

Технически университет

София

**Програмни технологии за сигурен код**

R.U.D.Y (R-U-Dead-Yet) attack

Изготвил: **Денис Заимов**

**501214031**

**51гр.**

1. Какво представляват DoS(denial-of-service) атаките?

[1] Атака за отказ на услуга е опит даден ресурс, предоставян от компютър/сървър (наричан-жертва), да бъде направен недостъпен за целевите му потребители. Атаката може да бъде чрез изтощаване на ресурси или чрез възползване от грешка в софтуера на жертвата.

Най-често биват атакувани популярни уеб сървъри, като целта е те да станат недостъпни. Според Борда на архитектите на Интернет, това действие е компютърно престъпление, нарушаващо Етиката в Интернет.

Мотивите за DoS атаките мога да бъдат различни, като някои от най-често срещаните са:

* Хактивизъм;
* Кибер вандализъм;
* Изнудване;
* Личностно съревнование;
* Бизнес интереси;
* Кибервойна.

1. Видове DoS атаки

[2] DoS атаките могат да се класифицират в две основни групи:

* Application layer attacks (насочени към приложни протоколи): Най-често тяхното действие се свежда до генериране на голям обем заявки към даден сървър или създаване на такива с грешен формат, водещи до значително увеличаване на работното натоварване на целта.
* Network layer attacks (насочени към мрежовите протоколи): В тази група са включени атаки, насочени към протоколи като TCP, UDP и такива, свързани с маршрутизирането на трафика. Много често при генерирането на голям обем от мрежови пакети, дадени устройства могат да спрат своето функциониране.

1. Distrubuted DoS (DDoS) атаки и каква е разликата с DoS атаките?

[2] За разлика от DoS атаките, които най-често се извършват от едно единствено атакуващо устройство , при DDoS генерираните злонамерени действия стартират от множество отдалечени спрямо целта хостове. Това води до значително увеличаване на обема на генерирания трафик. Много често в атаките се включват т.нар. зомбите или ботове – заразени системи, на които работи зловреден код. След неговото активиране започва и генерирането на съответните мрежови пакети. Възможно е броя на пакетите да надхвърля 25 милиона за 1 секунда. Една от най-известните DDoS атаки е R.U.D.Y (R-U-Dead-Yet).

1. Какво представлява атаката R.U.D.Y (R-U-Dead-Yet)?

[2, 6, 5] Атаката получава своето име от едноименния албум на финландската метъл група Children of Bodom. По същество действието се свежда до генериране на прекалено дълги данни за полета за въвеждане на информация при WEB сайтове. Скоростта на атаката е ниска и много често тя се описва с термина “low and slow”. Атаката отваря едновременни POST HTTP връзки към HTTP сървъра и забавя изпращането на тялото на заявката POST до точката, в която сървърните ресурси са наситени. Целта на бавната работа е да се задържат максимално дълго време отворени съответните HTTP комуникационни сесии. Най-често това злонамерено действие се активира от множество зомби-системи, но е възможно успешно да се реализира и от едно единствено устройство.

1. Методи за предпазване от атаката R.U.D.Y

[3,7]Low-and-Slow атаки като R.U.D.Y. са сравнително трудни за откриване, в сравнение с обемните DDoS атаки, при които има нестабилност на входящия трафик. Поради трудността за откриването на атаката може да бъдат въведени защити, които да я предотвратят.

За да се предотврати тази атака се препоръчва да се предприемат следните действия:

* Отказ или прекратяване на връзките чрез HTTP методи за които не съществуват съответните URL/ендпойнти.
* Ограничаване на броя на символите в header-a и в тялото на заявката до необходимия минимум. Задайте по-строги ограничения за URL адресите, които да са подходящи за всеки ресурс, който приема тялото на заявката.
* Задайте конкретно време за свързване към сървъра, ако е възможно. Разбира се, ако времето за изчакване е твърде кратко, рискувате да изпуснете легитимни бавни връзки; и ако е твърде дълго, нямате защита от атаки. Препоръчва се времето за изчакване да се избира въз основа на статистическите данни за времетраенето на връзката, напр. времето за изчакване, което да бъде малко по-голямо от средното, това трябва да задоволи повечето истински клиенти.
* Backlog-a на чакащите връзки позволява на сървъра да поддържа връзки, които не са готови да бъдат приети, и това му позволява да издържи на по-голяма бавна HTTP атака, както и да даде възможност на законните потребители да работят дори при по голямо натоварване. Въпреки че големия backlog също така удължава атаката, то той задържа всички заявки независимо от това дали са легитимни потребители или не. Ако сървъра подържа backlog, то се препоръчва да се направи достатъчно голям, че поне да устои на по-малка атака.
* Определете минималната скорост на входните данни и прекъснете връзките, които са по-бавни от тази скорост. Трябва да се внимава да не бъде твърде нисък минимума, защото по този начин се рискува да се отхвърлят законни връзки.

1. Препоръки за справяне с R.U.D.Y атака специфични за сървърите [7]

* Apache:
* Използването на директивите <Limit> и <LimitExcept> за отказване на заявки с методи, за които няма URL, няма да помогне, защото Apache чака цялата заявка да бъде завършена, преди да приложи тези директиви. Затова трябва да се използват LimitRequestFields, LimitRequestFieldSize, LimitRequestBody, LimitRequestLine, LimitXMLRequestBody заедно с тях.
* Задайте разумни стойности за TimeOut и KeepAliveTimeOut. Стойността по подразбиране от 300 секунди за TimeOut е „overkill“ в повечето ситуации.
* Може да се увеличи стойността на ListenBackLog (по подразбиране тя е 511), защото това би помогнало, когато сървърът не може да приеме опити за връзки достатъчно бързо.
* Може да се увеличи стойността на MaxRequestWorkers, за да позволите на сървъра да се справи с голям брой едновременни опити за връзки .
* Регулирайте директивата AcceptFilter (поддържа се от FreeBSD и Linux) и включете специфичните оптимизации за операционната система за слушане на socket-а за определен тип проткол.

Редица модули на Apache са достъпни за ограничаване на заплахата от бавни HTTP атаки. Например, директивата RequestReadTimeout на mod\_reqtimeout помага да се контролират бавните връзки чрез задаване на изчакване и минимална скорост за получаване на заявки.

* Nginx
* Проверявайте $request\_method променливата, за да се ограничат приетите заявки.
* Задайте по-малки стойности за client\_max\_body\_size, client\_body\_buffer\_size, client\_header\_buffer\_size, large\_client\_header\_buffers и ги увеличавайте само при нужда.
* Задайте по-малки стойности за client\_body\_timeout, client\_header\_timeout.
* Използвайте HttpLimitReqModule и HttpLimitZoneModule, за да ограничите броя на заявките или броя на едновременните връзки за дадена сесия към един и същи адрес.
* Конфигурирайте worker\_processes и worker\_connections спрямо броя на процесорите/ядрата и съдържанието. Формуалата за конфигурацията е макс.бр.клиент = процеси \* връзки.
* Lighttpd
* Ограничете методите на заявките, като използвате полето $ HTTP ["request-method"] в конфигурационния файл на основния модул.
* Използвайте server.max\_request-size, за да ограничите размера на цялата заявка, включително header-a.
* Задайте по-малки стойности на server.max-read-idle, така че сървърът да затваря бавните връзки.
* IIS 6
* Задайте стойности на свойствата connectionTimeout, HeaderWaitTimeout и MaxConnections, за да намалите въздействието на бавните HTTP атаки.
* IIS 7
* Ограничете заявките ката използвата атрибутите maxAllowedContentLength, maxQueryString, и maxUrl в <RequestLimits>.
* Задайте <headerLimits> за да конфигурирате типа и размера на header-a, който сървъра ще приема.
* Настройте атрибутите connectionTimeout, headerWaitTimeout и minBytesPerSecond, които са част от <limits> и <WebLimits>, за да намалите въздействието на бавните HTTP атаки.

1. Примерен скрипт на атака R.U.D.Y. на Python

Може да бъде отрит във файла *r-u-dead-yet-v2.2.py.* За да се изпълни скрипа, трябва да има инсталиран Python.

Примерна команда за изпълнение на скрипта:

*./r-u-dead-yet.py <URL към който ще се отправи атаката>* (скрипа ще отвори меню от което може да се избере формата която ще се атакува, както и кое поле от нея и броя на едновременните заявки)

Връзка към скрипта: <https://github.com/loganhasson/r-u-dead-yet>

1. Източници
   1. <https://bg.wikipedia.org/wiki/%D0%90%D1%82%D0%B0%D0%BA%D0%B0_%D0%B7%D0%B0_%D0%BE%D1%82%D0%BA%D0%B0%D0%B7_%D0%BD%D0%B0_%D1%83%D1%81%D0%BB%D1%83%D0%B3%D0%B0>
   2. <http://eh.ict-academy.bg/index.php?page=%D1%81%D1%8A%D0%B4%D1%8A%D1%80%D0%B6%D0%B0%D0%BD%D0%B8%D0%B5>
   3. <https://www.incapsula.com/ddos/attack-glossary/rudy-r-u-dead-yet.html>
   4. <https://en.wikipedia.org/wiki/R-U-Dead-Yet>
   5. <https://security.radware.com/ddos-knowledge-center/ddospedia/rudy-r-u-dead-yet/>
   6. <https://blog.qualys.com/securitylabs/2011/11/02/how-to-protect-against-slow-http-attacks>