Для подключения к сервису Cybert используется модуль для nginx ngx_http_cybert_s_module.so и вспомогательный демон cybert_s_hlprd. Ниже краткая инструкция для CentOS7.

1) Установка и настройка cybert_s_hlprd.

Данный шаг необходим в случае, если TCP трафик не проксируется через Servicepipe, а идет напрямую.

CentOS7:
Необходимо установить и настроить пакет cybert_s_hlprd-1.00-2.el7.x86_64.rpm.
Окружение для cybert_s_hlprd.service ищется тут:
/etc/sysconfig/cybert_s_hlprd

Пример содержимого:
```
OPTIONS="-s /var/run/cybert_s_hlprd.sock -d -i enp0"
```

где /var/run/cybert_s_hlprd.sock - общий путь для взаимодействия с модулем, enp0 - интерфейс через который приходят запросы

```
# systemctl enable cybert_s_hlprd
# systemctl start cybert_s_hlprd
```

Ubuntu/Debian:
Необходимо установить и настроить пакет cybertshlprd_1.0-3.deb. После установки внести изменения в юнит через:
```
# systemctl edit cybert_s_hlprd
```
или вручную. Пример изменений:
```
[Service]
Environment=OPTIONS='-s /var/run/cybert_s_hlprd.sock -d -i enp0'
```

где /var/run/cybert_s_hlprd.sock - общий путь для взаимодействия с модулем, enp0 - интерфейс через который приходят запросы

```
# systemctl enable cybert_s_hlprd
# systemctl start cybert_s_hlprd
```

Другие дистрибутивы:
Произвести установку и настройку для вашего дистрибутива вручную, используя файлы одного из пакетов.

2) Установка и настройка ngx_http_cybert_s_module.so.

Для установки нужной версии модуля можно установать предсобранный rpm пакет, либо соответствующий файл ngx_http_cybert_s_module.so.<ngx_version> переименовать в ngx_http_cybert_s_module.so и положить в нужную директорию.

Пример настрек конфигарации:

```
load_module /path/to/ngx_http_cybert_s_module.so;
http {
  cybert_s_backend_type multi_ssl;
  cybert_s_backend_address 91.206.127.81;
  cybert_s_backend_port 443;
  cybert_s_backend_relax_ms 50;
  cybert_s_backend_read_timeout_ms 500;

  cybert_s_helper1_address /var/run/cybert_s_hlprd.sock;
  cybert_s_helper1_relax_ms 5000;
  cybert_s_helper1_enabled on;
  cybert_s_only_system_cookies on;
  server {
    cybert_s_backend_sid $client_id;
    location / {
      cybert_s_backend_enabled on;
      cybert_s_backend_ssl_certificate /path/to/client.crt;
      cybert_s_backend_ssl_certificate_key /path/to/client.key;
      cybert_s_backend_ssl_certificate_ca /path/to/server.crt;
    }
    error_log /path/to/error.log;
    cybert_s_use_log $client_id;
  }
}
```

Ключи долны быть доступными для чтения nginx workers (обычно это непривелегированный пользователь).

3) Логирование

Для логирования фактического вердикта можно использовать переменную **cybert_s_verdict**, которая может принимать следующие значения:
"unknown" — запрос не проходил проверку, например, по причине отсутствия связи с сервисом Cybert;
"bot" — заблокировали, вернули 403;
"checking" — запрос отправлен на дополнительную проверку, вернули 307;
"human" — запрос пропустили дальше;

Для логирования классификации (разметки) пользователя на стороне ServicePipe можно использовать переменную **cybert_s_label**, которая может принимать следующие значения:
"unknown"
"bot"
"checking"
"human"
"white"

"black"
"block"

4) Описание опций модуля

```
cybert_s_backend_enabled on;
[main|server|location]
(default: off)
Включить работу модуля.

cybert_s_backend_type multi_ssl;
[main|server|location]
(default: must be determined)
Обязательная опция

cybert_s_backend_address 91.206.127.81;
[main|server|location]
(default: must be determined)
Адрес сервера Servicepipe.

cybert_s_backend_port 443;
[main|server|location]
(default: must be determined)
Порт для подключения к серверу Servicepipe.

cybert_s_backend_relax_ms 50;
[main|server|location]
(default: 5000)
Время в течении которого не будет производиться попыток подключения к
серверу Servicepipe после неудачи.

cybert_s_backend_read_timeout_ms 500;
[main|server|location]
(default: 500)
Максимальное время ожидания ответа от сервера ServicePipe включая
round-trip time.

cybert_s_helper1_enabled on;
[main|server|location]
(default: off)
Нужно включить, если будет использоваться демон cybert_s_hlprd

cybert_s_helper1_address /var/run/cybert_s_hlprd.sock;
[main|server|location]
(default: must be determined)
Адрес unix socketa для коммуникации с демоном cybert_s_hlprd.
Требуется указывать, если демон включен.
```

cybert_s_helper1_relax_ms 5000;
[main|server|location]
(default: 5000)
Время в течении которого не будет производиться попыток коммуникации с демоном cybert_s_hlprd после неудачи.

cybert_s_only_system_cookies on;
[main|server|location]
(default: off)
Если включено, в ServicePipe не будут переданы никакие куки кроме необходимых, выставленных самим ServicePipe

cybert_s_exclude_cookie cookie_name0 cookie_nameN;
[main|server|location]
(default: none)
Имена cookie, которые будут вырезаны в nginx и не будут переданы в ServicePipe. Обычно проще использовать cybert_s_only_system_cookies.

cybert_s_exclude_header header_name0 header_nameN;
[main|server|location]
(default: none)
Имена заголовков, которые будут вырезаны в nginx и не будут переданы в ServicePipe. Допускается вырезать только кастомные заголовки.

cybert_s_backend_sid sid;
[server|location]
(default: must be determined)
Обязательный параметр. Уникальный id клиента, который выдается ServicePipe.

cybert_s_monitoring_mode on;
[main|server|location]
(default: off)
Режим мониторинга: переменные логирования проставляются, но все запросы пропускаются как есть

cybert_s_use_log sid;
[main|server|location]
(default: read notice)
Данная опция позволяет модулю анализировать лог nginx для выявления эксплуатаций на уровне SSL/TLS. Это позволяет блокировать атаки, направленные на загрузку nginx через SSL хендшейки без фактической отправки запроса. Данная опция имеет смысл только если трафик проходит через инфраструктуру Servicepipe.

5) Оценка влияния включенного модуля но производительность nginx

Типовое снижение производительности nginx для защищаемых server/location	3-10%
Оценка роста потребления ОЗУ	до 10 MiB на nginx worker
Оценка трафика в сторону ServicePipe[1]	примерно 5 Mbit на 1000 RPS
Оценка задержки процессинга запроса	не менее round-trip delay до ServicePipe

[1] — дальнейшее совершенствование технологии направлено в том числе на уменьшение значения