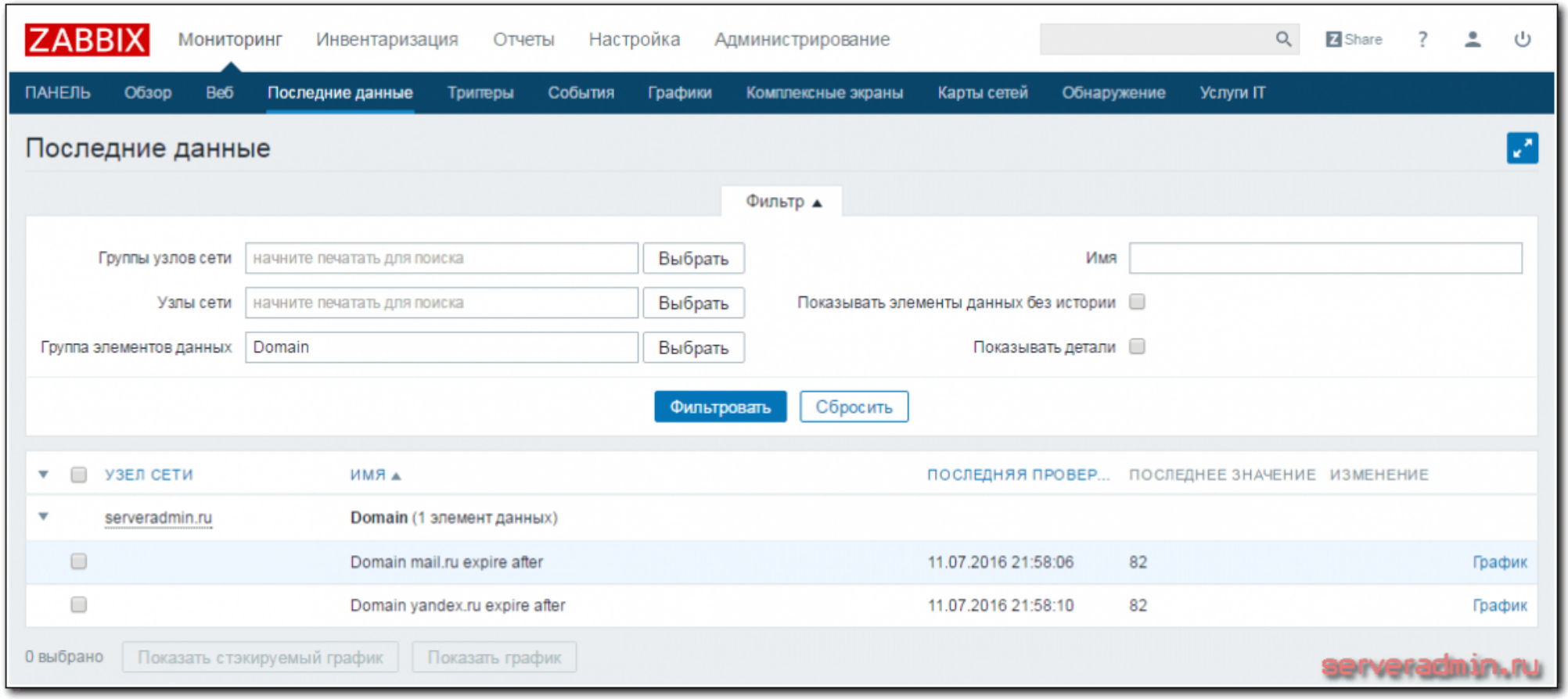


Статья исследование на тему проверки и мониторинга данных из whois сервисов. Задача мониторинга времени делегирования домена на деле оказалась не такой простой, как может показаться. Ниже я подробно расскажу обо всех нюансах мониторинга времени оплаты домена с помощью zabbix.

Содержание:

- 1 Введение
- 2 Подготовка к мониторингу за доменами
- 3 Парсинг времени делегирования домена через ruby
- 4 Linux утилиты whois и jwhois
- 5 Модуль python-whois для мониторинга оплаты доменов
- 6 Whois клиент для Node.js
- 7 Скрипт auto discovery доменов для zabbix
- 8 Настройка zabbix агента
- 9 Создание шаблона для наблюдения за доменами
- 10 Заключение

Введение

Данная статья была изначально написана пару лет назад. В целом, она отвечала на вопрос по мониторингу за делегированием доменов, но со временем к ней стало появляться все больше вопросов. В частности, появлялись новые доменные зоны, информацию по которым не получалось достать выбранным методом. К слову, в первоначальной версии статьи использовалась только одна проверка с помощью **ruby**.

Чем больше доменов я ставил на мониторинг, тем чаще сталкивался с тем, что по каким-то зонам не мог получить инфу. Из свежих примеров — .pro, .fm, .io. Наверняка были и другие, но я сталкивался с проблемами именно в этих зонах. Решил разобраться с этим вопросом поподробнее.

К моему удивлению, оказалось, что простого, понятного и надежного решения по получению информации о времени делегирования домена просто не существует. Очень подробно разобрана эта тема в статье на хабре. Прочитав и осознав все, что в ней написано, я понял, что реализовать простую потребность в мониторинге доменов малой кровью не получится.

Я стал разбираться, копать тему. Погрузился в нее плотно и потратил кучу времени на казалось бы простецкую задачу. Но увы, универсального и надежного решения так и не получил. Полученный опыт хочу зафиксировать и поделиться им с вами. Может кто-то пойдет дальше и придумает что-то более надежное.

Ниже я опишу несколько консольных способов получения информации о времени оплаты домена для передачи этой информации в zabbix:

- Ruby gem Whois.
- Консольная утилита linux — whois, jwhois.
- Питоновский модуль — python-whois.
- Whois client для node.js.

Первый способ был описан давно, поэтому он по содержанию немного выбивается из общего повествования. Мне не захотелось редактировать всю статью. Я только добавил в начало описание проблемы и другие способы получения информации в конце. Сама реализация со стороны zabbix осталась той же самой. Меняются только скрипты, которые передают в zabbix данные о сроках делегирования домена.

Для удобного выбора нужного способа, составил табличку, где указал, какой из способов какие домены поддерживает. Использовал только те домены, которые нужны мне. Стандартные домены .ru и .com поддерживают все описанные способы.

способ / домен	pro	io	fm
ruby	+	+	+
whois	—	+	—
python	+	—	+
nodejs	+	+	—

Если у вас еще нет своего сервера для мониторинга, то рекомендую материалы на эту тему. Для тех, кто предпочитает систему CentOS:

1. Установка CentOS 8.
2. Настройка CentOS 8.
3. Установка и настройка zabbix сервера.

То же самое на Debian 10, если предпочитаете его:

1. Установка Debian 10.
2. Базовая настройка Debian.
3. Установка и настройка zabbix на debian.

Подготовка к мониторингу за доменами

Для мониторинга за сроком делегирования доменов мы будем использовать скрипт на языке программирования **Ruby**. Чтобы скрипт работал, необходимо установить Ruby на сервер. Если вы используете, как я, сервер на базе CentOS 7, то можете воспользоваться моей инструкцией по установке последней версии Ruby на CentOS 7. Там как раз речь идет об установке необходимого гема **whois-parser** для мониторинга срока делегирования домена.

Парсинг времени делегирования домена через ruby

Если у вас настроен и готов к работе zabbix и установлен ruby, приступаем к настройке скриптов. Идем в папку */etc/zabbix/scripts* и создаем там скрипт:

```
# mcedit /etc/zabbix/scripts/domain_left.rb
```

```
#!/usr/bin/env ruby

require 'whois-parser'

record = Whois.whois(ARGV[0])
parser = record.parser
expire = parser.expires_on.strftime("%Y-%m-%d")
today = Time.now.strftime("%Y-%m-%d")
expire_date = DateTime.parse(expire)
today_date = DateTime.parse(today)
left = (expire_date - today_date).to_i
```

```
puts left
```

Его тут же можно проверить. Сохраняете скрипт и запускаете в командной строке, указывая через пробел имя домена для проверки:

```
# /etc/zabbix/scripts/domain_left.rb serveradmin.ru
394
```

Отлично, на выходе просто число, которое очень удобно передать в zabbix. Данный скрипт я написал не сам, а подсмотрел у автора этой статьи. У него есть большой минус. В скрипте используется whois-parser и его функция expires_on. Сделано это для упрощения создания скрипта. Не пришлось вручную парсить вывод, за нас это сделали авторы гема whois-parser. Но эти функции работают не со всеми доменами. В разных зонах вывод может быть разным. Покажу на примере.

Для этого создадим простой скрипт на ruby, который будет просто выводить информацию whois без парсинга, в сыром виде.

```
#!/usr/bin/env ruby

require 'whois-parser'
domain = ARGV[0]
whois = Whois::Client.new
r = whois.lookup(domain)
puts r
```

Сохраняем скрипт и проверим с его помощью 2 домена: из зоны .pro и зоны .ru.

```
# ./domain.rb server.pro
```

```
# ./domain.rb serveradmin.ru
```

Если вы внимательно посмотрите на вывод, то обнаружите, что в первом случае строка с информацией о сроке оплаты домена выглядит так:

```
Registry Expiry Date: 2019-08-18T00:00:00Z
```

а в другом вот так:

```
paid-till:     2019-08-09T11:01:06Z
```

Whois-parser умеет обрабатывать только первую строчку, вторую он не понимает, поэтому приведенный выше скрипт выдаст ошибку. Но в целом, библиотека ruby whois по моим опытам корректно обрабатывает больше всего доменов. Нужно только правильно распарсить вывод. Я программировать на ruby не умею, разбираться у меня не было времени, поэтому я оставил это решение как есть. Можно, конечно, выводить в текстовый файл, а потом парсить вывод через bash. Это я умею, но не захотелось костылить таким образом. Да и в целом, мне не нравится этот способ тем, что надо ставить ruby и регулярно обновлять его библиотеку. Только это гарантирует поддержку новых зон и всех изменений по старым.

В общем и целом способ с приведенным скриптом на ruby работает не хуже всех остальных способов, а если нормально распарсить выводы по всем нужным зонам, то это будет самый информативный и надежный вариант, при условии, что вы будете везде ставить ruby и обновлять библиотеку whois. Только этот способ показывал информацию по всем доменам, что мне попадались. Если доделаю скрипт, чтобы он работал со всеми доменами — обновлю.

Пока же я просто сделал костыль и распарсил вывод с помощью bash следующим образом:

```
#!/bin/bash

DOMAIN="$1"

data=$(/etc/zabbix/scripts/domain-simple.rb $1 | grep -E 'paid|Expir' | grep -o -E
```

```
'[0-9]{4}.[0-9]{2}.[0-9]{2}|[0-9]{2}/[0-9]{2}/[0-9]{4}' | tr . / | awk 'NR == 1')
expire=$((`date -d "$data" '+%s'`))
today=$((`date '+%s'`))
lefts=$(($expire - $today))
leftd=$(($lefts/86400))
echo $leftd
```

Сам скрипт domain-simple.rb:

```
#!/usr/bin/env ruby

require 'whois-parser'
record = Whois.whois(ARGV[0])
puts record
```

Я просто делаю полный вывод whois через ruby, а распарсиваю его башем, так как умею в нем работать. Получился вариант, который работает со всеми доменами. Можно этим скриптом передавать данные в zabbix.

Linux утилиты whois и jwhois

В комментариях к первоначальной статье мне подсказали про консольную утилиту whois, а позже я нагуглил и еще одну — jwhois. Я посмотрел на них внимательно и написал простенький скрипт для парсинга информации о делегировании. Данный способ самый простой и удобный. Обе утилиты ставятся через yum, первая из базового репозитория, вторая из репозитория epel. Не нужно дополнительных инструментов. Информативность средняя — все популярные домены есть, в том числе .io, но .pro и .fm не поддерживает. Увы и ах, мне они нужны.

Установим утилиту whois:

```
# yum install whois
```

Простой скрипт, который парсит вывод утилиты и показывает, сколько осталось дней до завершения проплаченного периода домена.

```
#!/bin/bash

DOMAIN="$1"

data=$(whois $1 | grep -E 'paid|Expir|expir' | grep -o -E '[0-9]{4}.[0-9]{2}.[0-9]{2}|[0-9]{2}/[0-9]{2}/[0-9]{4}' | tr .
/ | awk 'NR == 1')
expire=$((`date -d "$data" '+%s'`))
today=$((`date '+%s'`))
lefts=$(($expire - $today))
leftd=$(($lefts/86400))
echo $leftd
```

Сохраняйте скрипт и проверяйте работу. Работает так же, как и скрипт для ruby.

```
# ./domain_left.sh serveradmin.ru
358
```

Я для себя решил остановиться на этом способе, как самом простом и универсальном.

Модуль python-whois для мониторинга оплаты доменов

В рамках своего исследования я решил проверить как с помощью python можно получать информацию whois. Я подозревал, что должен быть готовый модуль для этого и не ошибся. Такой модуль есть — python-whois. Для его работы нужен python версии 2, который по-умолчанию стоит в системе CentOS 7, что очень удобно.

Нам нужно только установить этот модуль через pip. Pip для 2-й версии ставится из репозитория epel.

```
# yum install python2-pip
```

```
# pip install python-whois
```

Дальше набросал небольшой скрипт под заданную задачу, аналог предыдущих скриптов.

```
#!/usr/bin/env python

import whois
from datetime import datetime
from sys import argv,exit

now = datetime.now()
domain = argv[1]

w = whois.whois(domain)

if type(w.expiration_date) == list:
    w.expiration_date = w.expiration_date[0]
else:
    w.expiration_date = w.expiration_date

domain_expiration_date = str(w.expiration_date.day) + '/' + str(w.expiration_date.month) + '/' +
str(w.expiration_date.year)
timedelta = w.expiration_date - now
days_to_expire = timedelta.days
```

Проверка:

```
# ./domain-left.py serveradmin.ru
```

```
358
```

Работает так же, как и предыдущие скрипты. Решение на python поддерживает зоны .pro, .fm, не поддерживает .io.

Whois клиент для Node.js

Последнее рассмотренное мной консольное решение по получению данных whois основывается на клиенте для Node.js. Этот способ неудобен, как и с ruby, тем, что надо отдельно ставить node на сервер. Если python на centos сервере точно будет, то node придется ставить отдельно. Сделаем это.

```
# curl – silent – location https://rpm.nodesource.com/setup_8.x | sudo bash -
# yum install nodejs
```

Устанавливаем whois client:

```
# npm install whois
```

Под node.js я программировать вообще не умею и со скриптами не работал никогда. Так что готового скрипта для zabbix не сделал. Покажу на простом примере, как выполнять проверку. Делаем скрипт и в нем сразу указываем домен для проверки.

```
var whois = require('whois')
whois.lookup('serveradmin.ru', function(err, data) {
    console.log(data)
    })
```

Запускаем его:

```
# node domain-left.js
```

В выводе увидите привычную информацию whois в таком же виде, как и для остальных проверок. Если бы этот способ проверял все необходимые мне домены, я бы его доделал. Он понимает домены .pro и .io, но не понимает .fm. Так что я не стал на нем подробно останавливаться.

Скрипт auto discovery доменов для zabbix

Для того, чтобы автоматом передавать в заббикс сразу все наши домены, создадим еще один скрипт, который будет парсить текстовый файл со списком доменов и передавать его в zabbix.

```
# mcedit /etc/zabbix/scripts/domain_discovery.sh
```

```
#!/bin/bash
JSON=$(for i in `cat /etc/zabbix/scripts/domain_list.txt`; do printf "{\"{#DOMAIN}\":\"$i\"},"; done | sed
's/^\(.*\),$/\1/')
printf "{\"data\":["
printf "$JSON"
printf "]}"
```

И последнее — создадим файл со списком доменов. Каждый домен с новой строки.

```
# mcedit /etc/zabbix/scripts/domain_list.txt
```

```
yandex.ru
```

```
mail.ru
```

Проверим работу скрипта для парсинга:

```
# /etc/zabbix/scripts/domain_discovery.sh
{"data":[{"{#DOMAIN}":"mail.ru"},{"{#DOMAIN}":"yandex.ru"}]}
```

Вывод подходящий для передачи в заббикс.

Настройка zabbix агента

Я настраиваю zabbix агент на работу со скриптом на ruby. Вы можете выбрать скрипт по своему усмотрению. Изменить нужно будет только путь к скрипту в описании итема domain.expire.

Добавим скрипты через пользовательские параметры (UserParameter) агента. Для этого идем в папку с пользовательскими скриптами */etc/zabbix/zabbix_agentd.d* и создаем там конфиг для наших параметров:

```
# mcedit /etc/zabbix/zabbix_agentd.d/domain.conf
```

```
UserParameter=domain.discovery[*], /bin/bash /etc/zabbix/scripts/domain_discovery.sh
UserParameter=domain.expire[*], /usr/local/bin/ruby /etc/zabbix/scripts/domain_left.rb $1
```

Обращаю внимание на путь к ruby. В разных системах он может быть разный. Проверьте куда он у вас установлен и отредактируйте путь. Проверить можно командой:

```
# whereis ruby
```

```
ruby: /usr/lib64/ruby /usr/local/bin/ruby /usr/local/lib/ruby /usr/share/ruby /usr/src/ruby-2.3.1/ruby.o
/usr/src/ruby-2.3.1/ruby /usr/src/ruby-2.3.1/ruby.c
```

Сохраняйте конфиг и перезапускайте zabbix agent:

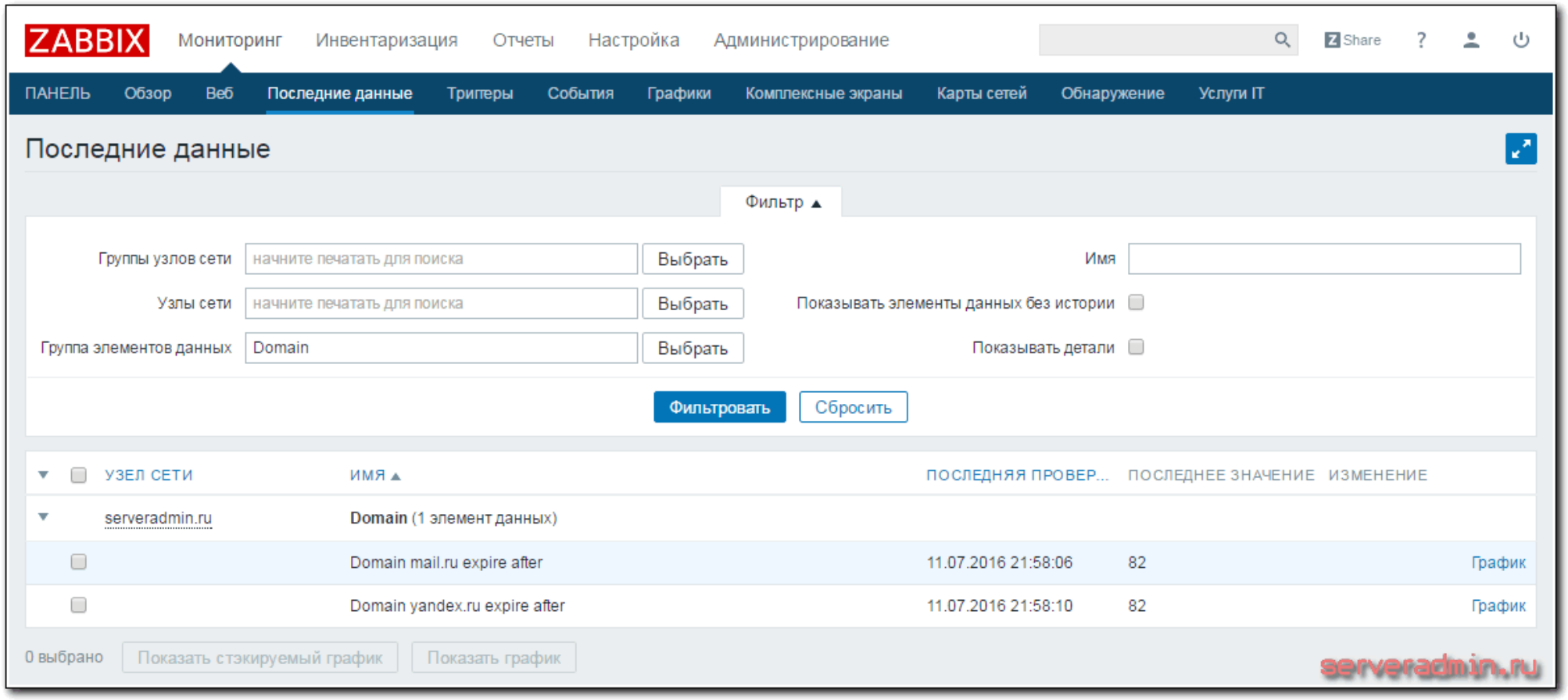
```
# systemctl restart zabbix-agent
```

В консоли закончили настройки, теперь идем в панель администрирования заббикс.

Создание шаблона для наблюдения за доменами

Дальше все стандартно и просто. Я выгрузил готовый шаблон у себя и предлагаю вам его скачать и импортировать, чтобы вручную не создавать все необходимое. Забираем файл — шаблон.

Импортируете шаблон себе в систему. Назначаете его хосту, в котором настроили скрипты и ждете появления данных. Минут через 5 проверяете в Latest Data:



Забавно, время делегирования обоих доменов одинаковое. Сначала подумал, что глюк, решил проверить вручную. Оказалось, что все верно. У них одинаковый срок продления. Без проблем работает наблюдение за доменами .рф. Их нужно перевести в Punycode и добавить так же, как и обычные домены.

Если необходимо, можете настроить повторяющиеся оповещения о времени делегирования домена. В данном случае это может быть актуально, так как не продлив сразу домен, можно потом позабыть.

Заключение

> Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоем распоряжении. Расскажи, как сделать правильно!

Очередной пример простоты и удобства системы мониторинга Zabbix. С его помощью можно мониторить все, что угодно. Недавно я настраивал мониторинг системы управления насосами. Система передавала параметры в текстовый файл, а я его парсил и отправлял интересующие параметры заббиксу. Подробнее об этом написал в отдельной статье.

Помогла статья? Подписывайся на telegram канал автора

Анонсы всех статей, плюс много другой полезной и интересной информации, которая не попадает на сайт.