

RedisSentry: защищаем python web-сервер от подбора пароля на примере django

Максимов Лев Викторович

9 Jul 2012



<http://www.devconf.ru>

Django administration

Username:

Password:

~7000 шт/час 1 IP, 1 процесс

Способы борьбы с перебором пароля

- **Captcha**, (Completely Automated Public Turing test to tell Computers and Humans Apart)
 - + отпугивает не особо заинтересованных атакующих
 - отпугивает и обычных пользователей тоже
 - + скрипт из пары строчек уже не сработает
 - есть специализированные программы-распознаватели и люди-распознаватели
- **Throttling**, задержка ответа сервера
 - + несколько снижает эффективность атаки
 - может раздражать пользователя
 - сложности реализации на синхронном сервере
- **Блокировки**, не допускает к аутентификации в течение некоторого времени
 - + практически неощутимо для пользователя
 - + более эффективно сдерживает атаки

Модули защиты от перебора пароля:

	first commit - last commit	author	storage	счетчики
django snippet #1083	24 Sep 2008 - 24 Sep 2008	jensbreit	main db	global <i>либо</i> IP
ratelimitcache	7 Jan 2009 - 24 Sep 2009	Simon Wilson	memcached	IP <i>либо</i> IP+username
django-brutebuster	30 Nov 2009 - 6 Mar 2011	(company)	main db	IP & username
django-axes	1 Oct 2010 - 18 May 2012	Josh VanderLinden	main db	IP
django-failedloginblocker	12 Jan 2011 - 3 Mar 2011	Alex Kuhl	main db	username
django-lockout	24 Jul 2011 - 18 Oct 2011	Brian Jay Stanley	memcached	IP
<i>django-redissentry</i>	9 Apr 2012 - 3 May 2012	me	redis	...



Модули защиты от перебора пароля: способы интеграции

	storage	счетчики	способ интеграции
django snippet #1083	main database	global <i>либо</i> IP	decorator
ratelimitcache	memcached	IP <i>либо</i> IP+username	decorator
django-brutebuster	main db	IP & username	auth
django-axes	main db	IP	login
django-failedloginblocker	main db	username	auth
django-lockout	memcached	IP	auth
<i>django-redissentry</i>	redis	...	auth

Особенности реализации

- /django-axes/ успешный вход сбрасывает счетчик:
 - 4*admin, 1*hacker, 4*admin, ...

Уязвимости реализации (устранимые)






	storage	счетчики
django snippet #1083	main database	global <i>либо</i> IP
ratelimitcache	memcached	IP <i>либо</i> IP+username
django-brutebuster	main db	IP & username
django-axes	main db	IP
django-failedloginblocker	main db	username
django-lockout	memcached	IP
<i>django-redissentry</i>	redis	...



Особенности реализации

- /django-axes/ успешный вход сбрасывает счетчик:
 - 4*admin, 1*hacker, 4*admin, ...
- /main db/ скорость обработки отказа в аутентификации => DDOS










Уязвимости реализации (устранимые)

	storage	счетчики
django snippet #1083	main db 	global <i>либо</i> IP
ratelimitcache	memcached	IP <i>либо</i> IP+username
django-brutebuster	main db 	IP & username
django-axes 	main db 	IP
django-failedloginblocker	main db 	username
django-lockout	memcached	IP
<i>django-redissentry</i>	redis	...

Особенности реализации

- /django-axes/ успешный вход сбрасывает счетчик:
 - 4*admin, 1*hacker, 4*admin, ...
- /main db/ скорость обработки отказа в аутентификации => DDOS
- /username/ возможность неограниченного переполнения БД

Уязвимости реализации (устранимые)

	storage	счетчики
django snippet #1083	main db 	global <i>либо</i> IP
ratelimitcache	memcached	IP <i>либо</i> IP & username 
django-brutebuster	main db 	IP & username 
django-axes 	main db 	IP
django-failedloginblocker	main db 	username 
django-lockout 	memcached	IP
django-redissentry	redis	...

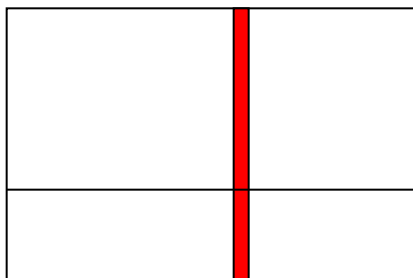
Архитектура модуля

	1-to-1	1-to-m	m-to-1	
IP-address	✓	✓	✗	проxy, NAT =>блок. <i>целая</i> подсеть
IP-address; USER_AGENT	✓	✗	✓	одно блок в браузер аккаунта
(IP-address, username) X_FORWARDED_FOR	✓	✗	✗	1M X 1M = 1Tинтерф.
IP-address; username	✓	✓	✓	1M + 1M = 2M; подсеть, владелец
IP-address; username; whitelist	✓	✓	✓	много счетчиков

Архитектура модуля

- IP-address:

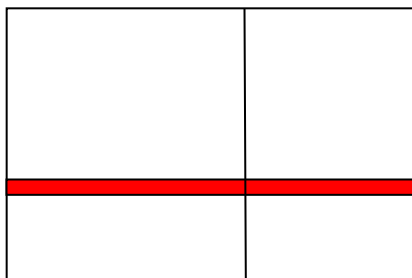
1.2.3.4



joe@me.ru

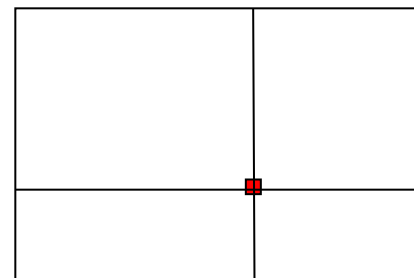
- username

1.2.3.4



- (IP,username)

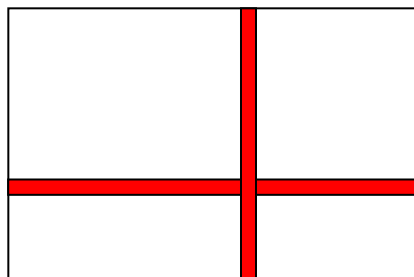
1.2.3.4



$6 \cdot 10^{77}$

- IP-address; username

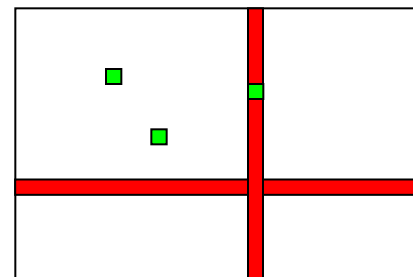
1.2.3.4



joe@me.ru

- IP-address; username; whitelist

1.2.3.4










joe@me.ru

$\sim 10^6$

13

Уязвимости архитектуры (неустранимые)

		1-to-1	1-to-m	m-to-1
django snippet #1083	global <i>либо</i> IP	+	+	
ratelimitcache	IP <i>либо</i> IP+username	+	+	
django-brutebuster	IP & username	+		
django-axes	IP	+	+	
django-failedloginblocker	username	+		+
django-lockout	IP	+	+	
<i>django-redissentry</i>	...	+	+	+

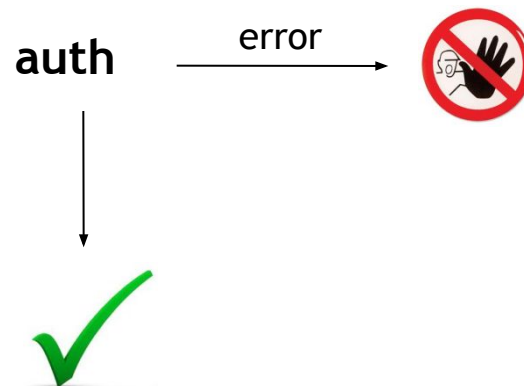
Цель работы

Создание надежного модуля защиты от брутфорса, который можно было бы использовать в production.

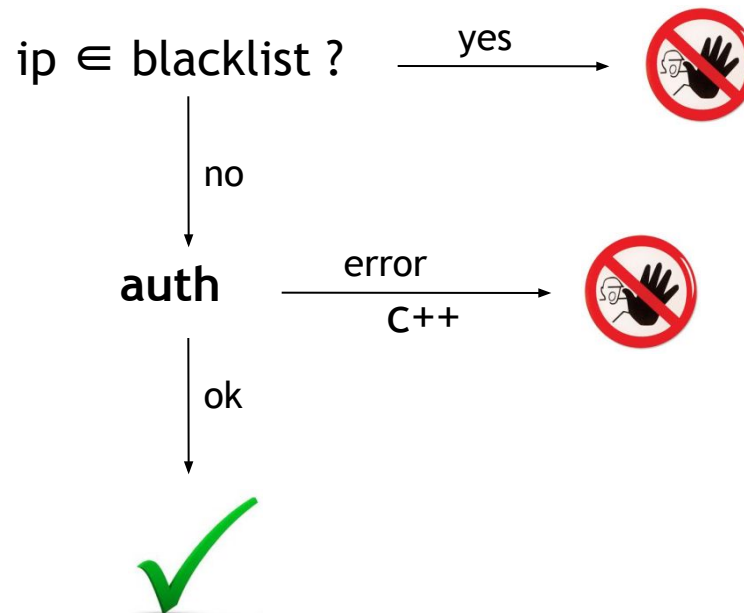
Подобрать набор фильтров, который бы одновременно:

- позволял защищать сервер от **всех перечисленных видов атак**
- **не блокировал лишний раз** пользователя только по той причине, что из-под его IP была атака и/или на его экаунт была атака
- не допускал возможности **переполнения** базы данных счетчиков.

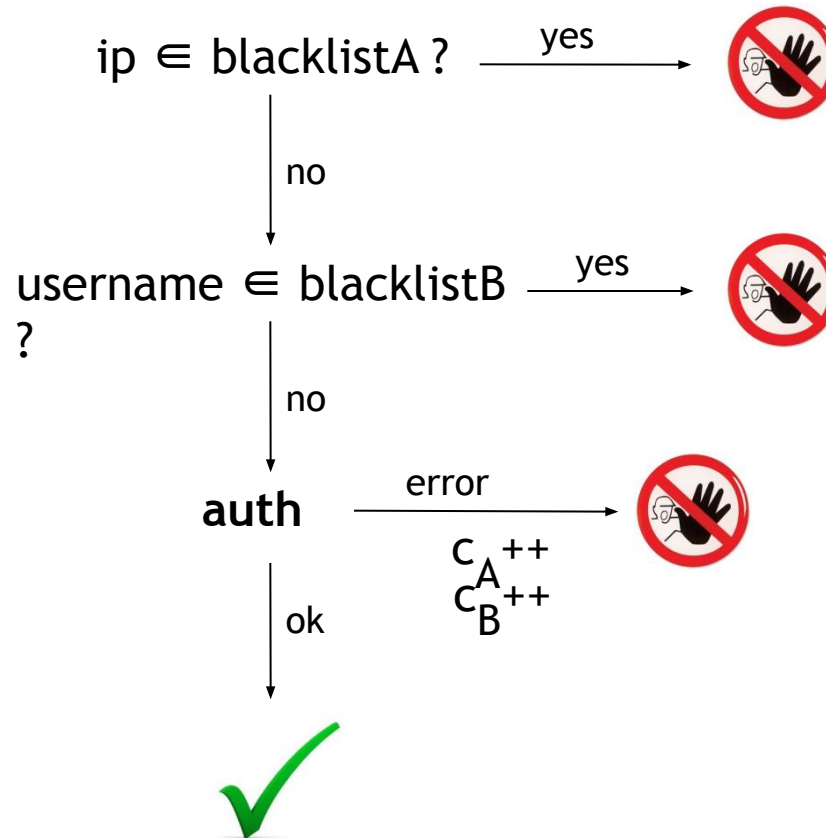
Блок-схема



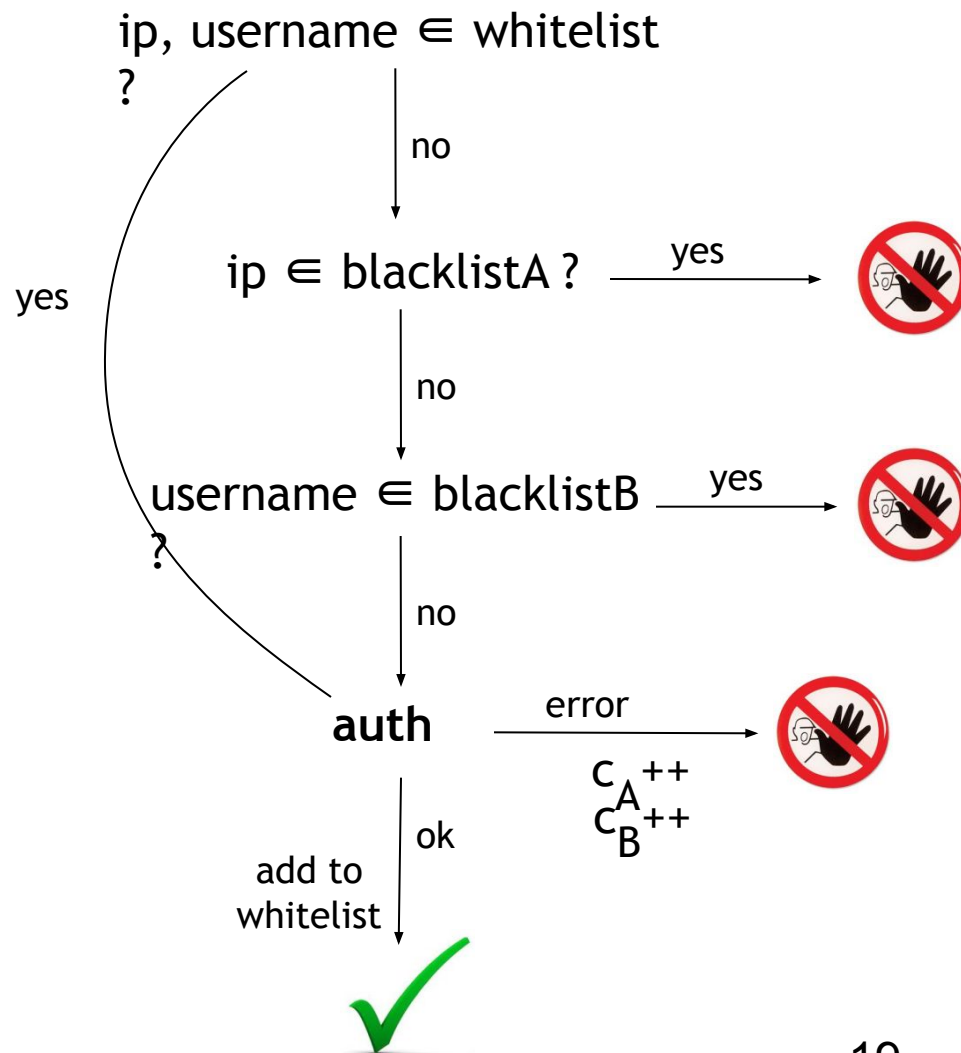
Блок-схема



Блок-схема



Блок-схема



Блок-схема

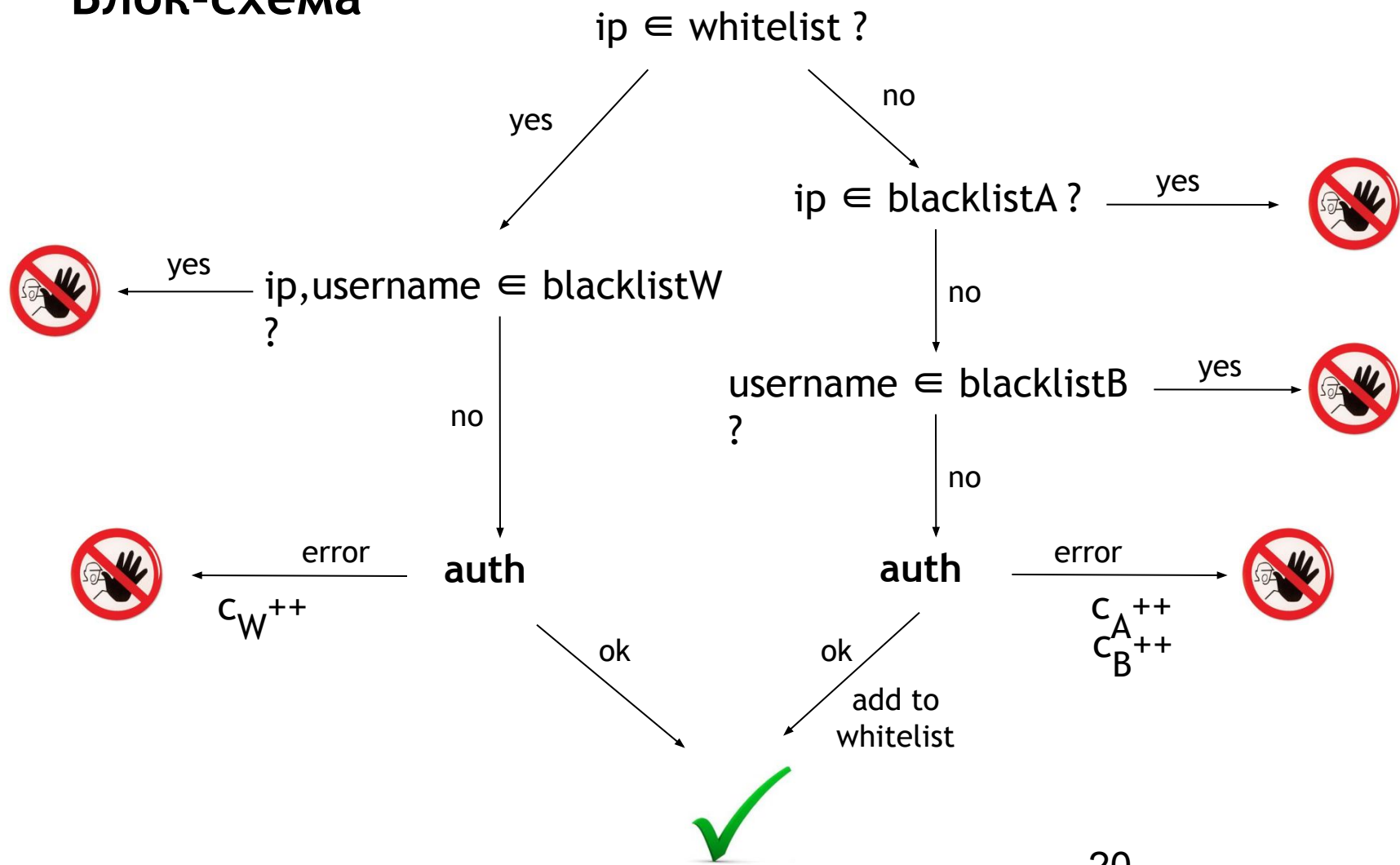


Диаграмма состояний - 1

Try again in 5 minutes

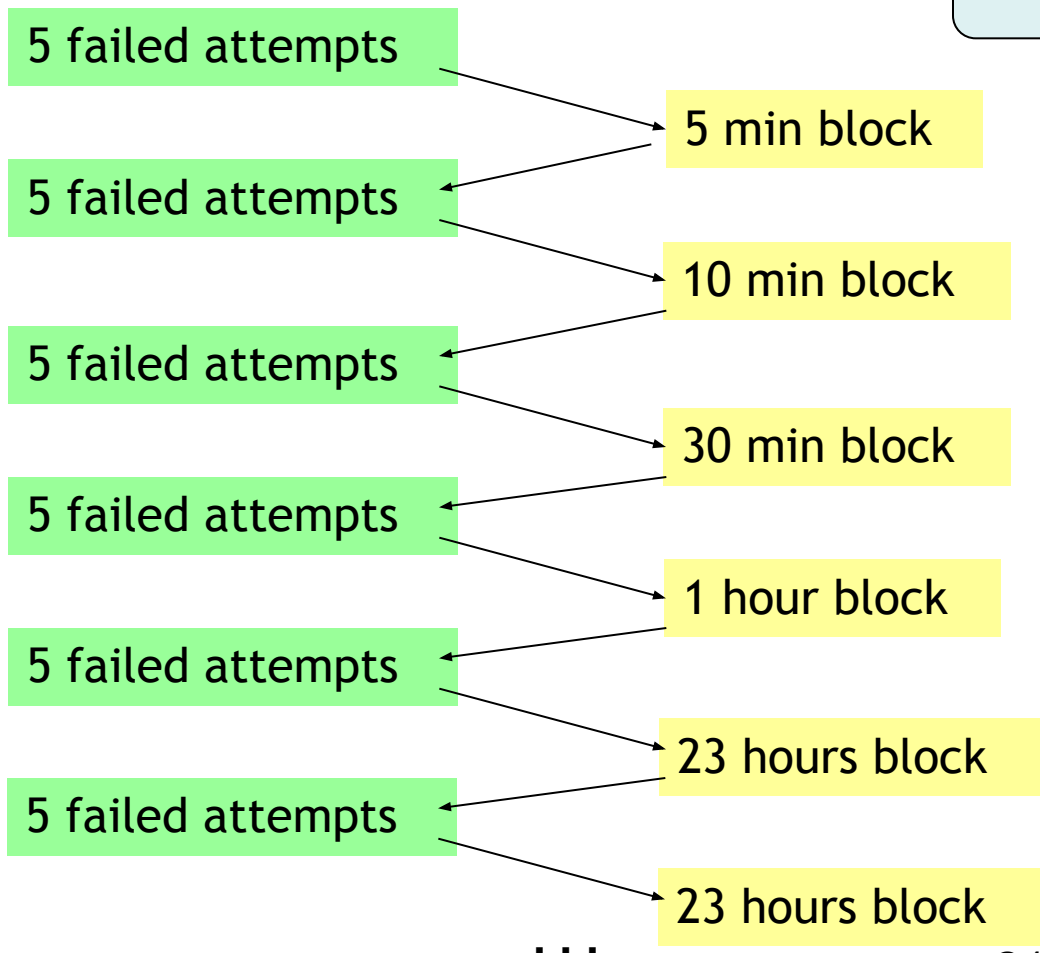
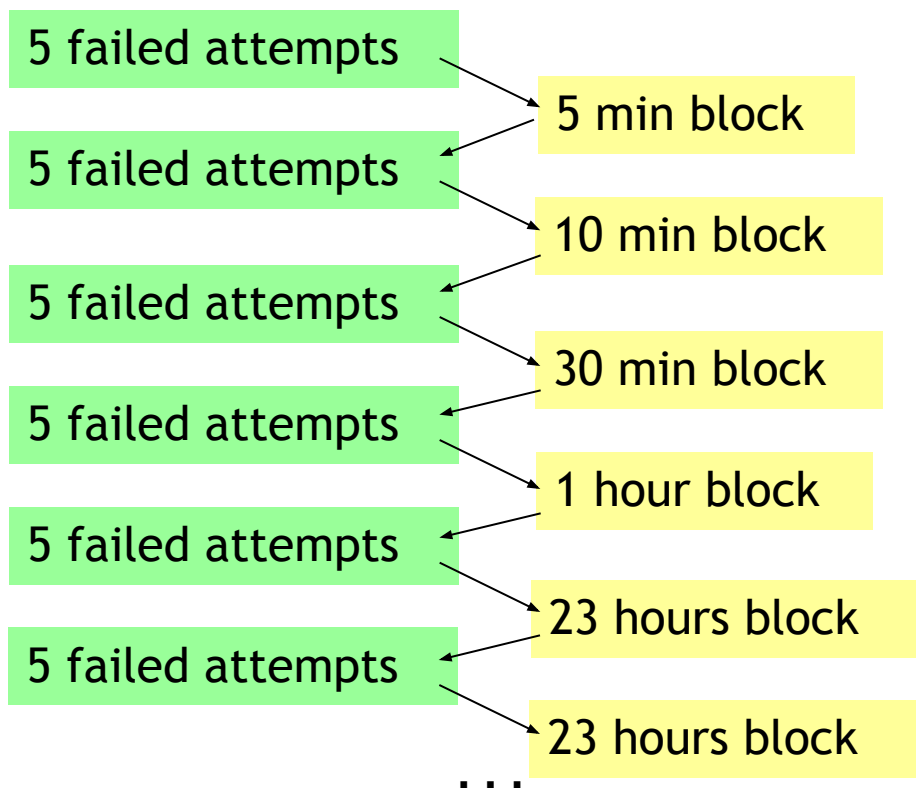


Диаграмма состояний - 2

из исходного состояния



из заблокированного состояния

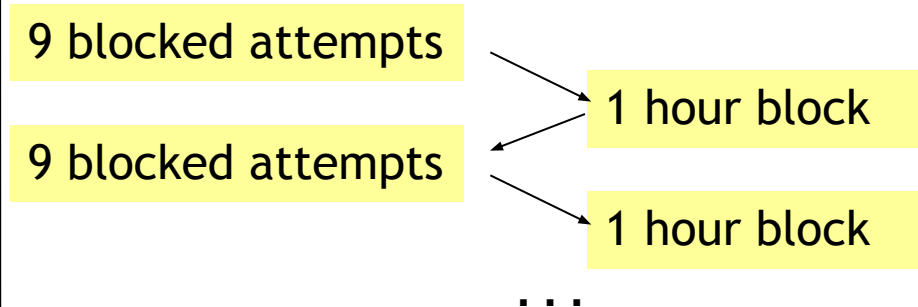
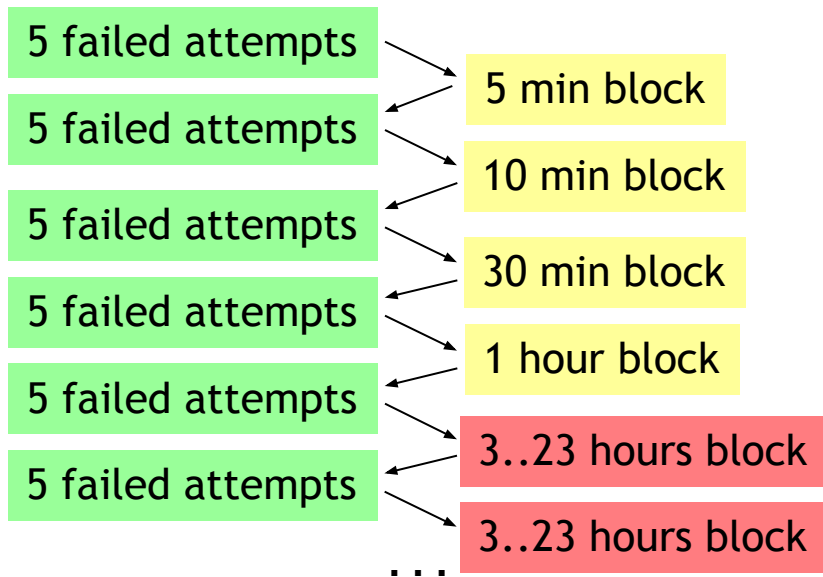


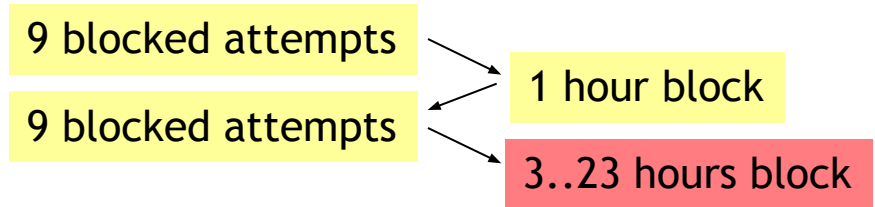
Диаграмма состояний - 3 (RedisSentryPlus)

из ИСХОДНОГО СОСТОЯНИЯ

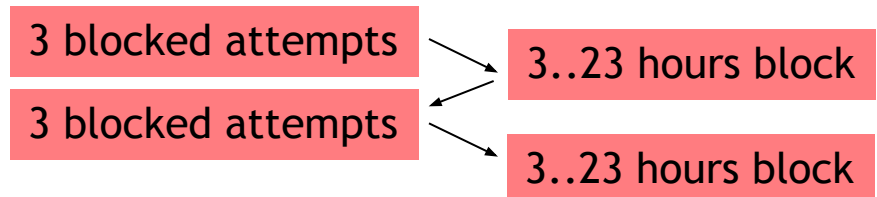


“Try again later”

из «явно» заблокированного состояния



из «неявно» заблокированного состояния



Счетчик whitelist'a

• IP: 193.124.10.53

Username: somebody@somewhere.com

15 bytes

~50 bytes

4*uchar 4 bytes

user_id ~8 bytes

/24 3 bytes

hash(username):

md5(username) 16 bytes

? 4 bytes

$$2^{32} = 4 \cdot 10^9$$

7*10⁹ (1 Nov 2011)

$$\log_{26} 2^{32} = 6.8$$

Hash-функции

• 4 bytes hash-functions:	100k	300k	
- additive/xor/rot	98k/100k/35	313k/316k/173	
- elf/snx/djb	364/5/0	2166/57/13	
- fnv/one-at-a-time	1/9	15/9	
- superfasthash/lookup3	18/10	18/10	
- murmur3	0	5	
- md5[:4]	1	9	
			pip install pyhash
		1m	2m
• 5 bytes hash-function: md5[:5]		0	4
			14m
• 6 bytes hash-function: md5[:6]		0	0

Счетчики

bytes	development	production	bytes
18	<i>Ac:ip</i>	a	5
18	<i>Ab:ip</i>	A . <u>ip</u> .	5
53	<i>Bc:username</i>	b	5
53	<i>Bb:username</i>	hash(<u>username</u>)	5
68	<i>Wc:ip:username</i>	c	8
68	<i>Wb:ip:username</i>	C . <u>ip</u> . <u>hash(username)</u>	8

Админка

The screenshot shows the Redis Sentry Dashboard administration interface. The browser window title is "Redis Sentry Dashboard | admin - Opera". The address bar shows "admin/redisentry/dashboard/". The page header includes "administration" and a welcome message for "Lev".

Redis Sentry Dashboard

Home > Redisentry > Dashboard

IP Counters

IP address	Failed attempts	Time left	Action
196.217.53.121	10	1 day, 17:06:46	remove

IP Blacklist

- empty -

Username Counters

- empty -

Username Blacklist

- empty -

Whitelist Counters

IP address	Username	Failed attempts	Time left	Action
41.237.74.55	nabho_voo@hotmail.fr	1	25 days, 5:52:12	remove
98.224.26.201	slabbi@brn.com	7	24 days, 1:46:20	remove

Whitelisted Users Blacklist

- empty -

Log file

```

2012-06-07 11:46:46 INFO 196.217.53.121 ninja_abach@hotmail.fr fa #1 from regular ip; ttl = 4:48:00
2012-06-07 11:47:00 INFO 196.217.53.121 ninja_abach@hotmail.fr fa #2 from regular ip; ttl = 9:36:00
2012-06-07 11:48:36 INFO 196.217.53.121 ninja_abach@hotmail.fr fa #3 from regular ip; ttl = 14:24:00
2012-06-07 11:48:43 INFO 196.217.53.121 ninja_abach@hotmail.fr fa #4 from regular ip; ttl = 19:12:00
2012-06-07 11:49:20 INFO 196.217.53.121 kaho_rohit@hotmail.fr fa #5 from regular ip, ip blocked for 5 min; ttl = 1 day, 0:00:00
2012-06-07 12:17:58 INFO 195.3.225.205 chelseal515@citromail user whitelisted
2012-06-07 12:36:28 INFO 196.217.53.121 kaho_rohit@hotmail.fr fa #6 from regular ip; ttl = 1 day, 4:48:00
2012-06-07 12:36:34 INFO 196.217.53.121 kaho_rohit@hotmail.fr fa #7 from regular ip; ttl = 1 day, 9:36:00
2012-06-07 12:36:43 INFO 196.217.53.121 kaho_rohit@hotmail.fr fa #8 from regular ip; ttl = 1 day, 14:24:00
2012-06-07 12:36:49 INFO 196.217.53.121 kaho_rohit@hotmail.fr fa #9 from regular ip; ttl = 1 day, 19:12:00
2012-06-07 12:36:59 INFO 196.217.53.121 kaho_rohit@hotmail.fr fa #10 from regular ip, ip blocked for 10 min; ttl = 2 days, 0:00:00
2012-06-07 13:08:16 INFO 41.218.243.64 cmaster1980@yahoo.com user whitelisted
2012-06-07 13:15:55 INFO 86.98.42.101 suhussain88@yahoo.com fa #1 from regular ip; ttl = 4:48:00
2012-06-07 13:15:57 INFO 41.218.243.64 cmaster1980@yahoo.com user whitelisted
    
```

Выбор инструментария

- почему nosql а не sql?
 - в основном доступ key-value
 - скорость
- почему redis, а не couchdb?
 - expire
- почему redis, а не mongodb?
 - expire
 - лучшая устойчивость к перезагрузкам
- почему redis, а не memcached?
 - удобные структуры данных
 - лучший контроль используемой памяти
 - масштабирование (шардинг)

Техническая информация

- Лицензия: MIT
- Исходный код:
 - <https://github.com/axil/redissentry-core>
 - <https://github.com/axil/django-redissentry>
- Тестовая инсталляция:
 - <http://redissentry.alwaysdata.net>

Установка и интеграция

```
pip install django-redissentry
```

- django:

```
MIDDLEWARE_CLASSES += 'redissentry.middleware.RequestMiddleware',  
INSTALLED_APPS += 'redissentry',
```

- flask:

```
def protected_auth(username, password):  
    sentry = RedisSentry(ip, username)
```

```
    msg = sentry.ask()  
    if msg != '':  
        raise Exception(msg)
```

```
    result = auth(username, password)
```

```
    msg = sentry.inform(bool(result))  
    if msg != '':  
        raise Exception(msg)
```

```
    return result
```

Заключение

- защита от атак: 1 IP-1 account; 1 IP-many accounts; many IP-1 account
- «гуманность» блокировки за счёт whitelist'a
- невозможность произвольного переполнения базы данных счетчиков
- крайне высокая скорость отказа в допуске к авторизации
- кумулятивное нарастание времени блокировки
- мелкие know-how:
 - простейший скрипт с фикс. временем задержки блокируется на всё время его работы;
 - эффективное время блокировки для атакующего больше, чем для обычного пользователя;
 - обработка попыток авторизоваться из заблокированного состояния;

Спасибо за внимание!









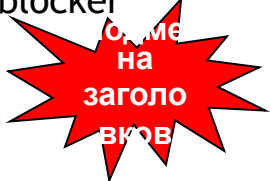
**RedisSentry: защищаем python web-сервер
от подбора пароля на примере django**

Максимов Лев Викторович

Архитектура модуля

	1-to-1	1-to-m	m-to-1	
IP-address	✓	✓	✗	проху, NAT => блок. <i>целая подсеть</i>
IP-адрес + USER_AGENT	✓	✓	✗	блок. <i>только 1 браузер</i>
IP-адрес + X_FORWARDED_FOR	✓	✓	✗	<i>ровно 1 сет.интерф.</i>
username	✓	✗	✓	заодно блок. <i>владелец экаунта</i>
(IP-address, username)	✓	✗	✗	$1M \times 1M = 1T$
IP-address; username	✓	✓	✓	$1M + 1M = 2M$; <i>подсеть, владелец</i>
IP-address; username; whitelist	✓	✓	✓	<i>много счетчиков</i>

Уязвимости реализации (устранимые)

django snippet #1083	storage	счетчики
	main db 	global <i>либо</i> IP
ratelimitcache	memcached	IP <i>либо</i> IP+username 
django-brutebuster	main db 	IP & username 
django-axes 	main db 	IP
django-failedloginblocker	main db 	username 
django-lockout 	memcached	IP
django-redissentry	redis	...