

# Элементы безопасности информационных систем



Сергей  
Мясников



**Сергей Мясников**  
Сетевой Инженер, T-Systems  
ex Mail.ru Group



## План модуля

1. Работа в терминале, лекция 1
2. Работа в терминале, лекция 2
3. Операционные системы, лекция 1
4. Операционные системы, лекция 2
5. Файловые системы
6. Компьютерные сети, лекция 1
7. Компьютерные сети, лекция 2
8. Компьютерные сети, лекция 3
9. **Элементы безопасности информационных систем**



## План занятия

1. [AAA: Аутентификация, Авторизация, Аккаунтинг](#)
2. [Пароли. OTP. Yubikey](#)
3. [Шифрование и Сертификаты, HTTPS](#)
4. [SSH](#)
5. [Firewall. NAT. ACL](#)
6. [Pentest: тестирование на проникновение](#)
7. [Домашнее задание](#)



**AAA: Аутентификация,  
Авторизация, Аккаунтинг**

# Аутентификация: кто?

Задача – идентифицировать пользователя или объект.

Примеры:

1. Логин+пароль;
2. Биометрия – отпечатки пальцев, сканирование ладони, голос, face-id;
3. Ключ шифрования – SSH, HTTPS.

# Авторизация: что разрешено делать?

**Примеры действий** – чтение, чтение и запись, исполнение программ.

**Примеры объектов** – файл, каталог, принтер, URL, проект в Jira, AWS тег.

**Групповые политики** – разделение на группы и назначение каждой группе отдельного набора прав.

Варианты разделения на группы:

- разделение по объектам: группа принтеры, файловый сервер и т. д
- разделение по пользователям: группа бухгалтерия, администраторы и т.д.

Основной инструмент – корпоративный каталог LDAP, например, Microsoft ActiveDirectory.

---

# Аккаунтинг: лог событий

Учет всех действий пользователей: попытки входа, запросы на чтение и т.д.

## Основные задачи:

- Выявление попыток несанкционированного доступа;
- Помощь в расследовании инцидента;
- Оценка ущерба;
- Устранение последствий.

## Основные инструменты:

- Системы логирования и визуализации: Syslog-NG, ELK, Grafana;
- На сетевом оборудовании – Tascacs сервер;
- [SEIM](#) системы – комплексное решение.



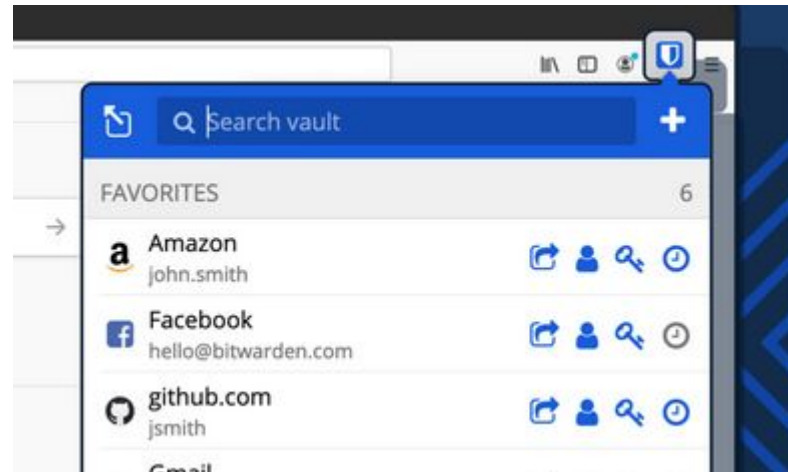


# Пароли. ОТР. Yubikey

# Где хранить пароли?

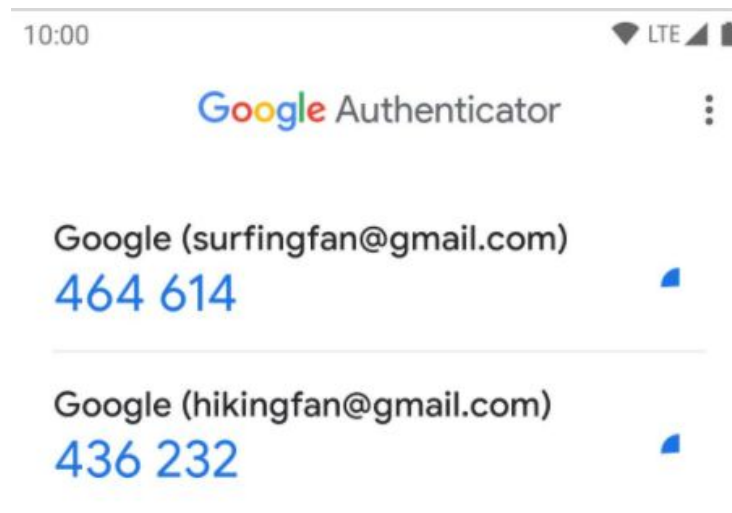
- На бумажке в сейфе (неудобно, но очень надежно);
- Онлайн менеджер паролей (удобно, ненадежно);
- Корпоративный сервер паролей (оптимальное решение).  
Bitwarden - <https://github.com/bitwarden/server>

## Пример настройки



# ОТР – 2-х факторная аутентификация

- One time password – сервис одноразовых паролей;
- СМС-коды или мобильное приложение;
- Важно использовать разные устройства – телефон+компьютер;
- Наиболее популярные сервисы: Google Authenticator, Microsoft Authenticator.



---


# Yubikey – U2F USB криптотокен

- Генерирует OTP-код при физическом нажатии;
- Защищенное хранилище SSH-ключей;
- SSH-агент, работает с кроссплатформенной библиотекой [OpenSC](#).



[Протокол FIDO U2F \(Universal 2nd Factor\)](#)

[YubiKey в Linux](#)



# Шифрование и сертификаты, HTTPS

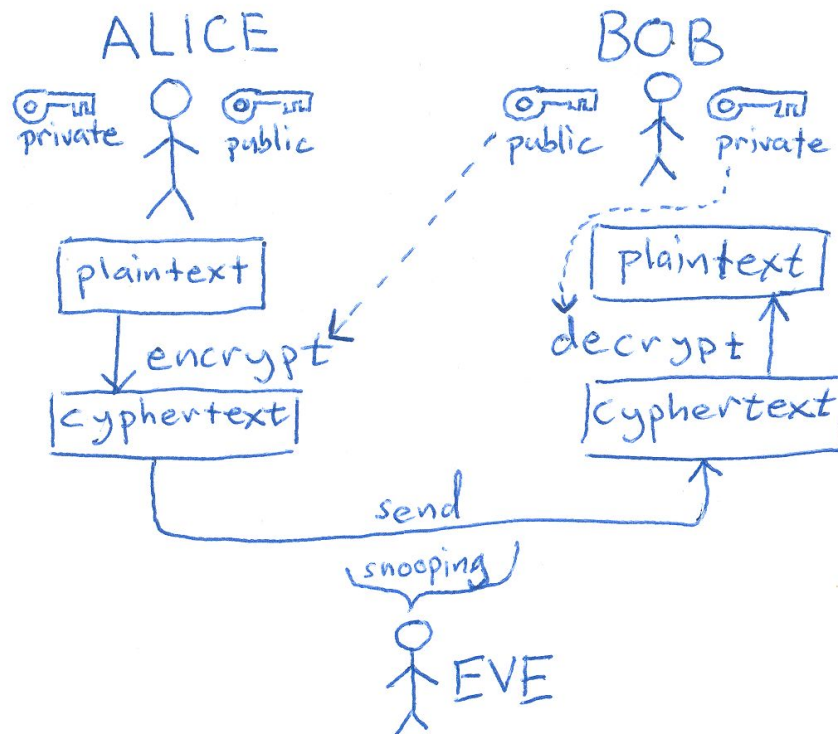
---

# Симметричное шифрование

- Один ключ для шифрования-дешифрования;
- Ключи нельзя передавать по открытым каналам;
- Очень высокая скорость работы;
- Пример: DES, AES шифр.

# Асимметричное шифрование

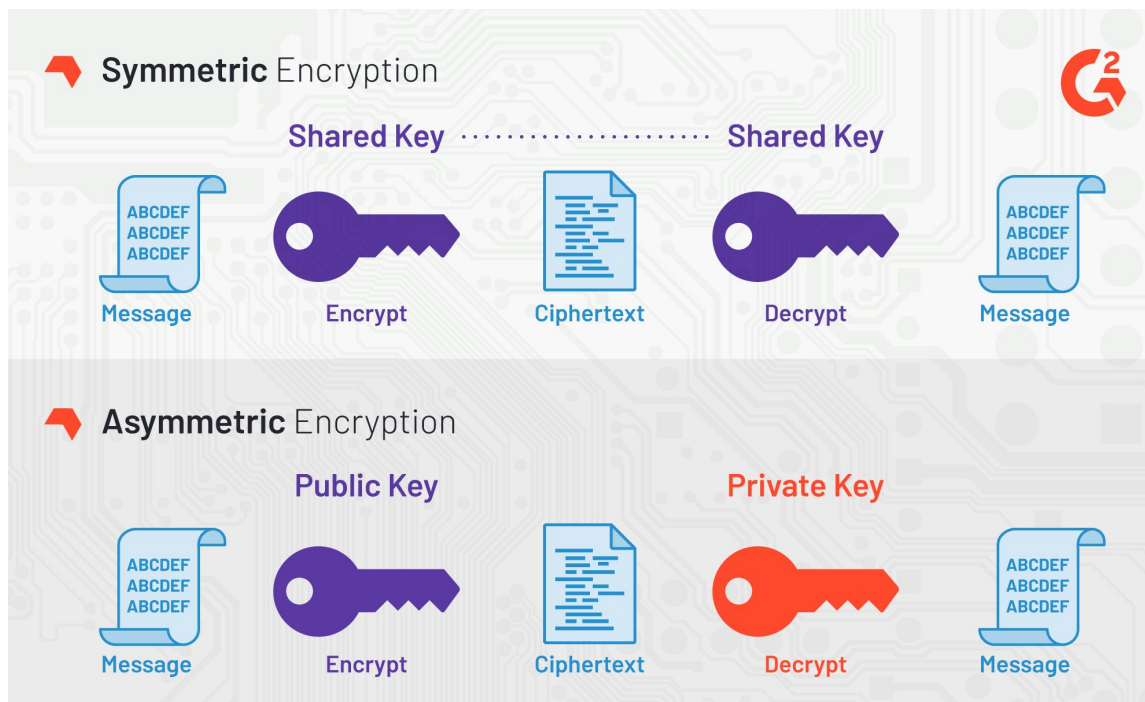
- 2 пары ключей у каждой стороны;
- Public ключ передается по открытым каналам, Private ключ – секретный;
- Секретность строится на основе Односторонней функции



# Сравнение симметричного и асимметричного шифрование

**Симметричное:** один ключ для шифрования-дешифрования, ключ секретный.

**Асимметричное:** один ключ для дешифрования (секретный), второй для шифрования (открытый). У каждой стороны по паре ключей.

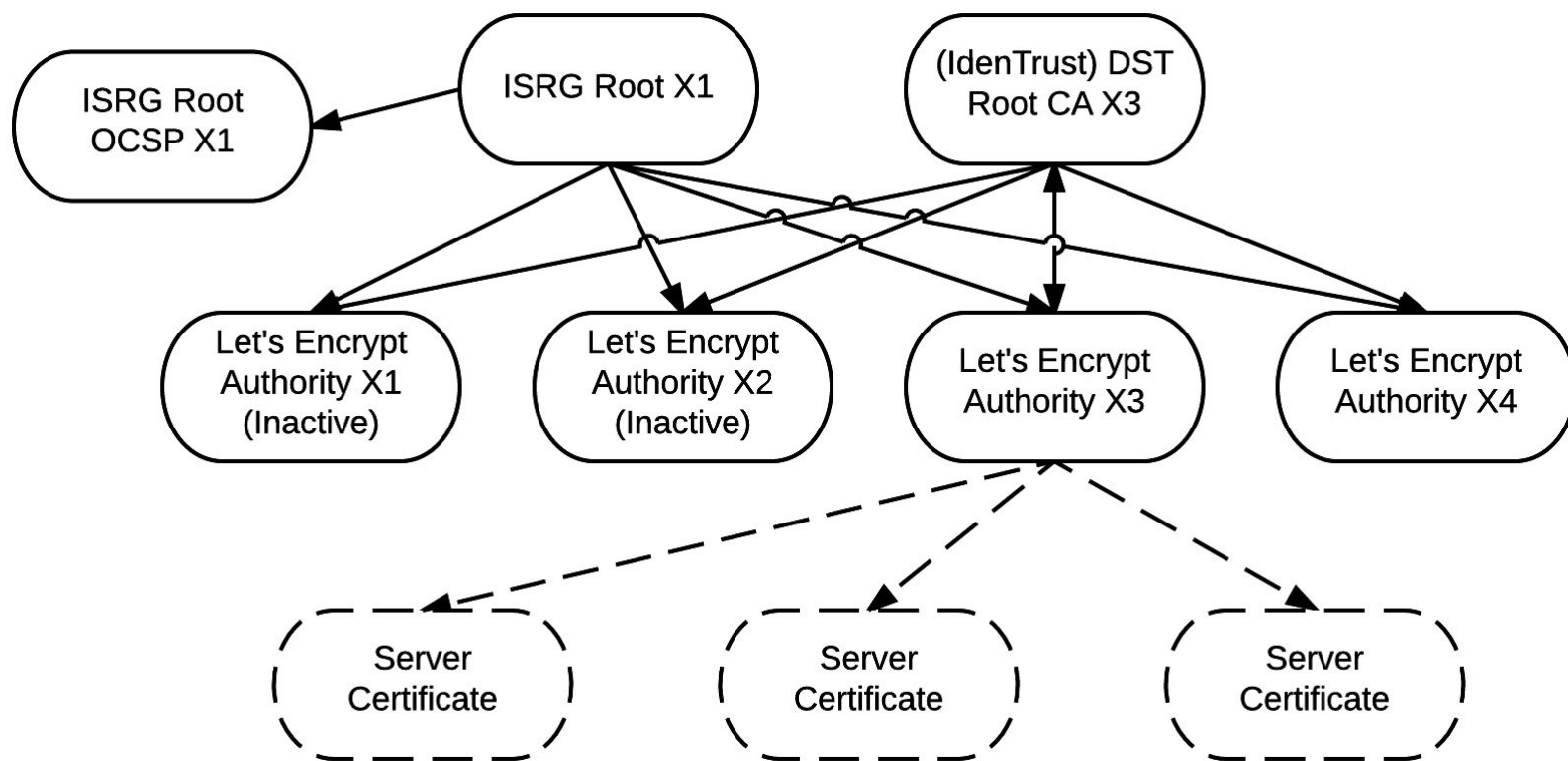




# HTTPS – SSL/TLS сертификаты и Root CA

В основе цепочка доверия серверов сертификации. Ответственность распределяется иерархически, аналогично иерархии серверов в DNS.

Пример: бесплатный центр сертификации Let's Encrypt.



# Self signed certificate

```
sudo apt install apache2
sudo a2enmod ssl
sudo systemctl restart apache2

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
-keyout /etc/ssl/private/apache-selfsigned.key \
-out /etc/ssl/certs/apache-selfsigned.crt \
-subj "/C=RU/ST=Moscow/L=Moscow/O=Company Name/OU=Org/CN=www.example.com"

sudo vim /etc/apache2/sites-available/your_domain_or_ip.conf
<VirtualHost *:443>
    ServerName your_domain_or_ip
    DocumentRoot /var/www/your_domain_or_ip

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>

sudo mkdir /var/www/your_domain_or_ip
sudo vim /var/www/your_domain_or_ip/index.html
<h1>it worked!</h1>

sudo a2ensite your_domain_or_ip.conf
sudo apache2ctl configtest
sudo systemctl reload apache2
```

# Проверка TLS настроек web-сервера

<https://github.com/drwetter/testssl.sh>

```
git clone --depth 1 https://github.com/drwetter/testssl.sh.git
cd testssl.sh
```

```
# быстрый тест
./testssl.sh -e --fast --parallel https://www.google.com/
```

```
# проверить сайт на уязвимости
./testssl.sh -U --sneaky https://www.google.com/
```

Testing vulnerabilities

Heartbleed (CVE-2014-0160)	not vulnerable (OK), no heartbeat extension
CCS (CVE-2014-0224)	not vulnerable (OK)
Ticketbleed (CVE-2016-9244), experiment.	not vulnerable (OK)
ROBOT	not vulnerable (OK)
Secure Renegotiation (RFC 5746)	supported (OK)
Secure Client-Initiated Renegotiation	not vulnerable (OK)
CRIME, TLS (CVE-2012-4929)	not vulnerable (OK)

[Примеры testssl](#)



# SSH

# SSH: управление ключами

```
# установка sshd сервера
apt install openssh-server
systemctl start sshd.service
systemctl enable sshd.service

# генерим ключи, /home/<username>/.ssh/id_rsa - приватный ключ
# id_rsa.pub - публичный ключ
ssh-keygen

# копируем публичный ключ на удаленный сервер
ssh-copy-id my_user@192.168.1.100

# или копируем вручную в файл authorized_keys
echo public_key_string >> ~/.ssh/authorized_keys

# подключаемся по стандартному ключу
ssh my_user@192.168.1.100

# подключаемся по нестандартному ключу
ssh -i ~/.ssh/some_server.key my_user@192.168.1.100

# проверка SSH шифров
sudo apt install -y ssh-audit
ssh-audit localhost
```

## [Пример настройки SSH](#)

# SSH: config файл

```
mkdir -p ~/.ssh && chmod 700 ~/.ssh
touch ~/.ssh/config && chmod 600 ~/.ssh/config
#-----общая структура файла ~/.ssh/config
Host hostname1
    SSH_OPTION value
Host *
    SSH_OPTION value
#-----Пример конфига
Host my_server
    HostName 192.168.1.100
    IdentityFile ~/.ssh/some_server.key
    User my_user
    #Port 2222
    #StrictHostKeyChecking no
Host *
    User default_username
    IdentityFile ~/.ssh/id_rsa
    Protocol 2

# пример ProxyJump
Host myserver
    HostName myserver.example.com
    User virag
    IdentityFile /users/virag/keys/ed25519
    ProxyJump jump
Host jump
    HostName jump.example.com
    User default
```

[ssh\\_config manual](#)

[ProxyJump безопаснее чем SSH agent forwarding](#)

# SSH: проблема с known\_host

```
ssh my_server
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
The fingerprint for the RSA key sent by the remote host is
5c:9b:16:56:a6:cd:11:10:3a:cd:1b:a2:91:cd:e5:1c.
```

Опция хэширования known\_host  
HashKnownHosts yes

# список известных публичных ключей

```
cat ~/.ssh/known_hosts
```

# проверить хост

```
ssh-keygen -F my_server
```

# удаляем старый ключ

```
ssh-keygen -R my_server
```

# временно отключить проверку known\_hosts

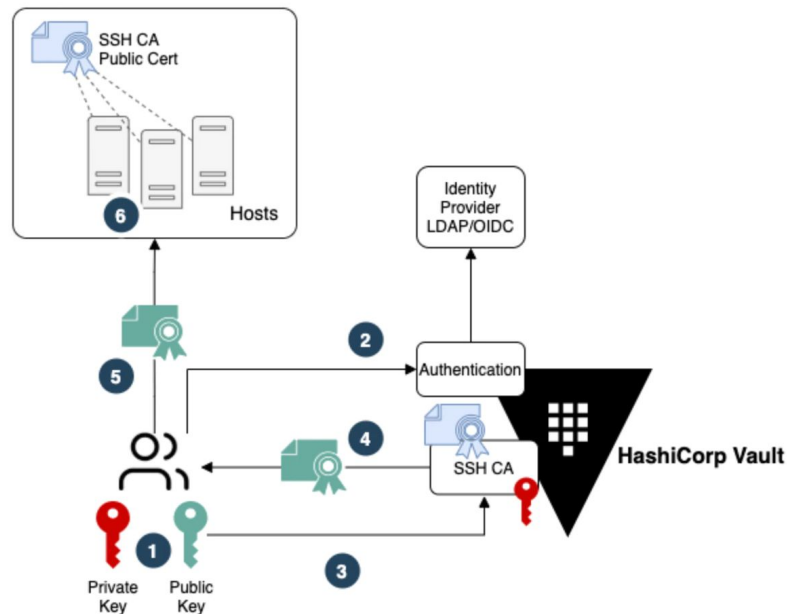
```
ssh -o StrictHostKeyChecking=no my_server
```

# сканировать и сохранить публичные ключи

```
ssh-keyscan server_ip
```

```
ssh-keyscan -H <host> >> ~/.ssh/known_hosts # -H for Hashed
```

# HashiCorp Vault



SSH Certificate Authentication Workflow

[Github Hashicorp Vault](#)

[Managing SSH Access at Scale with HashiCorp Vault](#)

[Управление секретами при помощи Hashicorp Vault в Авито](#)

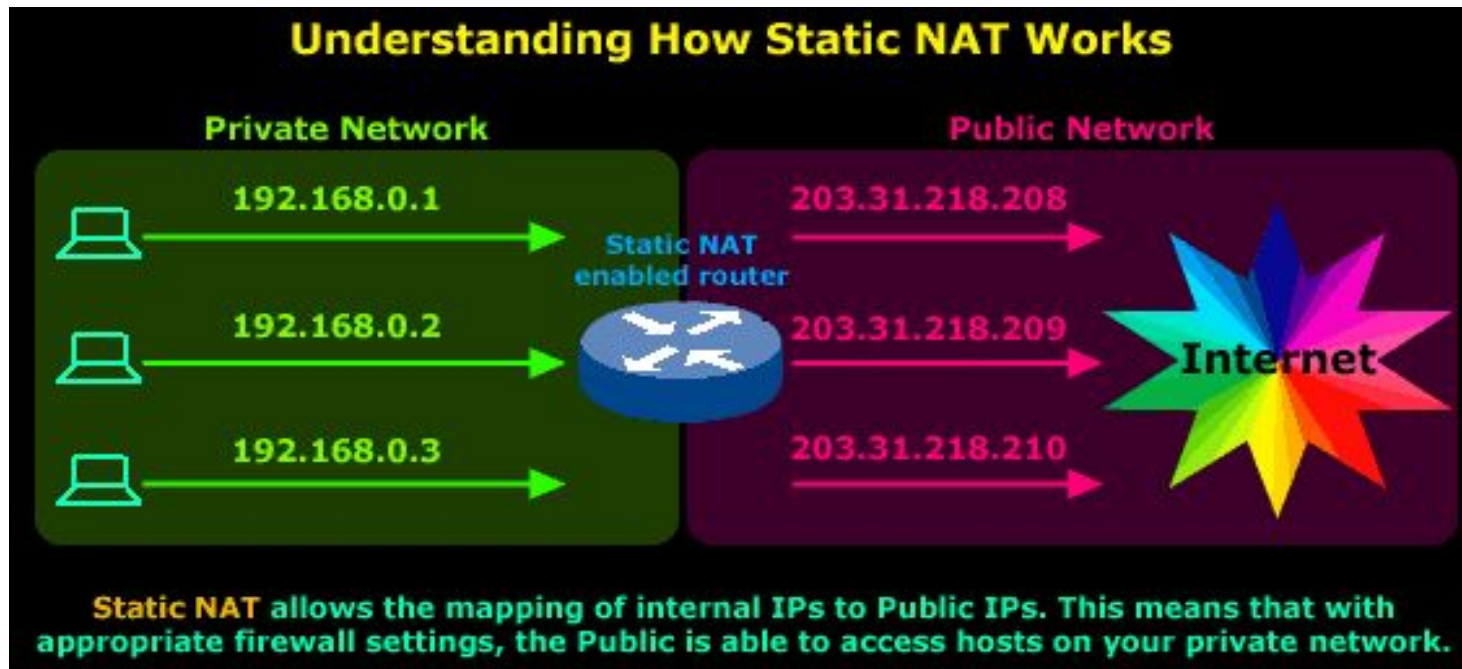




# Firewall. NAT. ACL

# Статический NAT

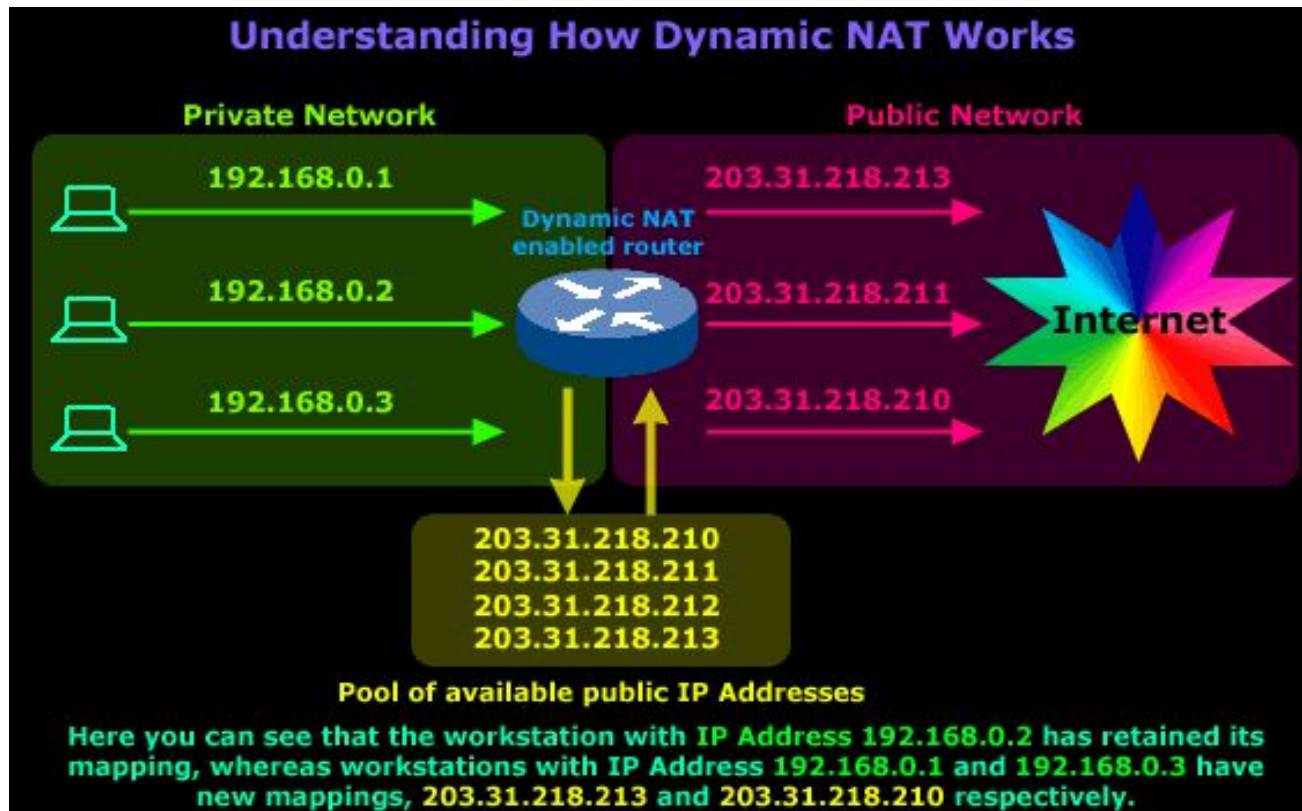
Трансляция IP адресов настроена заранее.



[Источник](#)

# Динамический NAT

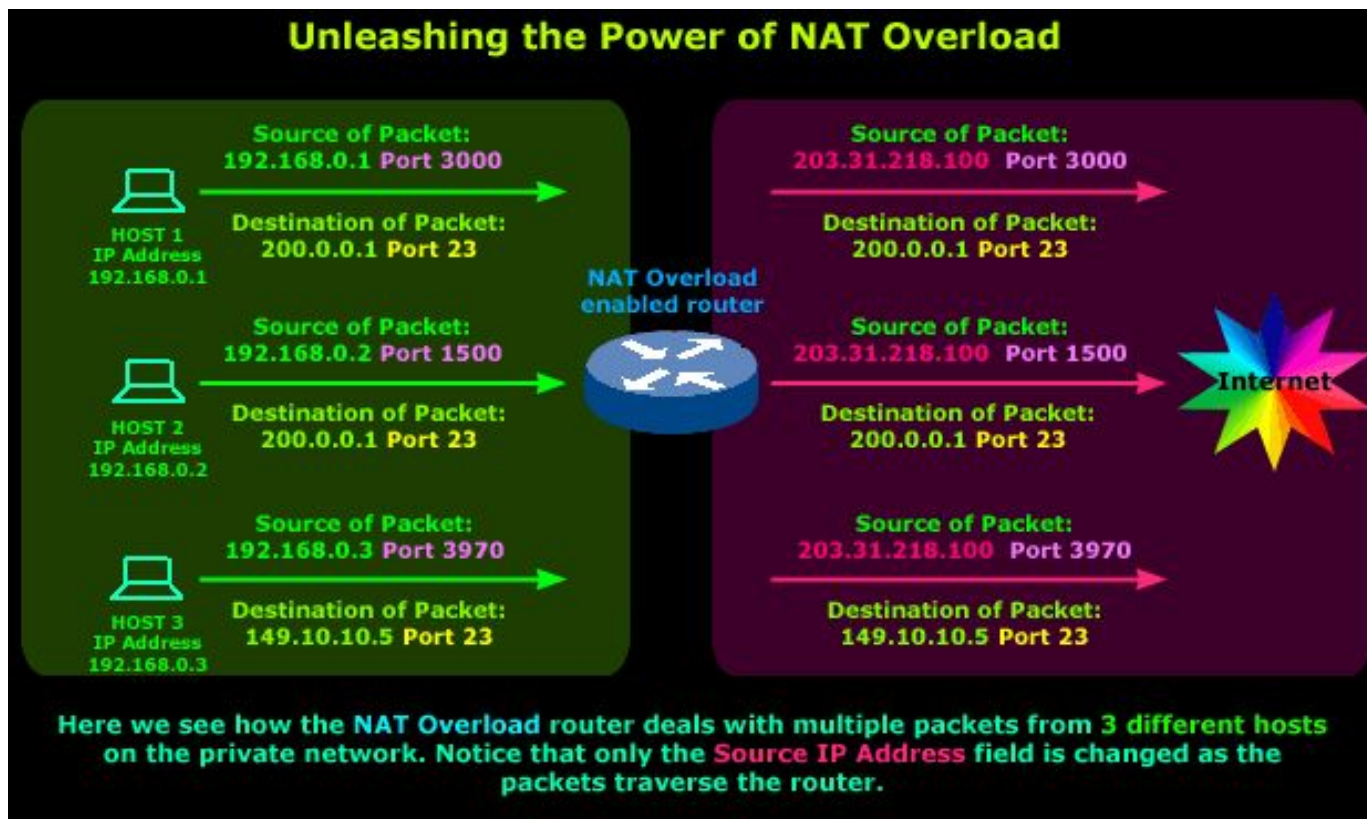
IP адреса выделяются из NAT пула динамически.



[Источник](#)

# NAT Overload

Один публичный адрес используется для всей внутренней сети.  
Трансляция IP адресов привязывается к L4 портам.



# Firewall в linux – iptables

```
apt install iptables

# список правил таблицы filter
iptables -t filter -L
iptables -t filter -L -v      # посмотрит кол-ва пакетов

# Удалить все правила
iptables -F

# заблокировать пакеты от Source IP 1.1.1.1
iptables -A INPUT -s 1.1.1.1 -j DROP

# заблокировать исходящий UDP трафик порт 53
iptables -A OUTPUT -p udp --dport 53 -j DROP

# Удалить правило
iptables -D OUTPUT -p udp --dport 53 -j DROP

# Сохранить правила
apt install iptables-persistent
iptables-save > /etc/iptables/rules.v4
Iptables-save > /etc/iptables/rules.v6

# Сохранить правила в Ubuntu
netfilter-persistent save
netfilter-persistent reload
```

# Firewall в linux – ufw

```
apt install ufw

# посмотреть настроенные правила
sudo ufw status verbose

# добавить правило для любых IP
sudo ufw allow allow ssh/tcp


# Добавить правило для определенной сети
sudo ufw allow from 172.30.0.7/24

# Удалить правило
sudo ufw delete allow from 172.30.0.7/24

# Удалить правило по номеру
sudo ufw status numbered
sudo ufw delete 2

# Настроить логирование
sudo ufw logging on
tail -f /var/log/ufw.log

# Включить фаервол
sudo ufw enable
```



# **Pentest: тестирование на проникновение**

# Kali Linux – pentest дистрибутив

Популярные инструменты:

1. Nmap – сканер портов, сетей;
2. Burp Suite – анализ web уязвимостей;
3. Metasploit Framework – тестирование эксплойтов;
4. Aircrack-ng – работа с Wifi сетями;
5. Autopsy – digital forensics анализ;
6. Wireshark – анализ трафика.

<https://tools.kali.org/tools-listing>





# нmap – сканер сети

```
apt install nmap

# живые хосты в сети - ping sweep
nmap -sP 192.168.1.0/24

# информация о хосте
sudo nmap -O scanme.nmap.org

# подробная информация о хосте
sudo nmap -A scanme.nmap.org

# traceroute
nmap -sn -Pn --traceroute 8.8.8.8

# ASN query
nmap --script asn-query 8.8.8.8

# Path MTU discovery
sudo nmap --script path-mtu 8.8.8.8
```

# tcpdump – сбор дампа трафика

```
apt install tcpdump

# интерфейс
tcpdump -i eth0

# список доступных интерфейсов
tcpdump -D

# кол-во пакетов
tcpdump -c 5 -i eth0

# запись в файл
tcpdump -w 0001.pcap -i eth0

# чтение из файла
tcpdump -r 0001.pcap

# фильтр только TCP port 22
tcpdump -i eth0 port 22

# фильтр Source или Destination IP адрес
tcpdump -i eth0 src 192.168.1.1
tcpdump -i eth0 dst 192.168.1.1
```

---

# Итоги

Сегодня мы:

- Рассмотрели принципы работы AAA, OTP пароли;
- Познакомились с принципами шифрования;
- Изучили методы работы с SSH;
- Рассмотрели принципы работы NAT;
- Познакомились с работой Firewall в linux;
- Познакомились с утилитами nmap, tcpdump.

---

# Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

---

## Дополнительные материалы

- [Руководство по подготовке к экзамену CISSP](#)
- [Awesome SSH проекты](#)
- [Teleport - альтернатива SSH](#)
- [ngrok - NAT traversal](#)
- [ssh-audit](#)
- [Mozilla SSL Configuration Generator](#)

**Задавайте вопросы и  
пишите отзыв о лекции!**

**Сергей Мясников**