збору, обробки даних про стан водних об'єктів, прогнозування їх змін та розробки науково обґрунтованих рекомендацій для прийняття управлінських рішень, які можуть позначитися на стані вод.

Основна мета налагодження системи спостережень і контролю за забрудненням водних об'єктів – це отримання інформації про природну якість води та оцінка змін якості води внаслідок дії антропогенних факторів.

Служба спостережень та контролю (моніторингу) виконує такі завдання:

– спостереження та контроль рівня забруднення водного середовища за хімічними, фізичними та гідробіологічними показниками;

– вивчення динаміки вмісту забруднюючих речовин і виявлення умов, за яких мають місце коливання рівня забруднення;

– дослідження закономірностей процесів самоочищення та накопичення забруднюючих речовин у донних відкладах.

В Україні сьогодні згідно з «Порядком здійснення державного моніторингу вод» та «Положенням про державну систему моніторингу навколишнього середовища» державний моніторинг вод є невід'ємною складовою частиною державної системи моніторингу довкілля. На основі цих

двох урядових документів розроблена «Єдина міжвідомча інструкція з організації та здійснення державного моніторингу вод» (ЄМІ). Цей документ встановлює єдині вимоги до організації та проведення спостережень за станом поверхневих вод, прибережних зон водосховищ, підземних вод, джерел забруднення вод, за гідрологічними, фізико-хімічними, біологічними, радіологічними показниками якості вод. Виконання вимог ЄМІ обов'язкове для всіх підрозділів суб'єктів державного моніторингу вод, а також відповідальних водокористувачів, які здійснюють спостереження за кількісним та якісним станом вод.

До головних суб'єктів державного моніторингу належать: Міністерство екології та природних ресурсів, у тому числі Головдержекоінспекція та Держуправління охорони навколишнього природного середовища в областях, організації Гідрометеорологічної служби; геологічні територіальні організації; Міністерство з питань надзвичайних ситуацій; Міністерство охорони здоров'я; Міністерство аграрної політики; Державний комітет України з водного господарства; Державний комітет будівництва, архітектури та житлової політики України.

Основний обсяг робіт із моніторингу річок виконують пункти спостережень Гідрометеослужби. Ці пункти розподілені за 10 річковими басейнами України. Найбільше пунктів спостережень розташовано в басейні Дніпра, розвинена мережа спостережень у басейнах Дунаю та Дністра. Сучасна гідрологічна мережа України налічує 374 пости.

7.1.2 Принципи організації контролю якості поверхневих вод

Моніторинг забруднення вод проводиться на постійних та тимчасових пунктах спостережень, які розміщують у місцях, де наявний або відсутній вплив господарської діяльності.

Пункт спостереження за якістю поверхневих вод – місце на водоймищі або водотоці, де проводять комплекс робіт для одержання даних про якісні й кількісні характеристики води.

Основними об’єктами, які потребують моніторингу, є: місця скидання стічних і дощових вод міст, селищ, сільськогосподарських комплексів, стічних вод окремих підприємств, ТЕС, АЕС; місця скидання колекторно-дренажних вод, які відводяться зі зрошуваних або осушуваних земель; кінцеві створи великих і середніх річок, які впадають у моря, внутрішні водоймища; кордони економічних районів, республік, країн, що перетинають транзитні річки.

На пунктах спостережень досліджують один або кілька створів.

Створ пункту спостереження – умовний поперечний переріз водоймища або водотоку, де проводиться комплекс робіт для одержання інформації про якість води.

Створи спостережень розміщують з урахуванням гідрометричних умов і морфологічних особливостей водоймища, наявності джерел забруднення, об’єму та складу стічних вод.

На водотоках у разі відсутності організованого скидання зворотних вод, у гирлах забруднених приток, на незабруднених ділянках водотоків, на кінцевих ділянках річок і в місцях перетину державного кордону України встановлюють один створ.

На водотоках за наявності організованого скидання зворотних вод встановлюють два і більше створів. Перший (фоновий) створ рекомендується розміщувати на відстані 1 км вище від джерела забруднення, другий – у зоні забруднення, на відстані 1 км вище від найближчого місця водозабору, третій – у місці достатнього змішування стічних вод із водами річки.

У процесі спостережень за водоймищем загалом встановлюють не менше трьох створів, по можливості рівномірно розподілених його акваторією з урахуванням конфігурації берегової лінії.

Кожний створ має кілька вертикалей та горизонталей.

Вертикаль створу – умовна вертикальна лінія від поверхні води до дна водоймища або водотоку, на якій здійснюють дослідження для отримання інформації про якість води.

Кількість вертикалей у створі на водотоці визначають з урахуванням умов змішування вод водотоку зі зворотними водами, а також із водами приток. За неоднорідного хімічного складу води у створі встановлюють не менше трьох вертикалей, а за однорідного – одну вертикаль на стрижні водотоку. Кількість вертикалей залежить також від ширини зони забруднення.

Горизонт створу – зона на вертикалі (углиб), де виконують комплекс дослід-жень для отримання інформації про якість води.

Кількість горизонтів на вертикалі визначають з урахуванням глибини водного об’єкта. Крім того, необхідно відокремити додаткові горизонти в кожному шарі зміни густини води.

7.1.3 Показники якості води

Оскільки не існує єдиного показника, який визначав би весь комплекс характеристик води, оцінювання якості води проводиться на основі системи показників. Ці показники поділяються на фізичні, бактеріологічні, гідробіологічні та хімічні. Інша форма класифікації показників якості води – їх розподіл на загальні та специфічні. До загальних відносять показники, характерні для будь-яких водоймищ. Присутність у воді специфічних показників обумовлена місцевими природними умовами, а також особливостями антропогенного впливу на водний об’єкт.

До основних фізичних показників якості води належать: температура, запах, прозорість, кольоровість, уміст зважених речовин.

Бактеріологічні показники характеризують забрудненість води патогенними мікроорганізмами. До найважливіших бактеріологічних показників відносять: колі-індекс – кількість кишкових паличок у літрі води; колі-титр – кількість води в мілілітрах, у якій може бути знайдена одна кишкова паличка.

Гідробіологічні показники дають змогу оцінити якість води за тваринним населенням та рослинністю водоймищ. Зміна видового складу водних екосистем може відбуватися за настільки слабкого забруднення водних об’єктів, яке не

виявляється жодними іншими методами. Тому гідробіологічні показники є найбільш чутливі.

Фізичні, бактеріологічні та гідробіологічні показники відносять до загальних показників якості води.

Хімічні показники можуть бути загальними та специфічними. До загальних хімічних показників якості води належать: уміст розчиненого кисню, хімічне та біохімічне споживання кисню; водневий показник; уміст азоту і фосфору та мінеральний склад.

До найбільш поширених специфічних показників якості води відносять феноли, нафтопродукти, поверхнево-активні речовини (ПАР), синтетичні поверхнево-активні речовини (СПАР), пестициди і важкі метали..

7.2 Індексний метод в екології

Статистична практика при вивченні екологічних явищ широко використовує індекси (хоча деякі екологи не підготовлені для такої роботи). Знання методології побудови індексів значно розширює аналітичні можливості дослідника, збагачує результативну інформацію досліджень.

Індекс англійський термін «index number» означає число-показник. Статистичні індекси – це відносні величини, які одержують внаслідок порівняння складних екологічних явищ, утворених з різнорідних елементів, що не підлягають безпосередньому підсумовуванню.

Індекс у статистиці узагальнюючий відносний показник, який характеризує співвідношення в часі чи просторі соціально-екологічних явищ і процесів. За своєю суттю статистичний індекс характеризує зміну рівня будь-якого суспільного явища в часі, просторі чи порівняно з планом, нормою, стандартом. У цих випадках зіставляються між собою числові значення однойменних показників, що мають однаковий екологічний зміст. Отже, індексом можна назвати відносну величину динаміки, виконання плану, порівняння.

За допомогою індексів можна характеризувати зміну в часі і просторі найрізноманітніших показників: обсяги викидів в атмосферу, скидів шкідливих речовин у водне середовище, інтенсивність забруднень і т. д. їх поділяють на дві групи: до першої належать об'ємні (сумарні) показники (наприклад, обсяг викидів та скидів кількість забруднювачів, площа забрудненої території та ін.), які виражаються абсолютними величинами; до другої – показники, розраховані на певну одиницю (наприклад, викиди в розрахунку на одиницю земельної площі або на одного жителя, працівника і т.д.). Останні умовно можна назвати якісними показниками, і виражаються вони у вигляді середніх величин. Ця особливість зумовлює поділ індексів на індекси кількісних та індекси якісних показників.

За допомогою статистичних індексів можна відображувати зміну в часі і просторі як окремих простих показників (наприклад, обсяг викидів вуглецю, окислів азоту, сірки і т.д.), так і однойменних показників за складними сукупностями (наприклад, зміна обсягу викидів по місту, району, області в цілому і т.д.).

За допомогою індексного методу вирішуються такі завдання:

– характеризують загальну зміну складного економічного явища чи окремих його елементів (складових);

– виділяють вплив одного з факторів через елімінування впливу інших;

– відокремлюють впливу зміни структури явища на зміну індексованої величини.

При цьому сама міра впливу може бути визначена як у відносних вимірниках, так і в абсолютних

Класифікація індексів. Класифікують індекси за різними ознаками:

– за змістом досліджуваних об'єктів, явищ і процесів індекси обсягу, індекси якісних показників;

– за повнотою охоплення елементів сукупності індивідуальні індекси, зведені (групові, загальні) індекси;

– за формою зображення агрегатні індекси, середні зважені індекси (арифметичні, гармонійні);

– за базою порівняння індекси динаміки (базові, ланцюгові), індекси виконання плану, територіальні індекси;

– за характером впливу на зміну складного явища індекси сталого складу, індекси структурних зрушень;

– за коефіцієнтами співвимірювання індекси зі змінними вагами, індекси зі сталими вагами.

Обчислення загальних індексів, що дають змогу співвіднести між собою показники за складними сукупностями, являє собою особливий прийом дослідження, який називається індексним методом. За його допомогою можна не тільки вивчати динаміку показників, а й вимірювати вплив окремих факторів на динаміку складного показника. При цьому залежно від завдань аналізу можна фактори вивчати ізольовано, абстрагуючись від дії інших, або розглядати їх взаємопов'язано.

Методологічні принципи побудови індексів. Індексний метод має свою термінологію та символіку. Її дотримання є обов'язковою умовою в індексному аналізі.

Для побудови статистичного індексу необхідно мати вихідну інформацію, як мінімум, за два періоди. Один з таких періодів називається базисним, другий – поточним. Базисний – це період, з яким порівнюють досліджувані явища, поточний – період, що порівнюється. Так, в індексах динаміки базисним є показник попереднього періоду (моменту) часу, в індексах порівняння з нормативною базою нормативний рівень, а в індексах порівняння (в просторі) базисним може бути показник, що належить до якоїсь з територій. Якщо досліджуються дані за кілька періодів, то один з них (як правило, початковий) буде базисним, а решта – поточними, або звітними.

У теорії індексів показник, зміну якого характеризує індекс, називають індексованою величиною, а пов'язану з нею величину, що використовують як постійну, – елімінованою величиною, або вагою. Остання відіграє роль сумірника. Використання цих двох видів величин вважається особливістю

індексного методу аналізу. При побудові статистичних індексів насамперед необхідно вирішити такі питання:

- який набір різнорідних елементів досліджуватиметься;
- які показники виступатимуть індексованими величинами;
- які величини виступатимуть сумірниками (вагами).

При цьому встановлюють, які досліджувані показники при побудові індексів вважаються базисними, а які – поточними.

Стандартні позначення, що використовують при побудові індексів:

– підписна нумерація за її допомогою позначається період, до якого належать дані показники базисного періоду мають у формулах підрядковий знак «0», а поточного «1»; якщо зміна явища вивчається не за два, а більше періодів, то кожний з них позначається відповідно «0», «1», «2», «3» тощо.;

– основні умовні позначення показників: x рівень показника, який вивчається; x_0 рівень показника за базовий період;

– x_i рівень цього ж показника за поточний період (якісний показник);

– u статистична вага показника в ряду розподілу, або об'ємний показник;

– u_0 і u_i теж за базисний та поточний періоди;

– i індивідуальний індекс;

Числове значення індексу (i) означає, що відповідний показник за досліджуваний період змінився в (i) разів, на певну кількість відсотків.

ВИСНОВОК

У цьому дослідженні були вивчені існуючі моделі машинного навчання в контексті виявлення ботнетних мереж P2P. І була запропонована краща модель для штучних нейронних мереж на основі байєсівської регуляризації, яка є дуже ефективною в контексті проблеми, яка потребує хороших узагальнюючих здібностей.

Таким чином, за допомогою статистичних тестів остаточно показано, що навчена модель BR-ANN здатна дуже добре узагальнити і здатна передбачити активність невідомих шкідливих дій ботів. Виявлення конкретного виду шкідливої діяльності та тим самим ідентифікація невідомого бота може допомогти експертам з безпеки вжити відповідних профілактичних заходів.

Завдяки великій пропускній здатності мережі, перехід на масштабовану та розподілену архітектуру є дуже привабливою альтернативою. Інтеграція пропонованої системи API з Mahout та Hadoop API та перезапис класифікатора відповідно до парадигми Map-Reduce – можливий майбутній напрямок розвитку цієї роботи.

ПЕРЕЛІК ПОСИЛАНЬ

1. “ZeroAccess Botnet Resumes Click-Fraud Activity,” SecureWorks, Jan. 2015. [Online]. Available: https://www.secureworks.com/blog/ zeroaccess-botnet-resumes-click-fraud-activity-after-six-month-break
2. O. Kupreev, J. Strohschneider, and A. Khalimonenko,“Kaspersky DDOS intelligence report for Q3 2016,” Kaspersky lab, Tech. Rep., Oct. 2016. [Online]. Available: https://securelist.com/analysis/quarterly-malware-reports/76464/ kaspersky-ddos-intelligence-report-for-q3-2016/
3. N. Falliere, L. O Muruchu, and E. Chien, “W32.Stuxnet Dossier,” Symantec Corporation, Tech. Rep. Version 1.4, Feb. 2011. [Online]. Available:
4. R. Sommer and V. Paxson, “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection,” in 2010 IEEE Symposium on Security and Privacy (SP), May 2010, pp. 305–316.
5. S. Garcıa, “Survey on Network-based Botnet Detection Methods,” Secur. Commun. Netw., 2013.
6. G. Gu, R. Perdisci, J. Zhang, and W. Lee, “BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection,” ser. 2, vol. 5, 2008, pp. 139–154.
7. A. B. Ashfaq, Z. Abaid, M. Ismail, M. U. Aslam, A. A. Syed, and S. A. Khayam, “Diagnosing bot infections using Bayesian inference,” J Comput Virol Hack Tech, pp. 1–18, Sep. 2016.
8. G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, “BotHunter: Detecting Malware Infection Through IDS-driven Dialog Correlation,” in M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones, “SOCKS Protocol Version 5,” RFC 1928, Mar. 1996.
9. Методичні вказівки по виконанню організаційно-економічної частини дипломних проектів науково-дослідницького характеру для студентів спеціальності 7.080401 “Інформаційні управляючі системи та технології” /

Кирич Н.Б., Зяйлик М.Ф., Брощак І.І., Шевчук Я.М – Тернопіль, ТНТУ, – 2009. –11 с.

10. Основы охраны труда: учебник / А. С. Касьян, А. И. Касьян, С. П. Дмитрюк. – Дн-ськ: Журфонд, 2007. – 494 с.
11. Безпека життєдіяльності: Навч. посібник./ За ред. В.Г. Цапка. 4–те вид., перероб. і доп. – К.: Знання, 2006. – 397 с.

ДОДАТКИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

VII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ »

11–12 грудня 2019 року

ТЕРНОПІЛЬ
2019

УДК 004.056
Р. Яворський, В. Амбок, В. Леньо
Тернопільський національний технічний університет імені Івана Пулюя

ПРОБЛЕМИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ПРИ РОЗГОРТАННІ СИСТЕМ ВИЯВЛЕННЯ ВТОРГНЕНЬ

UDC 004.056
R. Yavorskii, V. Ambok, V. Lenio
(Ternopil Ivan Puluj National Technical University, Ukraine)

INFORMATION SECURITY CHALANGES FOR DEPLOYMENT OF INTRUSION DETECTION SYSTEMS

Розглянемо основні класи небезпек, характерних для розгортання систем виявлення вторгнень на основі віртуальних машин – Virtual Machines (VM), оскільки саме вони є основним елементом побудови інформаційної інфраструктури організації у хмарних сервісах [1].

VM image sharing. Вважається, що існує репозиторій образів різних VM, а користувач на їх основі може сконфігурувати потрібний образ. Таке використання образів з репозиторію може спричинити появу вразливостей у системі [2]. Зловмисник може знайти вразливості в існуючому образі або завантажити у репозиторій власний, шкідливий, образ VM.

VM isolation. З іншого боку проблему становить використання VM в ізоляції від інших віртуальних машин, що працюють на тому ж комп'ютері. Очевидно, що вони мають бути ізольовані одна від одної. Попри логічну ізоляцію існує проблема доступу до спільних ресурсів (пам'яті, дискового простору). Через що виникає проблема крос-VM атак.

VM escape. Це ситуація, коли зловмисник обходить систему управління VM [3]. В цьому випадку зловмисник отримає доступ до інших VM, що може спричинити також неавторизований доступ до файлів на жорстких дисках. До таких вразливостей в основному схильні системи
IaaS [4].

VM migration. Під час міграції весь інформаційний контент VM стає відкритим при передачі даних по мережі [5]. На додачу модуль міграції може бути скомпрометований атакуючим зловмисником для переміщення VM на сторонній сервер. Тому критично важливим є виконання операції міграції VM з дотриманням всіх заходів безпеки.

Безпечне управління образами забезпечується за допомогою спеціально розробленого фреймворку, згідно якого кожну операцію може виконувати тільки авторизований користувач. Крім того рекомендується використовувати журналювання всіх операцій.

Література

1. F. Sabahi, “Secure Virtualization for Cloud Environment Using Hypervisor-based Technology,” Int. Journal of Machine Learning and Computing, vol. 2, no. 1, 2012.
2. S.-F. Yang, W.-Y. Chen, and Y.-T. Wang, “ICAS: An inter-VM IDS Log Cloud Analysis System,” in 2011IEEE International Conference on Cloud Computing and Intelligence Systems, 2011, pp. 285–289.
3. S. L. and Z. L. and X. C. and Z. Y. and J. Chen, S. Luo, Z. Lin, X. Chen, Z. Yang, and J. Chen, “Virtualization security for cloud computing service,” in International Conference on Cloud and Service Computing (CSC), 2011, pp. 174–179.
4. M. Ibrahim, A.S. and Hamlyn-Harris, J. and Grundy, J. and Almorsy, “CloudSec: A security monitoring appliance for Virtual Machines in the IaaS cloud model,” in 5th International Conference on Network and System Security (NSS), 2011, pp. 113–120.
5. J. Sedayao, S. Su, X. Ma, M. Jiang, and K. Miao, “A Simple Technique for Securing Data at Rest Stored in a Computing Cloud,” in Proceedings of the 1st International Conference on Cloud Computing, 2009, pp. 553–558.