Реалізація механізму Port Knocking як дієвого способу боротьби з brute-force атаками

Хошаба О.М.

Доцент кафедри захисту інформації, Вінницький національний технічний університет
вул. Хмельницьке шосе 21, м. Вінниця, Україна, khoshaba@mail.ru

***Анотація* — В роботі проаналізовано механізм Port Knocking, показана реалізація програмного засобу на стороні серверу та клієнта. Наведені приклади створення конфігураційного файлу. Вказано на можливі варіанти обходу механізму Port Knocking та способи уникнення цих дій.**

Ключові слова: Механізм Port Knocking, brute-force атаки, настройки файєрвола, конфігураційний файл knockd, програмні клієнти knock.

The Port Knocking mechanism of implementation as an effective way to fighting with brute-force attacks

Khoshaba O.M.

Ass.Prof., Department of information security, Vinnitsya National Technical University
Khmelnytsky Shose str., 21, Vinnytsya, Ukraine, khoshaba@mail.ru

***Abstract* — The paper analyzes the mechanism of Port Knocking, shows the implementation of the software on the server side and the client. These examples create the configuration file. Specified on the possible circumvention mechanism Port Knocking and how to avoid these actions.**

Keywords: the Port Knocking Mechanism, brute-force attack, firewall settings, configuration file knockd, software clients knock.

Актуальність теми

Дієвим елементом отримання несанкціонованого доступу до віддалених інформаційних ресурсів є broot-force атака.

Метод brute force (або повний перебір, «грубої сили») [1] — метод рішення криптографічної задачі шляхом перебору всіх можливих варіантів ключа. Складність повного перебору залежить від кількості всіх можливих рішень задачі. Так, якщо в паролі можуть використовуватися 36 різних символів (латинські літери одного регістру та цифри), а швидкість перебору становить 100 000 паролів в секунду то час знайдення паролю за допомогою повного перебору наведений у табл. 1.

Таблиця 1 – Час повного перебору в залежності від кількості символів [1]

Кількість символів	Кількість варіантів	Час перебору
1	36	менше секунди
2	1296	менше секунди
3	46 656	менше секунди
4	1 679 616	17 секунд
5	60 466 176	10 хвилин
6	2 176 782 336	6 годин
7	78 364 164 096	9 днів

Однак сучасні персональні комп'ютери можуть перевіряти зі швидкістю понад сто мільйонів паролів в секунду, використовуючи утиліти для злому паролів (наприклад John the Ripper [3]), запущених на CPU і мільярди паролів в секунду при використанні утиліт, що використовують GPU [2].

Також коли персональні комп'ютери користувачів об'єднуються для злому (в так звані ботнетах), можливості злому пароля значно розширюються. Ще на початку 2000 року відома мережа користувачів distributed.net [4] успішно підібрало 64-бітний ключ RC5 за 4 роки використовуючи більш ніж 300000 різних комп'ютерів в різний час. Генерація пароля проходила в середньому понад 12 мільярдів ключів в секунду. З 2011 комерційні продукти мають можливість тестування до 2,800,000,000 паролів в секунду на персональному комп'ютері з використанням потужного графічного процесора. В даний час обчислювальний пристрій може зламати 10-символьний пароль в одному регістрі витративши менше одного дня.

Головним у реалізації brute-force атаки є знайдення відкритих портів віддаленої інформаційної системи внаслідок чого згодом відбувається і сама атака. Тому підвищення об'ємів трафіку моніторингу відкритих портів IP-адрес з року в рік стає дедалі інтенсивнішими.

Рішення проблеми

Самими дієвим способом боротьби з brute-force атакою - є її недопущення шляхом приховання робочого віддаленого порту інформаційної системи.

Саме за досить ефективний алгоритм приховування, що не залишає ніяких шансів зловмисникам на знаходження справжнього сервісу, механізм Port Knocking можна вважати зірцем стеганографії.

Port Knocking - це мережевий захисний механізм, дія якого заснована на такому принципі. Існує сервіс який використовує мережевий порт що за замовчуванню є ззовні закритим.

Як тільки на мережевий інтерфейс надходить заздалегідь певна послідовність пакетів даних - порт відкритися (рис. 1).

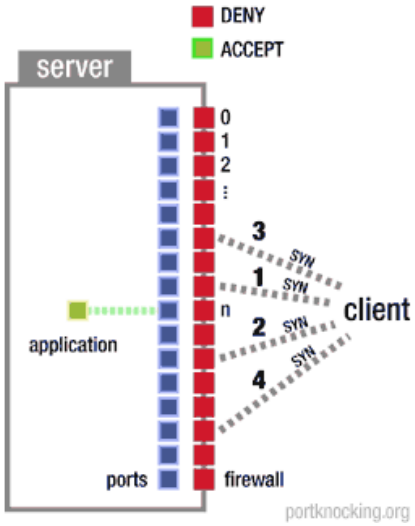

Рисунок 1 – Організація дії механізму Port Knocking [5]

На рис. 1 показана різна послідовність відсилання syn пакетів на зовнішній інтерфейс що контролюється файєрволом (червоних колір), відкриті порти що знаходяться за файєрволом (синій колір) та відповідні сервіси у вигляді прикладних програм що «прослуховують» визначені порти (зелений колір). Найчастіше використовується порт SSH який із зовнішнього боку інтерфейсу є невидимим.

Реалізація механізму Port Knocking

Реалізація механізму Port Knocking виконується в три етапи. Перший етап полягає у визначені загальної роботи Port Knocking: залучені порти, протоколи та довжини пауз між відсиланням syn пакетів (рис. 1). Другий та третій етапи пов'язані з налаштуванням серверної та клієнтських частин.

Налаштування серверної частини механізму Port Knocking

Для реалізації серверної частини механізму Port Knocking найбільш часто використовують демон knockd. Працюючи в режимі демона спільно з файєрволом iptables, knockd прослуховує мережевий інтерфейс, чекаючи коректного алгоритму послідовності запитів на підключення у вигляді syn пакетів (рис. 1). Як тільки демон knockd розпізнає коректну послідовність запитів, він виконує команду що знаходиться в файлі конфігурації knockd. Як правило - це виклик iptables, що дозволяє виконанню з'єднання на певний мережевий порт де знаходиться необхідний сервіс (прикладна програма). Наприклад, це може бути демон SSH що очікує вхідних підключень до 22 порту.

Однак, за встановленими правилами iptables може забороняти вхідні з'єднання на 22-й порт. Тоді, knockd, що прослуховує мережевий інтерфейс eth0, очікує послідовності з TCP протоколу syn послідовності пакетів, наприклад, на порти 9000, 6501 і 1234. Як тільки ця послідовність з'єднань буде виявлена, knockd за допомогою виклику iptables змінить правило файєрволу та дозволить підключення ззовні до 22-му TCP-порту на дану сесію.

В цьому випадку основні опції (що позначені у квадратних дужках) конфігураційний файл knockd (/etc/knockd.conf) буде виглядати наступним чином:

[openSSH]
sequence = 9000,6501,1234
seq_timeout = 5
command = /sbin/iptables -A INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags = syn
[closeSSH]
sequence = 1234,6501,9000
seq_timeout = 5
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags = syn

Значення параметра sequence визначає послідовність. Числа є номерами TCP-портів. Також, існує можливість вказати використання протоколів TCP або UDP за допомогою суфіксів tcp або udp, наприклад (рис. 2):

sequence = 3333:tcp,9999:udp,1010:udp,8675:tcp

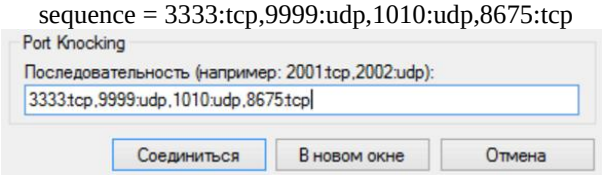

Рисунок 2 – Організація дії механізму Port Knocking [5]

На рис. 2 представлені значення параметрів для встановлення механізму Port Knocking на клієнті putty.

Значення параметра seq_timeout задає максимальний час в секундах, яке відводиться на вчинення клієнтом послідовності підключень. Якщо клієнт не вкладається в цей час - підключення буде відхилено. За допомогою позначення Interface

можна змінити мережевий інтерфейс який буде прослуховуватись демоном knockd:

Interface = eth1

Значення параметра command визначає шлях і параметри сервісу що викликається в разі виявлення правильного алгоритму.

Існує також можливість визначати інші прапорці транспортного протоколу TCP за допомогою параметра tcpflags які беруть участь в послідовності відправлення пакетів до віддаленої інформаційної системи. В цьому випадку декілька прапорців необхідно розділяти комою:

tcpflags = syn, ack, urg

Для явного виключення окремих прапорців потрібно використовувати знак оклику:

tcpflags = syn, !ack, urg

Іншим важливим варіантом конфігурації knockd є використання параметрів *start_command, cmd_timeout* і *stop_command*. Наприклад:

[OpenSSH]
one_time_sequences = /etc/knockd/smtp_sequences
seq_timeout = 15
tcpflags = fin, !ack
start_command = / usr / sbin / iptables -A input -s% IP% -p tcp --dport 25 -j ACCEPT
cmd_timeout = 5
stop_command = / usr / sbin / iptables -D INPUT -s% IP% -p tcp --dport 25 -j ACCEPT

Параметр *start_command* за змістом ідентичний параметру command. Значення параметра *cmd_timeout* визначає часовий інтервал в секундах. Після закінчення часу виконується команда, що прописана у значенні параметра *stop_command*. Таким чином, здійснюється можливість відкриття певного порту на довільний проміжок часу.

Для автоматичного запуску демон knockd під час старту інформаційної системи, в конфігураційному файлі /etc/default/knockd прописується:

START_KNOCKD=1

НАЛАШТУВАННЯ КЛІЄНТСЬКОЇ ЧАСТИНИ МЕХАНІЗМУ PORT KNOCKING

Після встановлення серверної частини механізму Port Knocking необхідно виконати тестування за допомогою сервісу telnet та прикладних програмних засобів putty для ОС Windows (рис. 2) або knock (програмний засіб що входить до складу knockd, для ОС Unix), наприклад:

knock 192.168.1.100 3333:tcp 9999:udp 1010:udp 8675:tcp

Складність може бути присутня під час використання певних прапорців, відмінних від syn. В цьому випадку, необхідно використовувати інші програмні засоби, такі як packit, SendIP [6].

Клієнтські частини механізму Port Knocking являють собою різноманітні програмні засоби що працюють з різними операційними системами: Unix, Windows, Android, iOS. Для Unix, та Windows — поширено використовуються вищезгадані програмні засоби. Для iPhone поширеними є Port Knock Lite [7] та KnockOnD [8], для Android відомий клієнт knock-android [9].

ВИСНОВКИ

Механізм Port Knocking дозволяє виконувати приховування відкритих портів віддаленої інформаційної системи. Наявна можливість використання різних протоколів транспортного рівня стеку TCP/IP з визначенням довільних прапорців, проміжку часу між пакетами та довільною кількістю самих пакетів дозволяє побудувати гнучкий алгоритм розпізнавання санкціонованого доступу до інформаційних ресурсів серверу. Звісно, існують способи перехоплення послідовності пакетів у мережі та інші шляхи обходу цього захисту кваліфікованими користувачами, але використання механізму Port Knocking унеможливлює проведення brute-force атаки для великої частини зловмисників. Також дієвим способом боротьби з перехопленням правильних послідовностей пакетів у мережі є генерація хибних послідовностей, використання параметру one_time_sequences у конфігураційному файлу демона knockd, використання криптоалгоритмів аутентифікації, тощо.

[1] Метод «грубої сили» – Режим доступу: https://uk.wikipedia.org/wiki/Метод_«грубої_сили» – Назва з екрану.

[2] Взлом пароля – Режим доступу: https://ru.wikipedia.org/wiki/Взлом_пароля – Назва з екрану.

[3] John the Ripper – Режим доступу: https://ru.wikipedia.org/wiki/John_the_Ripper – Назва з екрану.

[4] John the Ripper – Режим доступу: https://ru.wikipedia.org/wiki/John_the_Ripper – Назва з екрану.

[5] Port Knocking – Режим доступу: http://www.portknocking.org/ – Назва з екрану.

[6] SendIP – Режим доступу: http://snad.ncsl.nist.gov/ipv6/sendip.html – Назва з екрану.

[7] Port Knock – Режим доступу: http://www.sungheroes.com/portknock/ – Назва з екрану.

[8] KnockOnD – Режим доступу: http://bluezbox.com/knockond.html – Назва з екрану.

[9] knock-android – Режим доступу: https://code.google.com/archive/p/knock-android/ – Назва з екрану.