????????? ???????????? ??????? ????? ????????????? ??????-??????

Дата обновления: **08.07.2026**

За основу написания статьи послужила страница в справке: https://support.kaspersky.ru/kedr-expert-on-premise/8.1/322641

Для начала создадим две A записи, которые будут доступны для агентов за пределами периметра организации:

- Запись, которую мы будем указывать в качестве сервера сбора телеметрии, например, **telemetry.<example.com>**
- Запись, которую мы будем указывать в качестве сервера реагирования, например, **response.<example.com>**

Далее подготовим рабочий каталог для сертификатов:

```
sudo mkdir -p /etc/nginx/mtls && cd /etc/nginx/mtls
```

Создадим самоподписанный корневой сертификат (Root CA):

```
openssl genrsa -out ca.key 4096

openssl req -x509 -new -nodes -key ca.key -sha256 -days 3650 \
   -subj "/C=US/O=ExampleCorp/OU=IT/CN=Example Root CA" -out ca.crt
# Укажите свои параметры владельца
```

Создадим сертификат для сервера реагирования:

```
openssl genrsa -out response.key 2048

openssl req -new -key response.key \
   -subj "/C=US/O=ExampleCorp/OU=IT/CN=response.<example.com>" -out response.csr
# Укажите свои параметры владельца
```

```
openssl x509 -req -in response.csr -CA ca.crt -CAkey ca.key -CAcreateserial \
   -out response.crt -days 825 -sha256 \
   -extfile <(printf
"subjectAltName=DNS:response.<example.com>\nkeyUsage=digitalSignature,keyEncipherment\nextende
dKeyUsage=serverAuth")
# Укажите свой SAN
```

По аналогии создадим сертификат для сервера телеметрии:

```
openssl genrsa -out telemetry.key 2048

openssl req -new -key telemetry.key \
   -subj "/C=US/O=ExampleCorp/OU=IT/CN=telemetry.<example.com>" -out telemetry.csr
# Укажите свои параметры владельца

openssl x509 -req -in telemetry.csr -CA ca.crt -CAkey ca.key -CAcreateserial \
   -out telemetry.crt -days 825 -sha256 \
   -extfile <(printf
"subjectAltName=DNS:telemetry.<example.com>\nkeyUsage=digitalSignature,keyEncipherment\nextend
edKeyUsage=serverAuth")
# Укажите свой SAN
```

Далее в OSMP экспортируем сертификат для Endpoint Agent: **Параметры --> Тенанты --> Root tenant --> Параметры --> Сертификаты** и экспортируем сертификат Endpoint Agent, который размещаем в ранее созданной директории **/etc/nginx/mtls**

Из экспортируемого pfx сертификата Endpoint Agent извлекаем данные:

```
openssl pkcs12 -in certificate.pfx -cacerts -nokeys -out ca-xdr.crt
openssl pkcs12 -in certificate.pfx -clcerts -nokeys -out client.crt
openssl pkcs12 -in certificate.pfx -nocerts -nodes -out client.key
```

Далее установим права доступа:

```
chmod 600 *.key
chown root:root *.crt *.key *.csr
```

Далее установим Nginx:

```
sudo apt update
sudo apt install nginx
```

```
nginx -V
# Проверьте наличие модуля http_ssl_module (по умолчанию включен в Nginx)
```

Создайте файл конфигурации nginx по пути **/etc/nginx/conf.d/asproxy.conf** и отредактируйте его:

Пример конфигурационного файла

```
# Использование map для поддержки WebSocket
map $http_upgrade $connection_upgrade {
   default upgrade;
   ''      close;
}

server {
   listen 443 ssl http2;
   listen 9443 ssl;
   server_name response.<example.com>; # Укажите свой сервер

   ssl_certificate         /etc/nginx/mtls/resposne.crt;
   ssl_certificate_key     /etc/nginx/mtls/resposne.key;
   ssl_trusted_certificate /etc/nginx/mtls/ca.crt;
   ssl_client_certificate  /etc/nginx/mtls/ca-xdr.crt;
   ssl_verify_client       on;
   ssl_verify_depth        2;
   ssl_protocols           TLSv1.2 TLSv1.3;
   ssl_session_cache       shared:SSL:10m;

   proxy_set_header X-Client-Verify $ssl_client_verify;
   proxy_set_header X-Client-DN     $ssl_client_s_dn;
   proxy_set_header X-Client-Cert   $ssl_client_escaped_cert;

   proxy_http_version                1.1;
   proxy_set_header Host             agentserver.<smp_domain>; # Укажите адрес
agentserver
   proxy_set_header X-Real-IP        $remote_addr;
   proxy_set_header X-Forwarded-For   $proxy_add_x_forwarded_for;
   proxy_set_header X-Forwarded-Proto $scheme;
```

```
    proxy_connect_timeout 5s;
    proxy_send_timeout    60s;
    proxy_read_timeout    60s;
    send_timeout          60s;

    proxy_ssl_server_name on;
    proxy_ssl_name        agentserver.<smp_domain>; # Укажите адрес agentserver
    proxy_ssl_verify      off;
    proxy_ssl_protocols   TLSv1.2 TLSv1.3;

    proxy_ssl_certificate     /etc/nginx/mtls/client.crt;
    proxy_ssl_certificate_key /etc/nginx/mtls/client.key;

     location / {
      proxy_pass https://agentserver.<smp_domain>; # Укажите адрес agentserver
 }

      location ^~ /vthost/sessions {
      proxy_pass https://agentserver.<smp_domain>:9443; # Укажите адрес agentserver

      proxy_set_header Upgrade    $http_upgrade;
      proxy_set_header Connection $connection_upgrade;

      proxy_read_timeout  3600s;
      proxy_send_timeout  3600s;

      proxy_request_buffering off;
      proxy_buffering         off;
 }

    access_log /var/log/nginx/response.access.log;
    error_log  /var/log/nginx/response.error.log warn;

    error_page 495 496 497 = /__mtls_error;
    location = /__mtls_error { return 403; }
}
server {
    listen 443 ssl http2;
```

```
    server_name telemetry.<example.com>; # Укажите свой сервер

    ssl_certificate         /etc/nginx/mtls/telemetry.crt;
    ssl_certificate_key     /etc/nginx/mtls/telemetry.key;
    ssl_trusted_certificate /etc/nginx/mtls/ca.crt;
    ssl_client_certificate  /etc/nginx/mtls/ca-xdr.crt;
    ssl_verify_client       on;
    ssl_verify_depth        2;
    ssl_protocols           TLSv1.2 TLSv1.3;
    ssl_session_cache       shared:SSL:10m;

    proxy_set_header X-Client-Verify $ssl_client_verify;
    proxy_set_header X-Client-DN     $ssl_client_s_dn;
    proxy_set_header X-Client-Cert   $ssl_client_escaped_cert;

    proxy_http_version                1.1;
    proxy_set_header Host             in.ehoqhrjot93jdnmw23jqkxyfq.kuma.<smp_domain>; #
Укажите свой адрес коллектора EDR
    proxy_set_header X-Real-IP        $remote_addr;
    proxy_set_header X-Forwarded-For   $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    proxy_connect_timeout 10s;
    proxy_send_timeout    300s;
    proxy_read_timeout    300s;
    send_timeout          300s;

    proxy_ssl_server_name on;
    proxy_ssl_name        in.ehoqhrjot93jdnmw23jqkxyfq.kuma.<smp_domain>; # Укажите свой
адрес коллектора EDR
    proxy_ssl_verify      off;
    proxy_ssl_protocols   TLSv1.2 TLSv1.3;



    proxy_ssl_certificate     /etc/nginx/mtls/client.crt;
    proxy_ssl_certificate_key /etc/nginx/mtls/client.key;

    proxy_request_buffering off;
    proxy_buffering         off;
```

```
   location / {
     proxy_pass https://in.ehoqhrjot93jdnmw23jqkxyfq.kuma.<smp_domain>:443; # Укажите свой
адрес коллектора EDR
   }

   access_log /var/log/nginx/telemetry.access.log;
   error_log  /var/log/nginx/telemetry.error.log warn;

   error_page 495 496 497 = /__mtls_error;
   location = /__mtls_error { return 403; }
}
```

Далее в политике для KES'ов в настройках Endpoint Detection and Response Expert (on-premise) указываем в качестве сервера сбора телеметрии - **https://telemetry.<example.com>** на 443 порту, а в настройках подключения указываем созданный сертификат **telemetry.crt**, и для дополнительной защиты подставляем ранее экспортируемый с OSMP pfx сертификат для Endpoint Agent. Для сервера реагирования указываем - **response.<example.com>** на 443 порту, а в настройках подключения - созданный сертификат **response.crt**, и для дополнительной защиты всё тот же pfx сертификат для Endpoint Agent.

Revision #6
Created 8 July 2026 12:13:56 by Кирилл
Updated 8 July 2026 13:15:01 by Кирилл