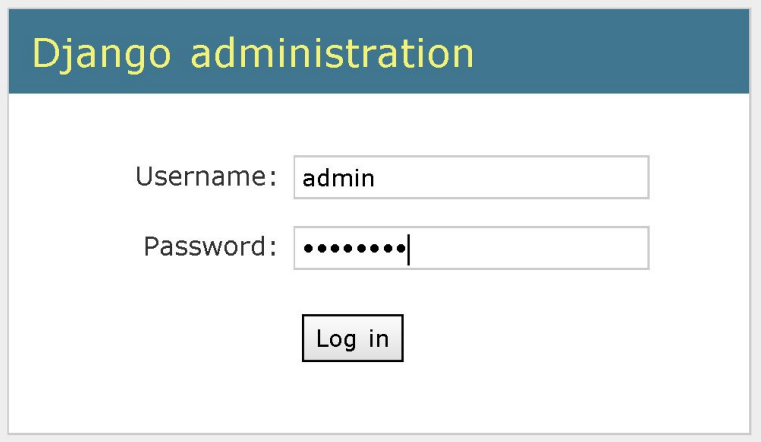
RedisSentry: защищаем python web-сервер от подбора пароля на примере django

Максимов Лев Викторович

9 Jul 2012



~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 обычних 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 *либо* IP
ratelimitcache	7 Jan 2009 - 24 Sep 2009	Simon Wilson	memcached	IP *либо* 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
django-redissentry	9 Apr 2012 – 3 May 2012	me	redis	...

Социальная сеть

взлом твиттера

Модули защиты от перебора пароля: способы интеграции

	storage	счетчики	способ интеграции
django snippet #1083	main database	global *либо* IP	**decorator**
ratelimitcache	memcached	IP *либо* 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**
django-redissentry	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 *либо* IP
ratelimitcache	memcached	IP *либо* 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	...

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 *либо* IP
ratelimitcache	memcached	IP *либо* 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	...

DD OS

DD OS

DD OS

DD OS

сброс счетчика

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 *либо* IP
ratelimitcache	memcached	IP *либо* 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	...

DDOS

DDOS

DDOS

DDOS

overflow

overflow

overflow

сброс счетчика

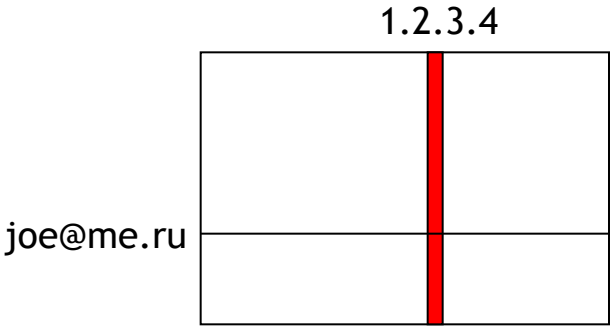
на заголовков

Архитектура модуля

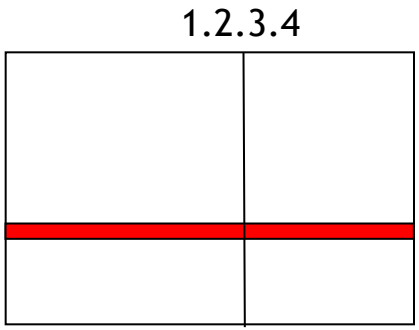
	1-to-1	1-to-m	m-to-1	
IP-address	✓	✓	✗	proxy, NAT =>блок. *целая* подсеть
~~IP-адрес + USER_AGENT~~	✓	✗	✓	~~блок. *подсеть* + браузер~~
~~(IP-address, username)~~	✓	✗	✗	~~1M x 1M = 1T~~
~~IP-адрес + X_FORWARDED_FOR~~	✓	✗	✗	~~*робот* + интерф.~~
IP-address; username	✓	✓	✓	1M + 1M = 2M; подсеть, владелец
IP-address; username; whitelist	✓	✓	✓	много счетчиков

Архитектура модуля

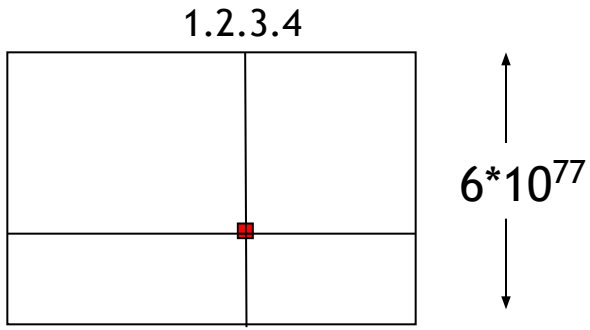
- IP-address:



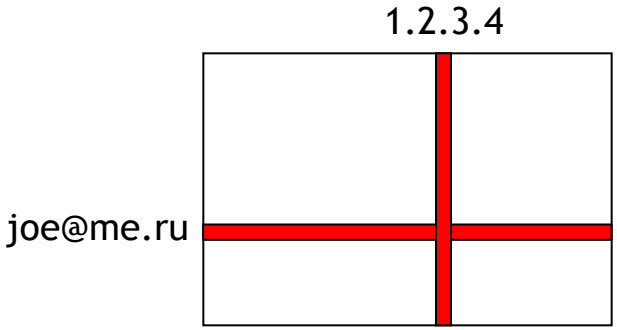
- username



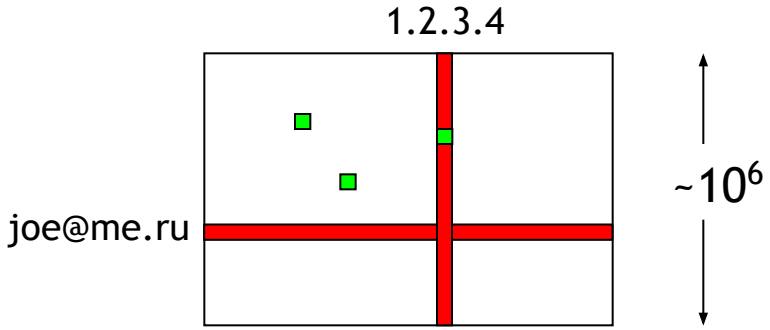
- (IP,username)



- IP-address; username



- IP-address; username; whitelist



Уязвимости архитектуры (неустранимые)

		1-to-1	1-to-m	m-to-1
django snippet #1083	global *либо* IP	+	+	
ratelimitcache	IP *либо* IP+username	+	+	
django-brutebuster	IP & username	+		
django-axes	IP	+	+	
django-failedloginblocker	username	+		+
django-lockout	IP	+	+	
django-redissentry	…	+	+	+

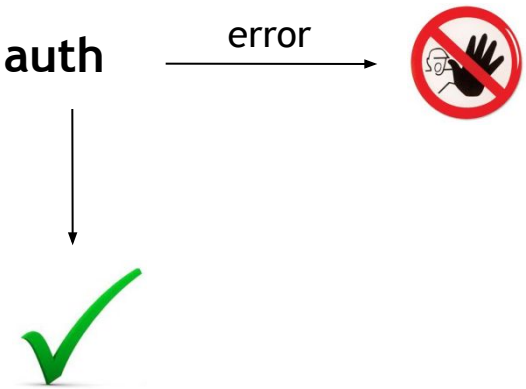
Цель работы

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

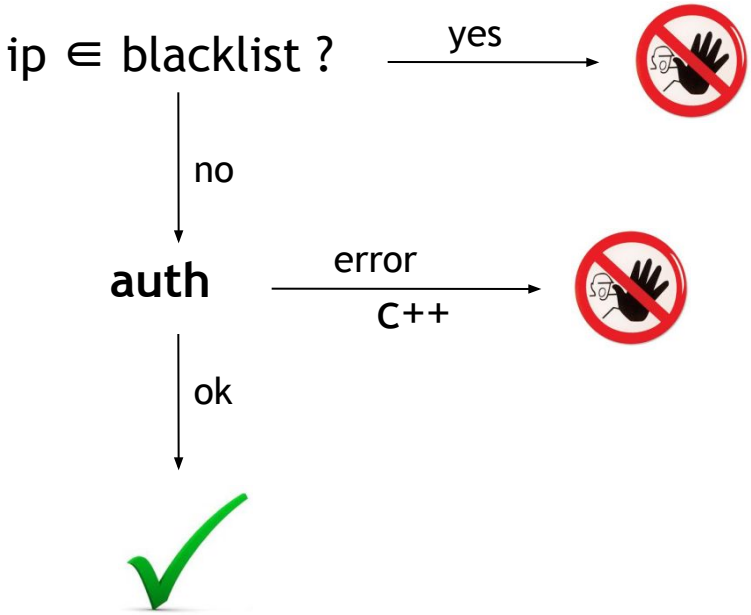
Подобрать набор фильтров, который бы одновременно:

- 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 эккаунт была атака
- не допускал возможности **переполнения** базы данных счетчиков.

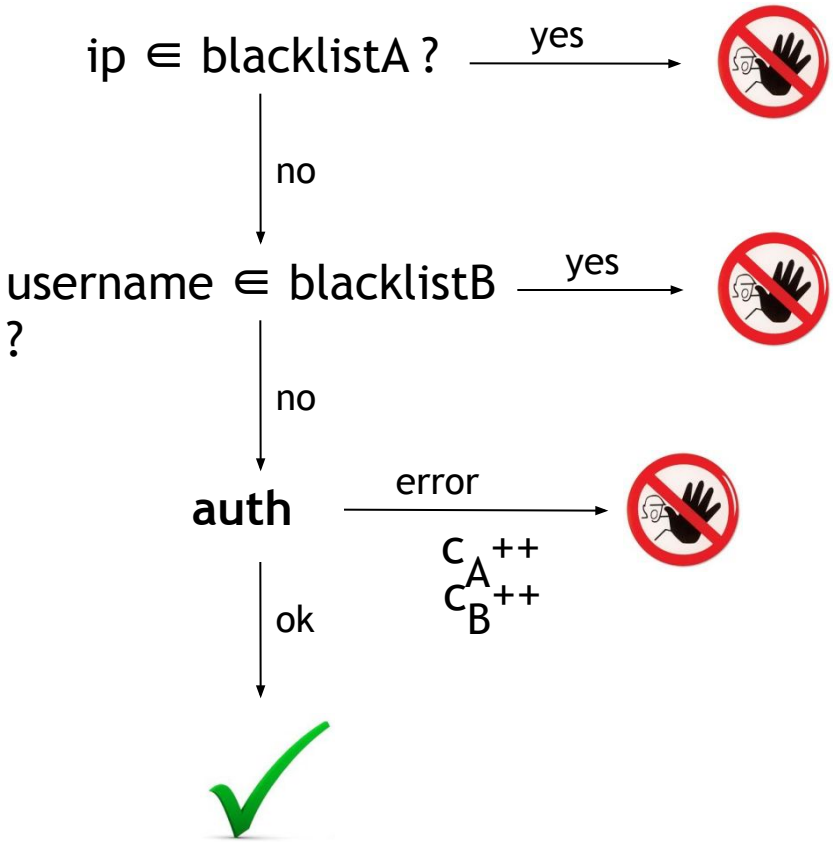
Блок-схема



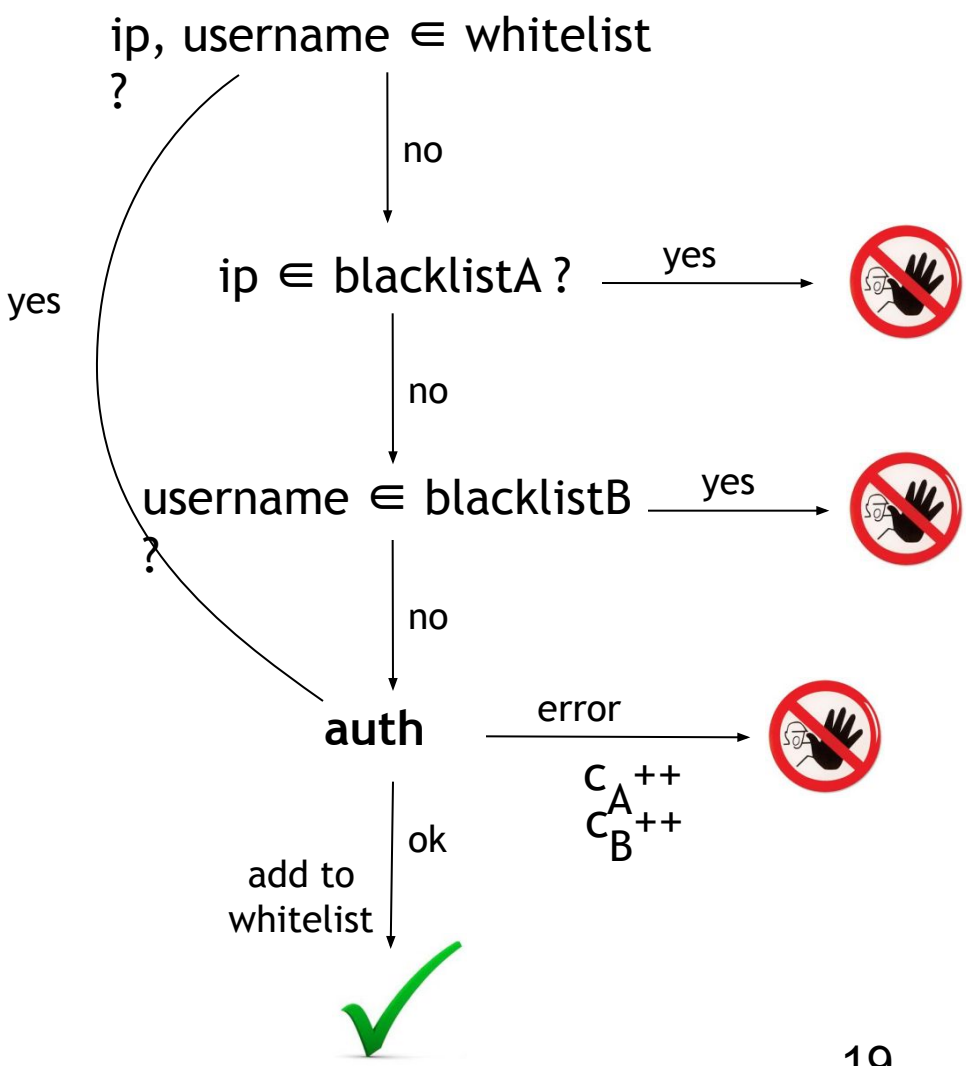
Блок-схема



Блок-схема


ip ∈ blacklistA ?
yes
no
username ∈ blacklistB ?
yes
no
auth
error
c_A++
c_B++
ok

Блок-схема


ip, username ∈ whitelist ?
no
ip ∈ blacklistA ?
yes
yes
no
username ∈ blacklistB ?
yes
no
auth
error
c_A++
c_B++
ok
add to whitelist

Блок-схема

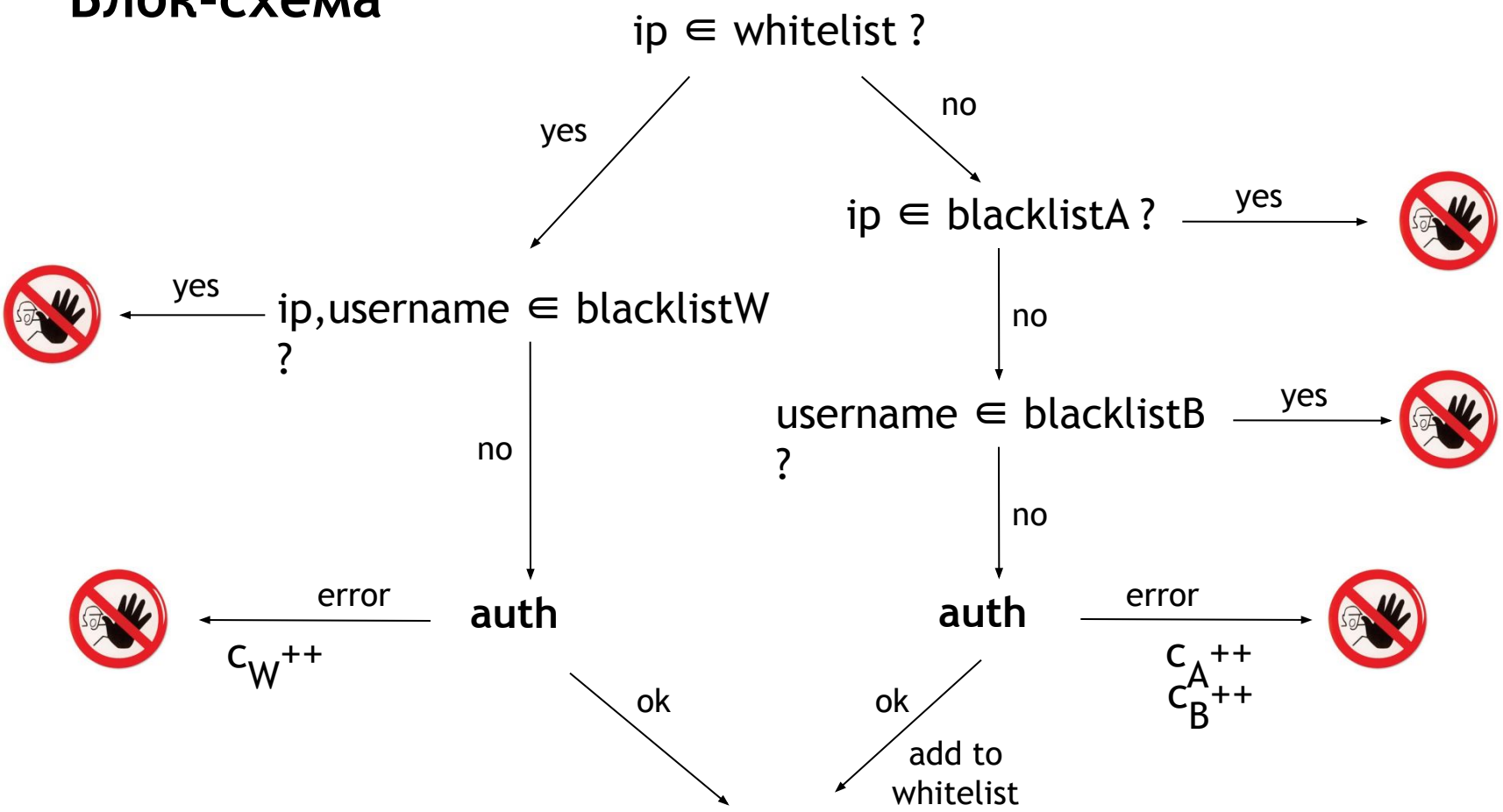

ip ∈ whitelist ?
yes
no
ip ∈ blacklistA ?
yes
no
ip,username ∈ blacklistW ?
yes
no
username ∈ blacklistB ?
yes
no
error
c_W++
auth
auth
error
c_A++
c_B++
ok
ok
add to whitelist

Диаграмма состояний - 1

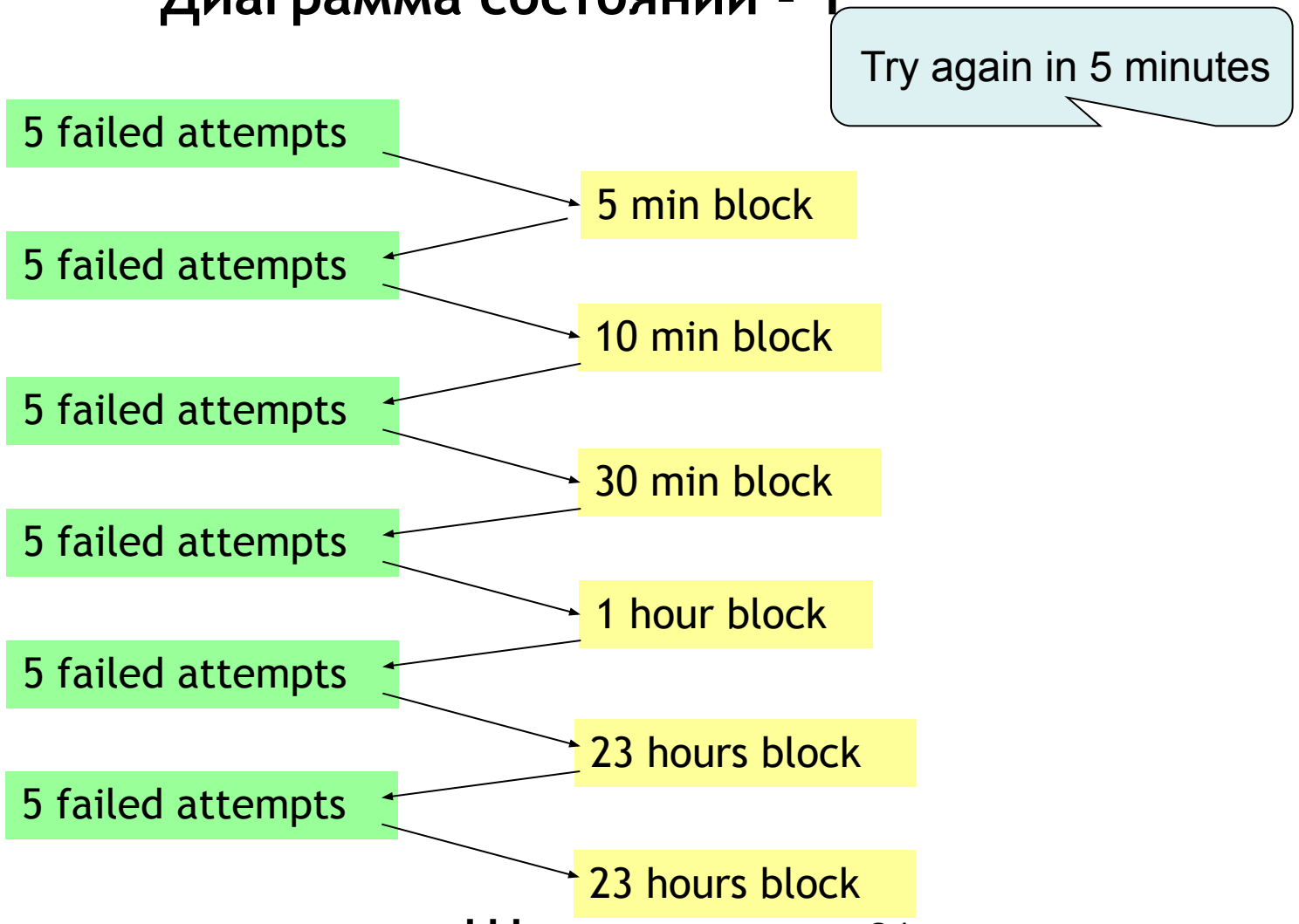


Диаграмма состояний - 2

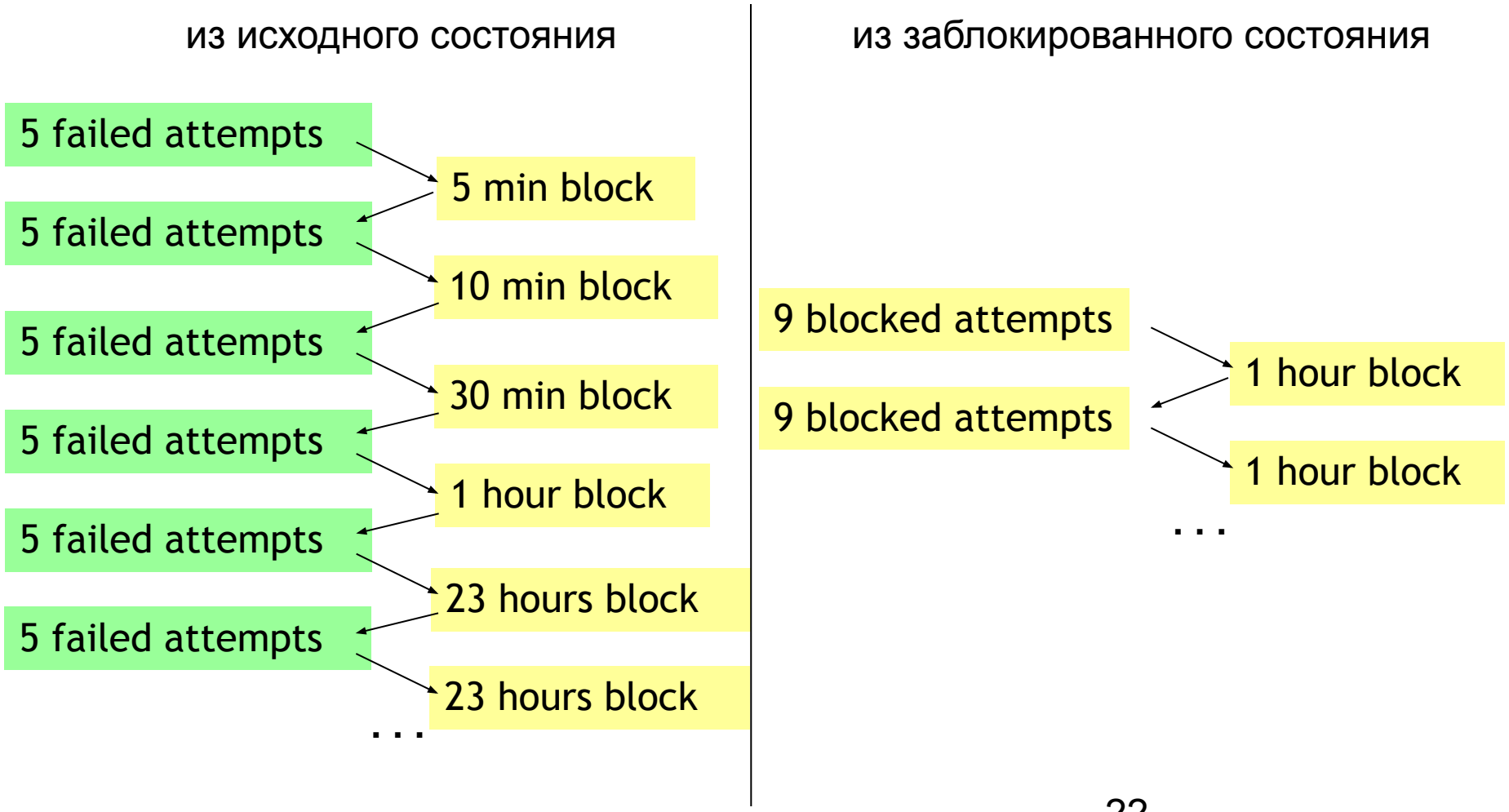
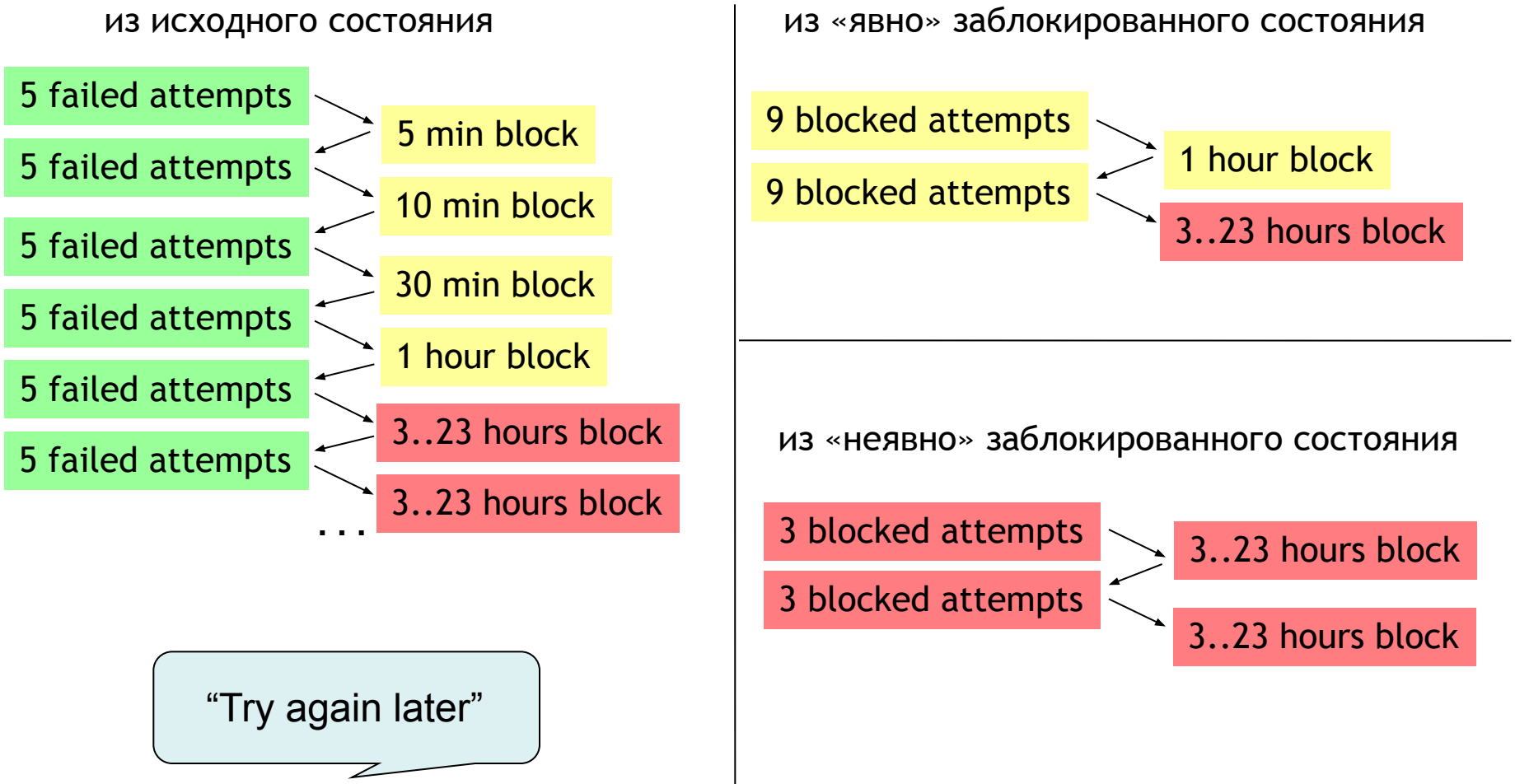


Диаграмма состояний - 3 (RedisSentryPlus)



Счетчик whitelist'а

- IP: 193.124.10.53

15 bytes

4*uchar 4 bytes

/24 3 bytes

Username: somebody@somewhere.com

~50 bytes

user_id ~8 bytes

hash(username):

md5(username) 16 bytes

? 4 bytes

$2^{32} = 4*10^9$

$7*10^9$ (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

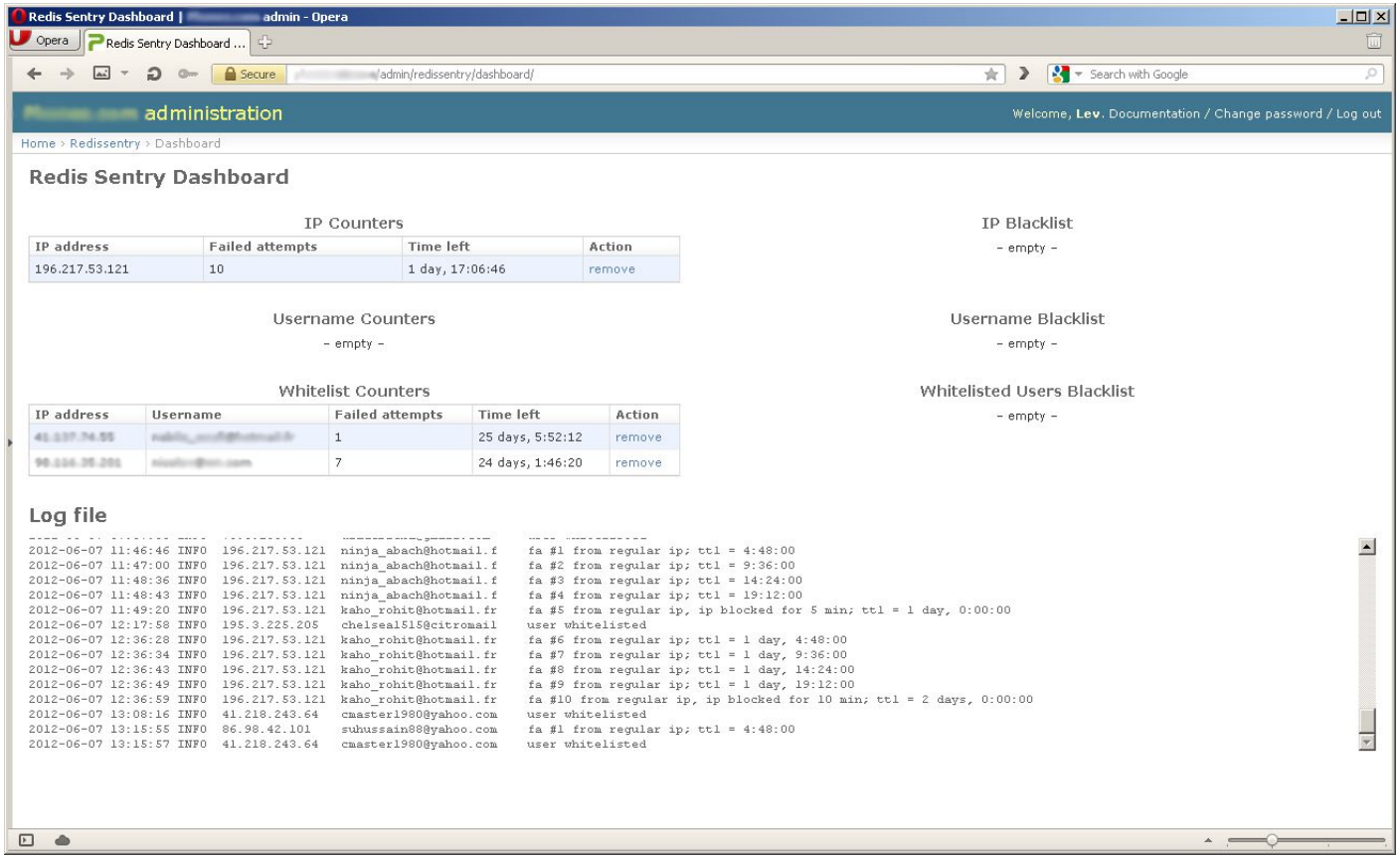
pip install pyhash

	1m	2m	14m
• 5 bytes hash-function: md5[:5]	0	4	
• 6 bytes hash-function: md5[:6]	0	0	0

Счетчики

bytes	development	production	bytes
18	Ac:*ip*	a....	5
18	Ab:*ip*	A..ip..	5
53	Bc:*username*	b....	5
53	Bb:*username*	B..hash(username)	5
68	Wc:*ip*:*username*	c.....	8
68	Wb:*ip*:*username*	C..ip.hash(username)	8

Админка


Redis Sentry Dashboard | admin - Opera
Opera
Redis Sentry Dashboard ...
Secure
admin/redissentry/dashboard/
Search with Google
administration
Welcome, Lev. Documentation / Change password / Log out
Home › Redissentry › Dashboard
Redis Sentry 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
1 | 25 days, 5:52:12 | remove
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.f fa #1 from regular ip; ttl = 4:48:00
2012-06-07 11:47:00 INFO 196.217.53.121 ninja_abach@hotmail.f fa #2 from regular ip; ttl = 9:36:00
2012-06-07 11:48:36 INFO 196.217.53.121 ninja_abach@hotmail.f fa #3 from regular ip; ttl = 14:24:00
2012-06-07 11:48:43 INFO 196.217.53.121 ninja_abach@hotmail.f 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’а
- 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	✓	✓	⛔	proxy, NAT =>блок. *целая* подсеть
~~IP-адрес + USER_AGENT~~	~~✓~~	~~✓~~	~~⛔~~	~~блок. *только* 1 браузер~~
~~IP-адрес + X_FORWARDED_FOR~~	~~✓~~	~~✓~~	~~⛔~~	~~*ровно* 1 сет.интерф.~~
username	✓	⛔	✓	заодно блок. владелец эккаунта
(IP-address, username)	✓	⛔	⛔	$1M^{x}1M = 1T$
IP-address; username	✓	✓	✓	1M + 1M = 2M; подсеть, владелец
IP-address; username; whitelist	✓	✓	✓	много счетчиков

Уязвимости реализации (устранимые)

	storage	счетчики
django snippet #1083	**main db** DDOS	global *либо* IP
ratelimitcache	memcached	IP *либо* IP+**username** overflow
django-brutebuster	**main db** DDOS	IP & **username** overflow
django-axes	**main db** DDOS	IP
django-failedloginblocker	**main db** DDOS	**username** overflow
django-lockout	memcached	IP
django-redissentry	redis	...

сброс счетчика

подмена заголовков