СВД
Встраиваемые
Системы

Применение интеллектуальной платформы «Синаптика» для мониторинга аномальной активности во встраиваемых системах

Константин Жвакин

Инженер-программист
Отдел операционных систем

Мониторинг активности

- **Мониторинг активности** — совокупность методов сбора, преобразования и анализа информации о процессах, происходящих в информационной системе.

- Задачей системы мониторинга является определение, какая активность является доверенной, а какая — аномальной.

Сбор информации об активности в Нейтрино

В Нейтрино доступны следующие способы получения информации об активности системы:

- данные ядра (виртуальная файловая система proc);
- сбор трассировочной информации;
- данные из сетевого стека;
- дисковая подсистема;
- другое.

Монитор аномальной активности

Для мониторинга активности в системе, был разработан программный комплекс мониторинга аномальной активности — **amon** (**a**nomaly **mon**itor).

Монитор аномальной активности - это обучаемая система, которая предназначена для анализа данных и выявления аномалий из различных источников. Взаимодействие с источниками данных и ядром приложения осуществляется через API платформы Синаптика.

Особенности программного комплекса

Программный комплекс имеет следующие особенности:

- модульность;
- дообучение в процессе эксплуатации;
- выполнение на конечном устройстве;
- аппаратное ускорение анализа;
- гибкое конфигурирование комплекса;
- API для реализации клиентских приложений.

Состав программного комплекса

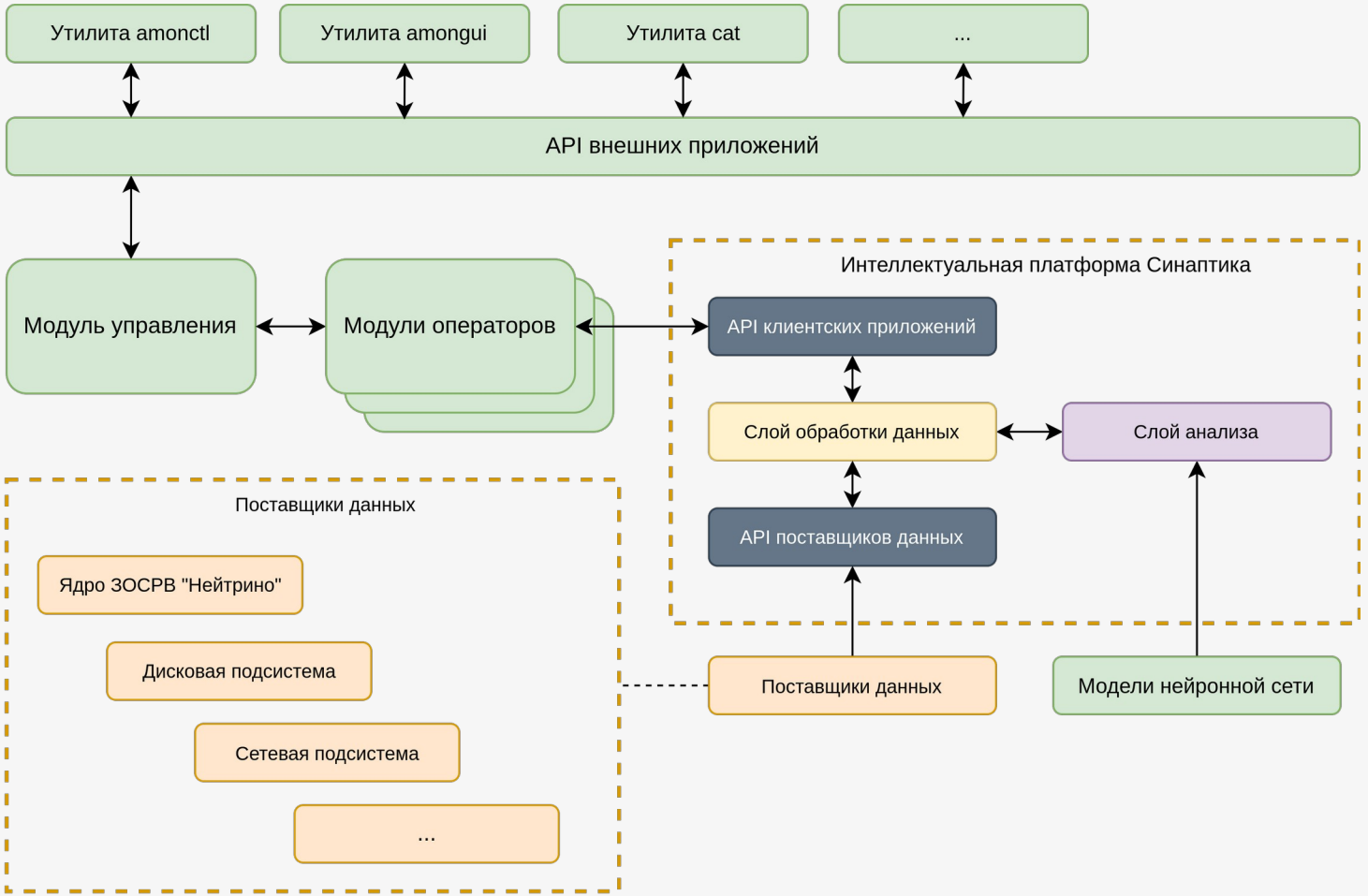
Программный комплекс состоит из следующих компонентов:

- **amon** — основной сервис мониторинга активности процессов в Нейтрино, осуществляющий сбор и анализ данных, выявление аномалий.
- **amonctl** — утилита управления сервисом **amon**.
- **amongui** — графический интерфейс **amon**.
- **amon-operator-kernel.so** — модуль оператора данных, собирающий и анализирующий информацию об активности процессов из ядра Нейтрино.

Архитектура программного комплекса

На схеме представлены следующие компоненты:

- **модуль управления** - осуществляет управление другими модулями, хранит информацию об объектах и оповещениях и предоставляет интерфейс взаимодействия с сервисом;
- **модули операторов** - каждый из них осуществляют управление пайплайном обработки данных для своего поставщика данных и генерирует оповещения при появлении аномалий;
- **слой обработки данных** - организовывает процесс обработки данных (пайплайн);
- **слой анализа данных** - выполняет анализ посредством нейронной сети;
- **поставщики данных** - предоставляет информацию об активности.



Взаимодействие с сервисом amon

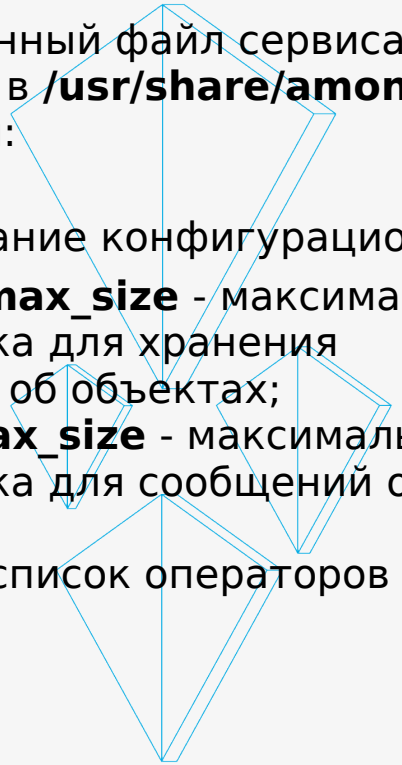
Способы взаимодействия с сервисом **amon** (через менеджер ресурсов **/dev/amon**):

- стандартные утилиты (cat, …);
- утилита управления сервисом **amonctl**;
- графический интерфейс **amongui**;
- API.

Пример конфигурационного файла

Конфигурационный файл сервиса **amon** располагается в **/usr/share/amon/conf** и содержит поля:

- **name** - название конфигурационного файла;
- **object_list_max_size** - максимальный размер списка для хранения информации об объектах;
- **alert_list_max_size** - максимальный размер списка для сообщений об аномалиях;
- **operators** - список операторов данных.



```
{
   "name": "default",
   "object_list_max_size": 150,
   "alert_list_max_size": 500,
   "operators": [
      {
         ...
      }
   ]
}
```

Сбор информации о процессах (оператор данных kernel)

Оператор данных **kernel** позволяет собирать и анализировать следующую информацию об активности процессов:

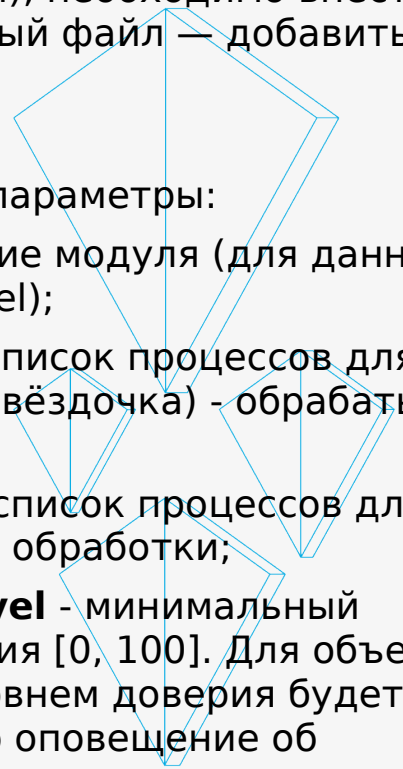
- запущенные процессы и потоки;
- потребление памяти;
- состояние стека;
- открытые файловые дескрипторы;
- открытые соединения;
- состояние каналов;
- запущенные таймеры;
- обработчики прерываний;
- разделяемые библиотеки в памяти.

Параметры оператора данных kernel

Для того, чтобы добавить новый оператор данных (например, kernel), необходимо внести изменения в конфигурационный файл — добавить его в список операторов.

Рассмотрим его параметры:

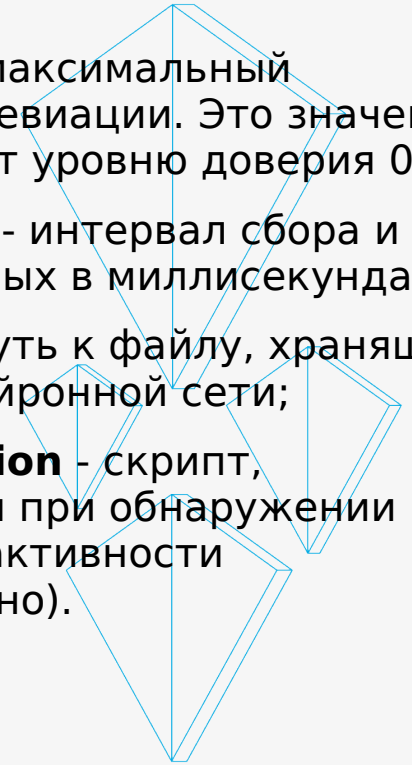
- **name** - название модуля (для данного модуля — kernel);
- **include_list** - список процессов для обработки, * (звёздочка) - обрабатывать все процессы;
- **exclude_list** - список процессов для исключения из обработки;
- **alert_trust_level** - минимальный уровень доверия [0, 100]. Для объектов с меньшим уровнем доверия будет сгенерировано оповещение об аномалии.



```
{
   "name": "default",
   "object_list_max_size": 150,
   "alert_list_max_size": 500,
   "operators": [
      {
         "name": "kernel",
         "include_list": [
            "*"
         ],
         "exclude_list": [
            "usr/bin/amon",
         ],
         "alert_trust_level": 85,
         "trust_max": 0.7,
         "polling_time": 1000,
         "structure": "/usr/share/amon/amon-operator-kernel-
structure.json",
         "anomaly_action": "/usr/share/amon/scripts/action.sh"
      }
   ]
}
```

Параметры оператора данных kernel

- **trust_max** - максимальный показатель девиации. Это значение соответствует уровню доверия 0%;
- **polling_time** - интервал сбора и анализа данных в миллисекундах;
- **structure** - путь к файлу, хранящему структуру нейронной сети;
- **anomaly_action** - скрипт, выполняемый при обнаружении аномальной активности (необязательно).



```
{
   "name": "default",
   "object_list_max_size": 150,
   "alert_list_max_size": 500,
   "operators": [
      {
         "name": "kernel",
         "include_list": [
            "*"
         ],
         "exclude_list": [
            "usr/bin/amon",
         ],
         "alert_trust_level": 85,
         "trust_max": 0.7,
         "polling_time": 1000,
         "structure": "/usr/share/amon/amon-operator-kernel-
structure.json",
         "anomaly_action": "/usr/share/amon/scripts/action.sh"
      }
   ]
}
```

Пример использования amon

Шаг 0. Для начала нам необходимо запустить сервис.

```
# amon -v &
[1] 2117671
# Loading module amon-operator-kernel.so
Configuration file loaded
Running kernel operator...
Loading analyzer...
Using learned structure...
Model name  : amon neural network
Model desc  : neural net structure for amon
Model ver   : 1.1
Neuron count: 224

Kernel operator init done
```

```
# amon -h
amon - Intellectual activity monitor
Usage: amon [params]
-h                      -  show help
-v                      -  verbose level (default - 0)
-c <file>               -  path to amon configuration file
```

Пример использования amon

Шаг 1. Рассмотрим утилиту управления **amon** - **amonctl:**

```
# amonctl -h

amonctl - intellectual anomaly monitor control utility
Usage: amonctl [params]
-h           -  show help
-S           -  show summary
-P           -  show captured objects list
-A           -  show anomaly list
-p name      -  show information about object
-a aid       -  show information about anomaly
-L           -  switch to learning mode
-R           -  switch to recognition mode
-c           -  if using -L or -R option, choose object class for mode switch
-o           -  if using -L or -R option, choose object id for mode switch
-l aid       -  learn anomaly with specified id
```

Пример использования amon

Шаг 2. Просмотрим общую информацию о состоянии системы.

Тут мы видим, что:

- Уровень доверия в системе составляет 0% (что говорит о необученности сети);
- Проанализировано 37 объектов;
- 0 предупреждений;
- Время обработки — 137 мс.

```
# amonctl -S

amon summary

-------------------------------------------------------------

-------------------------------------------------------------

Current system trust value: 0%
Objects captured: 37
Alerts captured: 0
Processing time: 137 ms
```

Пример использования amon

Шаг 3. Для обучения необходимо запустить режим обучения на некоторое время.

```
# amonctl -L
All objects switched to learning mode
```

Обучение накладывает определённые требования на воспроизводимость доверенной активности.

Шаг 4. Через некоторое время после обучения мы можем перевести систему в режим анализа активности.

```
# amonctl -R
All objects switched to recognition mode
```

Пример использования amon

Шаг 5. Снова просмотрим состояние системы.

Можем увидеть, что:

- текущий уровень доверия составляет 99.9997%,
- было просканировано 37 объектов,
- сгенерировано 0 оповещений (alerts) о достижении минимального порога,
- а также среднее время обработки составляет порядка 176 мс.

```
# amonctl -S
amon summary

------------------------------------------------------------

              ************************************************
              ************************************************
              ************************************************
              ************************************************
              ************************************************
              ************************************************
              ************************************************
              ************************************************
              ************************************************
              ************************************************
------------------------------------------------------------

Current system trust value: 99.9997%
Objects captured: 37
Alerts captured: 0
Processing time: 176 ms
```

Пример использования amon

Шаг 6. Для того чтобы изучить объекты, анализ которых выполняется в данный момент, необходимо выполнить команду:

```
# amonctl -P

================================================================================
ID              Object name                             Class name      Status          Trust value
241685          bin/login                               kernel          ANALYZING       100.00      0.00000
585764          bin/sh                                  kernel          ANALYZING       100.00      0.00000
...
258074          usr/sbin/dhcp.client                    kernel          ANALYZING       100.00      0.00000
344094          usr/sbin/inetd                          kernel          ANALYZING       100.00      0.00000
360458          usr/sbin/sshd                           kernel          ANALYZING       100.00      0.00000
577565          usr/sbin/sshd                           kernel          ANALYZING       100.00      0.00000
335894          usr/sbin/sshd                           kernel          ANALYZING       100.00      0.00000
Processing time: 150 ms
```

Пример использования amon

Шаг 7. Просмотрим детально информацию по каждому объекту.

Например, рассмотрим процесс **/usr/sbin/sshd** с **PID 360458**.

Выводится информация об отклонении параметров от состояния, на котором была обучена система, список анализируемых параметров, в данном случае это, например, информация:

- об используемой памяти,
- о совместно-используемых библиотеках,
- о созданных потоках,
- …

```
# amonctl -p usr/sbin/sshd
Object ID:                                  360458
Object name:                                usr/sbin/sshd
Object classname:                           kernel
Trust value:                                100 / 100
Deviation value:                            0
 Deviation reasons information
   Process memory alloc anomaly             0
   Opened channels count anomaly            0
   Opened file descriptors anomaly          0
   Used timers anomaly                      0
   Used CPU time anomaly                    0
   Thread stack anomaly                     0
   Used libraries anomaly                   0
Parameters info:
{
   "common": {
       "cpu_load": "0.00",
       "hash": "9999428",
       "mem_code": "4583424",
       "mem_data": "167936",
       "mem_heap": "516096",
       "mem_load": "0.99",
       "mem_other": "0",
       "mem_stack": "24576",
       "mem_total": "5292032",
       "name": "usr/sbin/sshd",
       "num_chancons": "2",
       "num_fdcons": "12",
       "num_timers": "0",
       "pid": "360458"
   },
   "libs": [
       {
           "hash": "19999616",
           "mem_data": "8192",
           "name": ""
       },
       {
           "hash": "69999530",
           "mem_data": "20480",
           "name": ""
       },
       {
           "hash": "19999521",
           "mem_data": "24576",
           "name": ""
       },
       {
           "hash": "19999426",
           "mem_data": "114688",
           "name": ""
       }
   ],
   "threads": [
       {
           "mem_stack": "24576",
           "name": "thread 1"
       }
   ]
}
```

Пример использования amon

Шаг 8. Попробуем теперь спровоцировать систему сгенерировать оповещение об аномалии. Запустим в Нейтрино калькулятор (**usr/photon/bin/phcalc**). Поскольку во время обучения он не запускался, его активность покажется системе подозрительной.

```
# amonctl -A

================================================================================
Alert ID Object name                        Timestamp              Trust value  Reason
1        [1257513] usr/photon/bin/phcalc    08.10.2024 14:15:12    0.000000     Process memory alloc anomaly, Opened channels count anomaly, Opened file descriptors anomaly, Thread stack anomaly
2        [1257513] usr/photon/bin/phcalc    08.10.2024 14:15:13    0.000000     Process memory alloc anomaly, Opened channels count anomaly, Opened file descriptors anomaly, Thread stack anomaly
3        [1257513] usr/photon/bin/phcalc    08.10.2024 14:15:14    0.000000     Process memory alloc anomaly, Opened channels count anomaly, Opened file descriptors anomaly, Thread stack anomaly
4        [1257513] usr/photon/bin/phcalc    08.10.2024 14:15:15    0.000000     Process memory alloc anomaly, Opened channels count anomaly, Opened file descriptors anomaly, Thread stack anomaly
...
```

Когда уровень доверия падает ниже указанного в конфигурационном файле (например, 85%), генерируется оповещение (alert) и оно попадает в журнал аномалий (который содержит информацию об аномальных объектах в хронологическом порядке). Каждому оповещению присвается Alert ID (aid). Интервал опроса 1000мс.

Допустим, объекту **usr/photon/bin/phcalc** присваивается **aid** равный **146**.

Пример использования amon

```
# amonctl -a 146
Alert ID:                                146
Timestamp:                               08.10.2024 14:34:50

Object info
 ID                                      1323049
 Name                                    usr/photon/bin/phcalc
 Classname                               kernel
 Trust value                             0 / 100
 Total deviation                         2.29798

 Deviation reasons info
   Process memory alloc anomaly          15.9103
   Opened channels count anomaly         0.000210943
   Opened file descriptors anomaly       0.000259945
   Used timers anomaly                   0
   Used CPU time anomaly                 0
   Thread stack anomaly                  0.175085
   Used libraries anomaly                0

 Parameters info
{
   "common": {
       "cpu_load": "0.00",
       "hash": "69999340",
       "mem_code": "2588672",
       "mem_data": "110592",
       "mem_heap": "233472",
       "mem_load": "0.55",
       "mem_other": "0",
       "mem_stack": "28672",
       "mem_total": "2961408",
       "name": "usr/photon/bin/phcalc",
       "num_chancons": "4",
       "num_fdcons": "6",
       "num_timers": "0",
       "pid": "1323049"
   },
   "libs": [
```

Шаг 9. Просмотрим подробную информацию об этой аномалии.

Шаг 10. Для того, чтобы в дальнейшем система не определяла калькулятор как аномалию, дообучим её:

```
# amonctl -l 146
Amon accepted anomaly with id 146
```

Пример использования amon

Шаг 11. Теперь если вывести общую информацию о состоянии системы и открыть калькулятор, то доверие не будет падать.

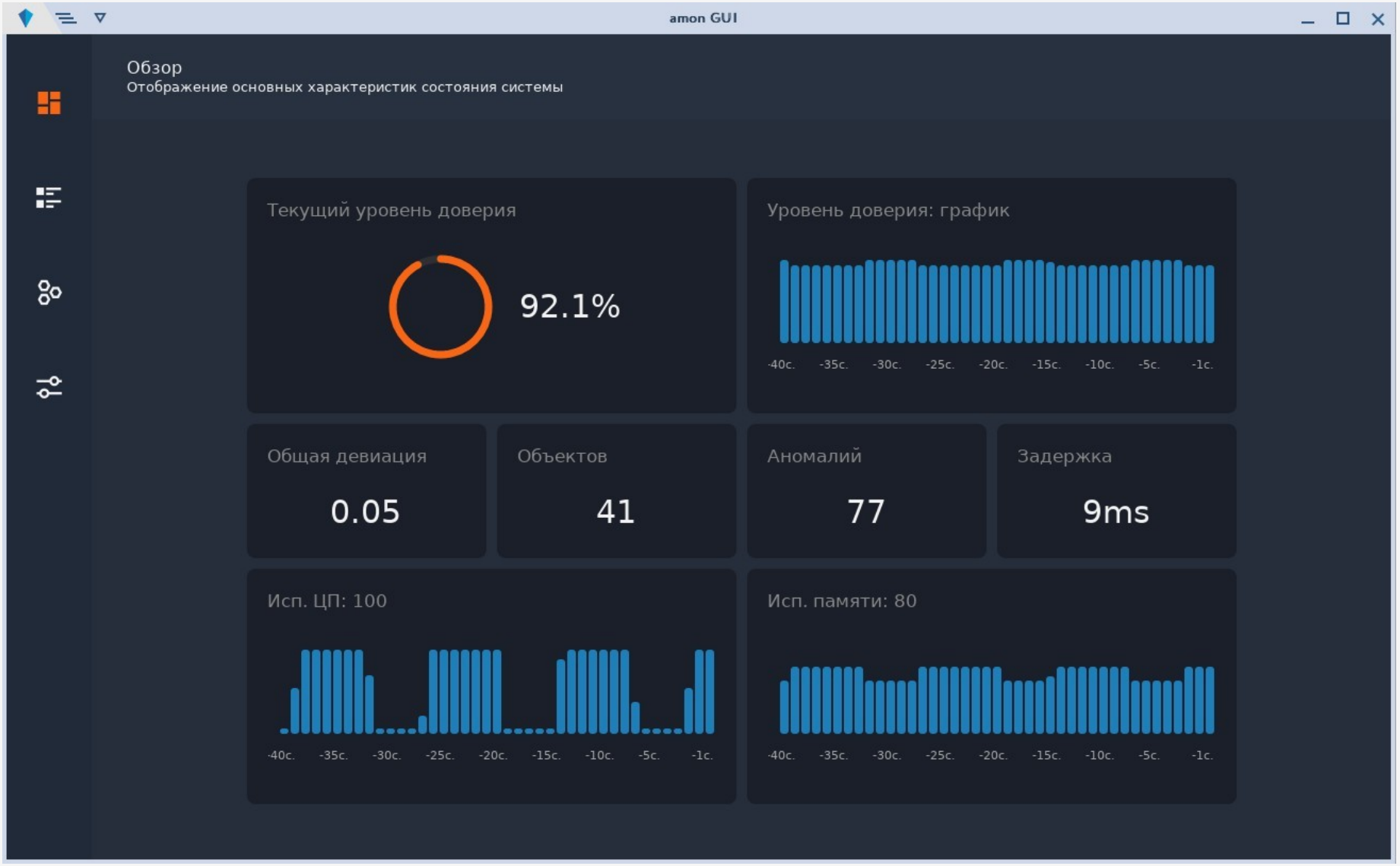
```
# amonctl -S
amon summary

-------------------------------------------------------------

                                          ********************
                                          ********************
                                          ********************
                                          ********************
                                          ********************
                                          ********************
                                          ********************
                                          ********************
                                          ********************

-------------------------------------------------------------

Current system trust value: 99.9997%
Objects captured: 38
Alerts captured: 147
Processing time: 147 ms
```

Пример использования amon

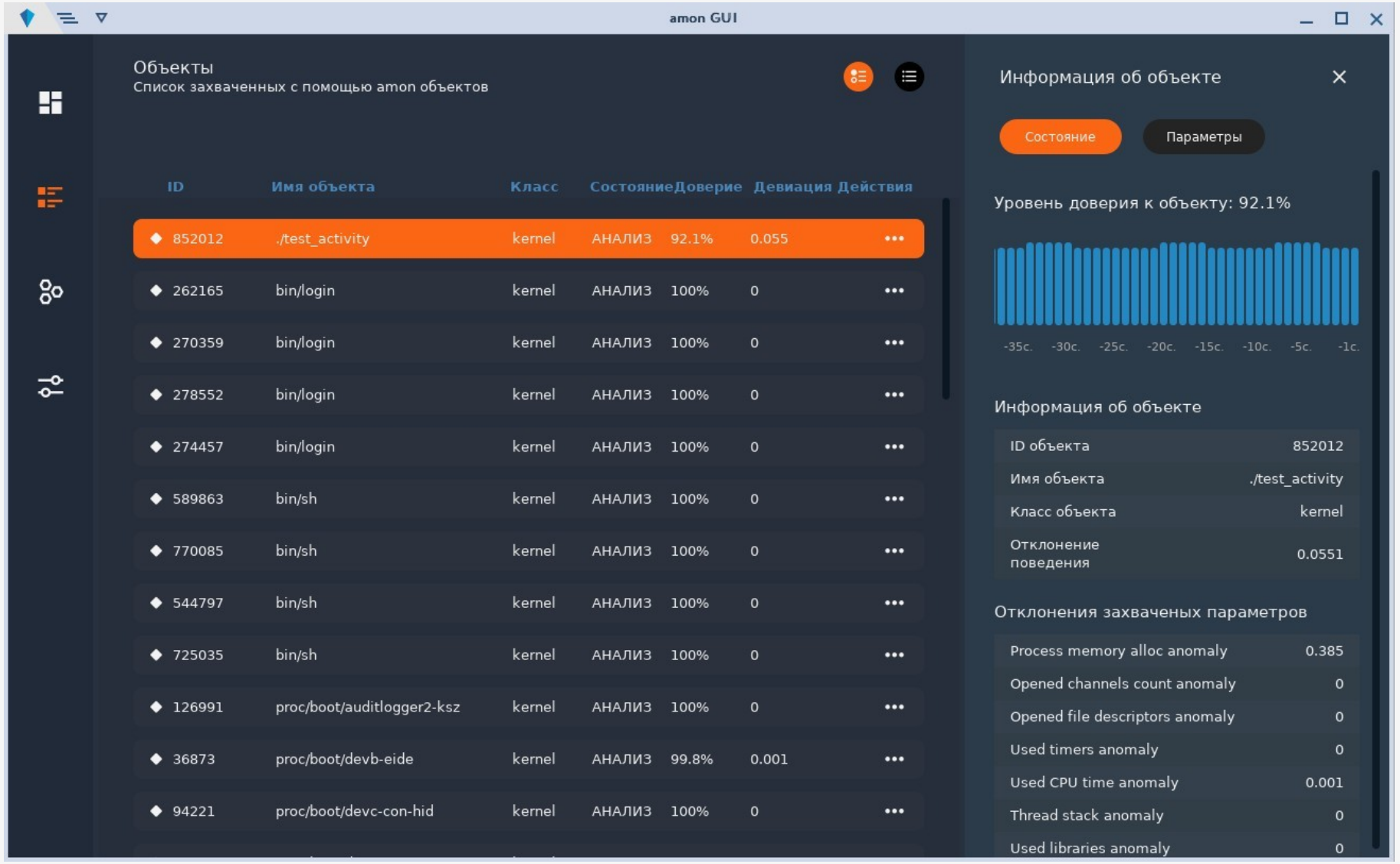
Шаг 12. Для того, чтобы откатить **amon** к необученному состоянию:

```
# slay amon
# rm -rf /usr/share/amon/amon-operator-structure-learned.json
```

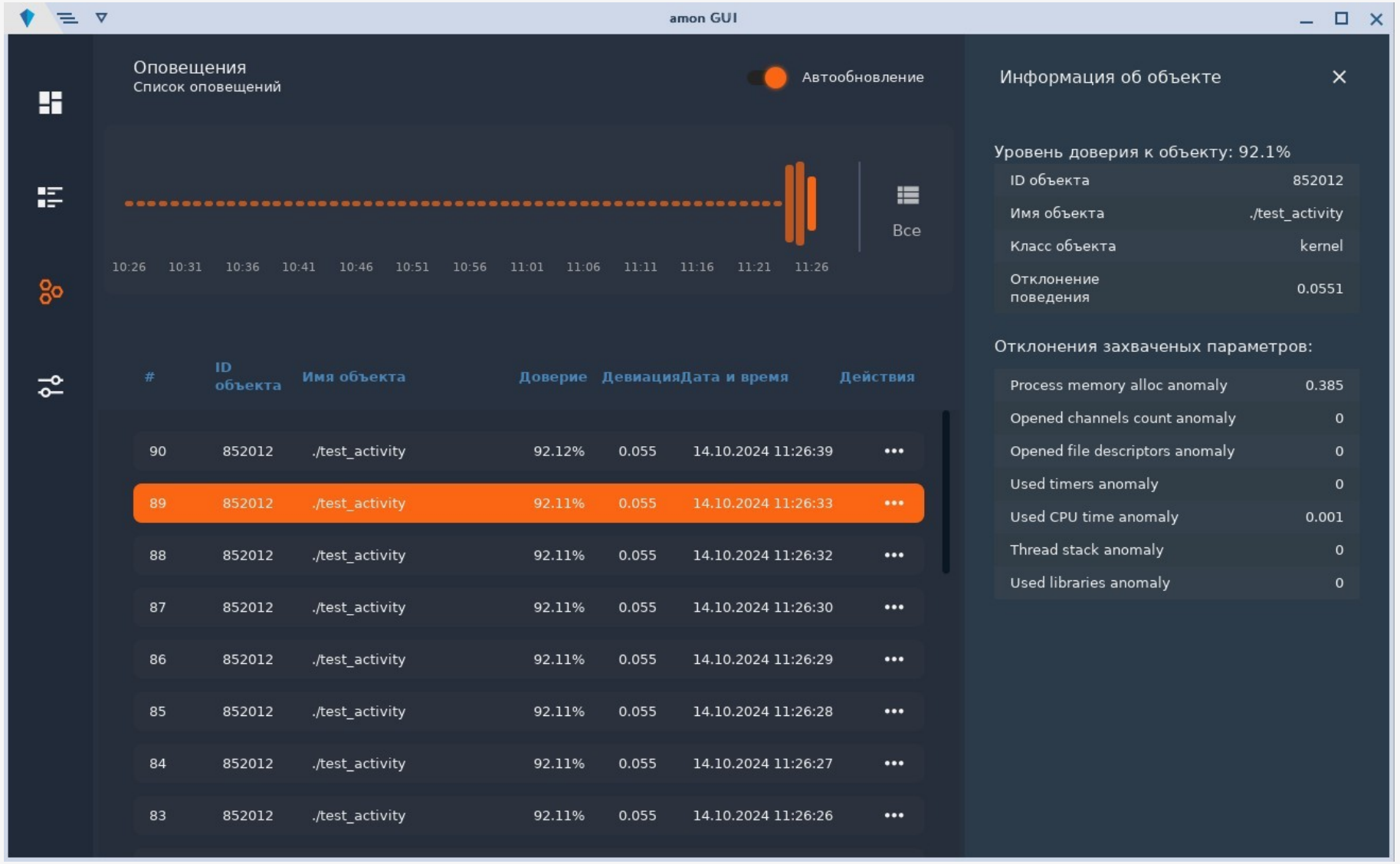
Внешний вид графического интерфейса amongui



Внешний вид графического интерфейса amongui



Внешний вид графического интерфейса amongui



СВД
Встраиваемые
Системы

Спасибо за внимание!

Константин Жвакин

Инженер-программист
Отдел операционных систем

ул. Кузнецовская, д. 19,
г. Санкт-Петербург
+7 (812) 346-89-56
www.kpda.ru
support@kpda.ru