Port Knocking for security

Допълнително ниво на защита за вашата мрежа **с** MikroTik RouterOS

Да започнем със защо!

- Защо точно аз водя този семинар в SoftUni?
- Защо съм избрал темата за port knocking?
  - Какво е port knocking и за какво служи?
  - Как се конфигурира port knocking сървър и клиент?
  - Може ли да се автоматизира?
- Защо правя демонстрации с MikroTik RouterOS?

- https://www.sli.do/ | Code: *#PortKnocking*

Добри Бояджиев

- **Бакалавър**, **Магистър и Доктор** в направление „Информатика и компютърни науки“ в УниБИТ, с обучения, посетени от 2000+ студенти, участие в 50+ конференции, семинари, работни групи (2009 г. ~ 2017 г.)
- Първият у нас MikroTik **сертифициран инструктор**, едновременно по двете програми за преподаватели, с 40+ обучения и 500+ сертифицирани курсисти (2014 г. ~)
- **Автор** на първата българска книгата „Основни на MikroTik RouterOS“, с тираж от 3000+ (2016 г. ~)

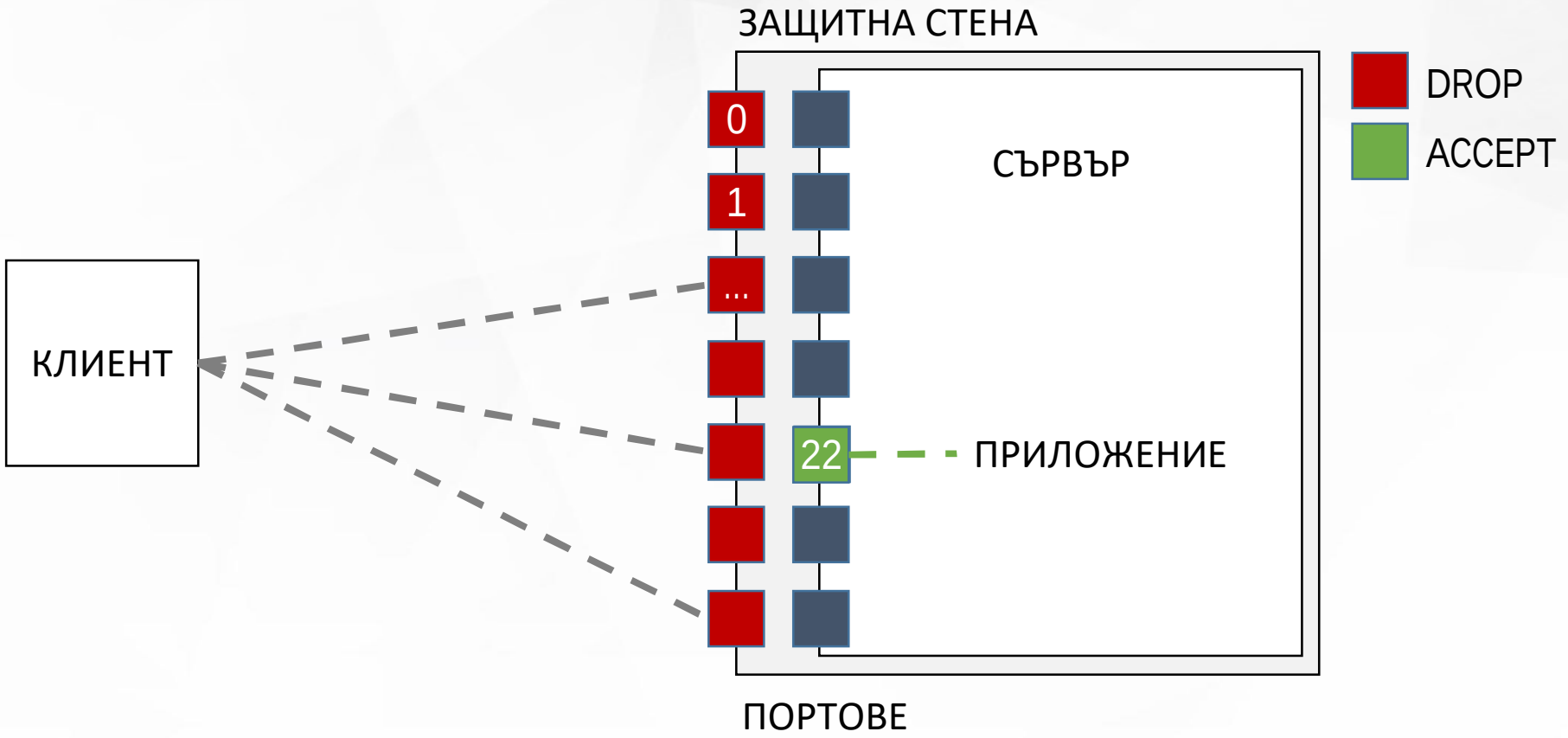
Добри Бояджиев

- **Координатор** по академичната програма на MikroTik, подпомогнал създаването на 5+ академии; обучил и сертифицирал 10+ преподаватели (2017 г. ~)
- Cloud Operation Manager в SAP Labs Bulgaria, с фокус върху облачните продукти от групата на Analytics and Intelligent Technologies, разработвани от капанията и хоствани, както на собствената облачна платформа (SAP Business Technology Platform), така и на другите ключови доставчици на облачни услуги (2017 г.~)

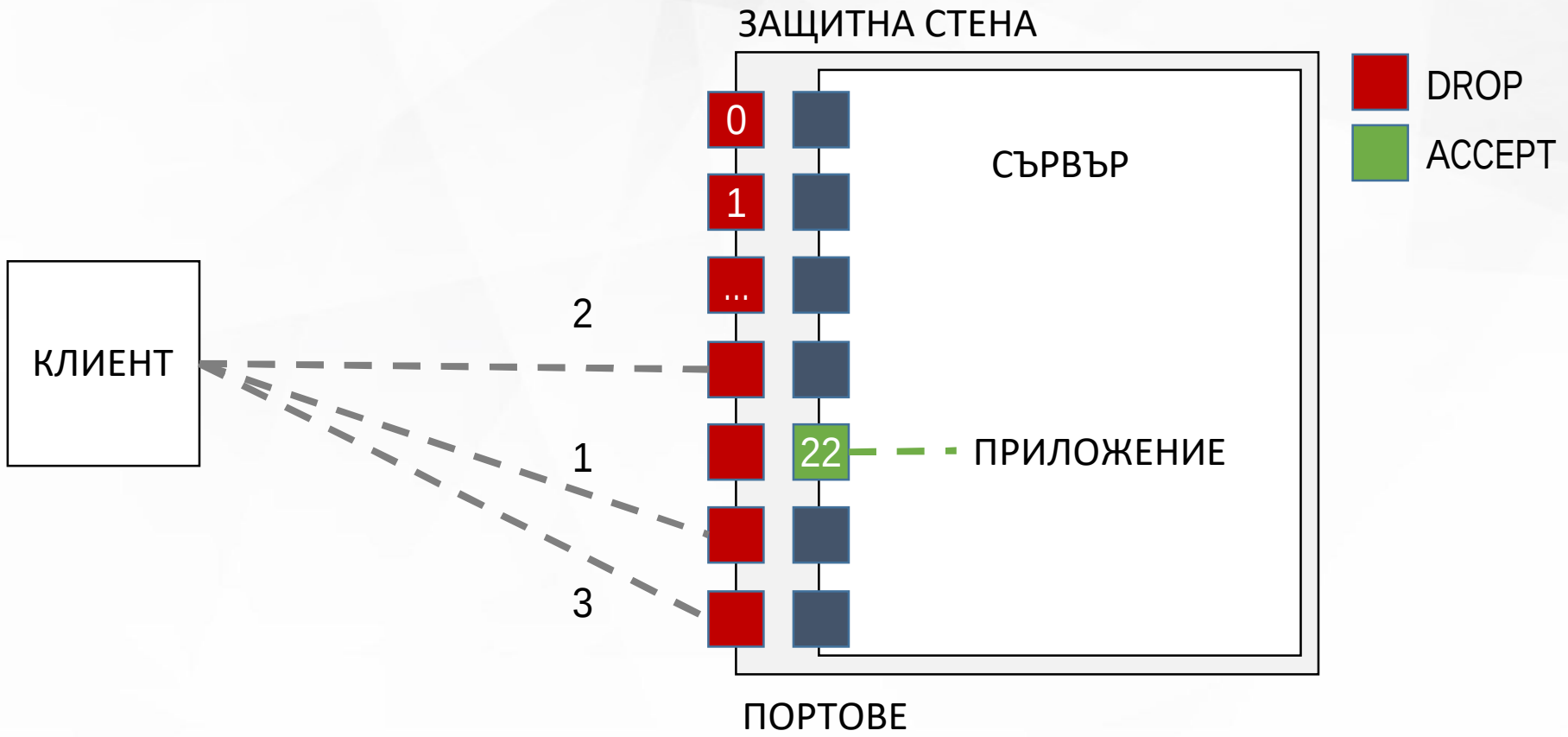
Цел и предназначение

- Port knocking-ът е:
  - **Метод за** последователно предаване на данни към затворени портове на мрежови устройства
  - **Неговата цел** е удостоверяване, което да осигури достъп до защитени услуги
- Това е механизъм, предназначен да:
  - Попречи на нападателя, който извърши сканиране на портове, да открива потенциално уязвими услуги

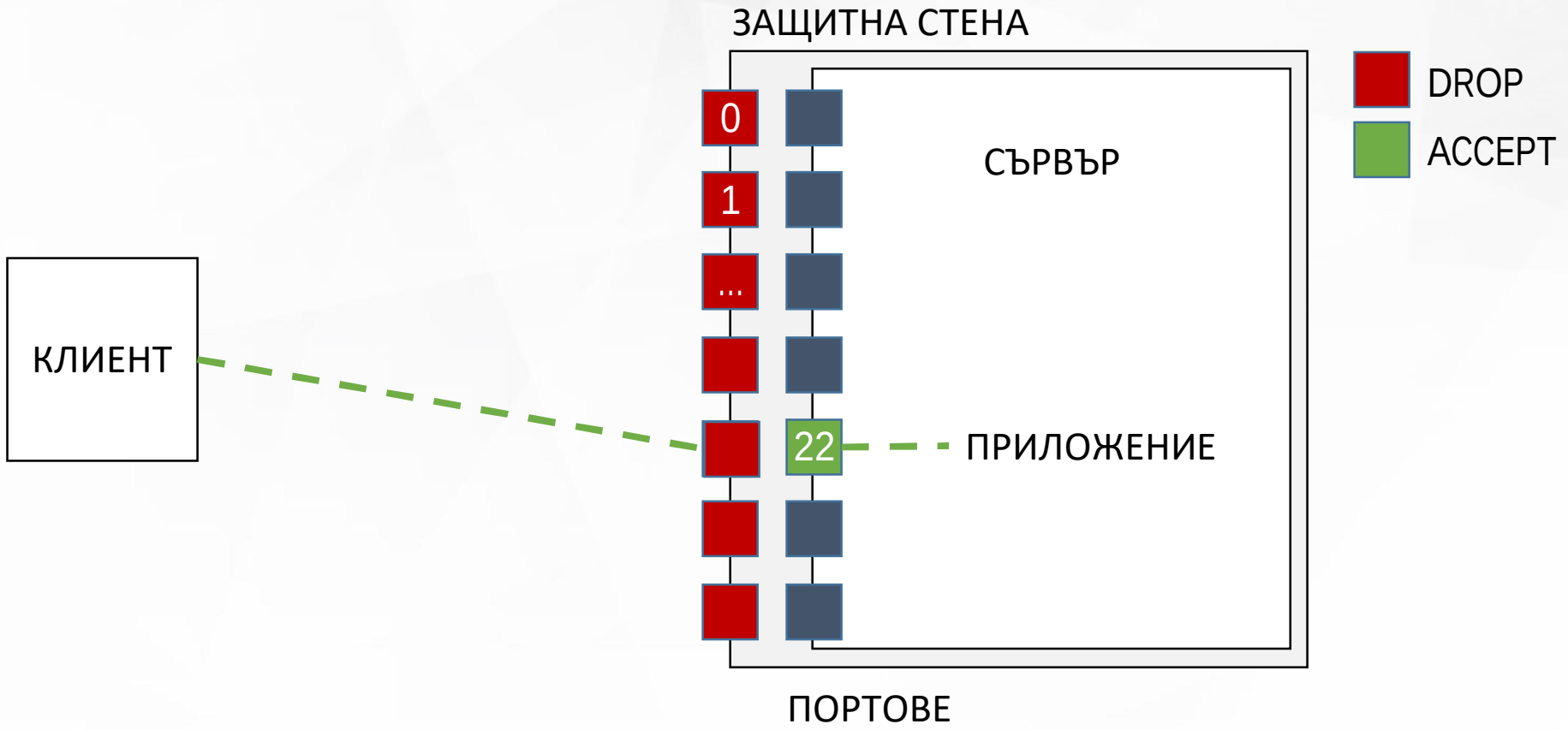
В лесни стъпки (1)



В лесни стъпки (2)

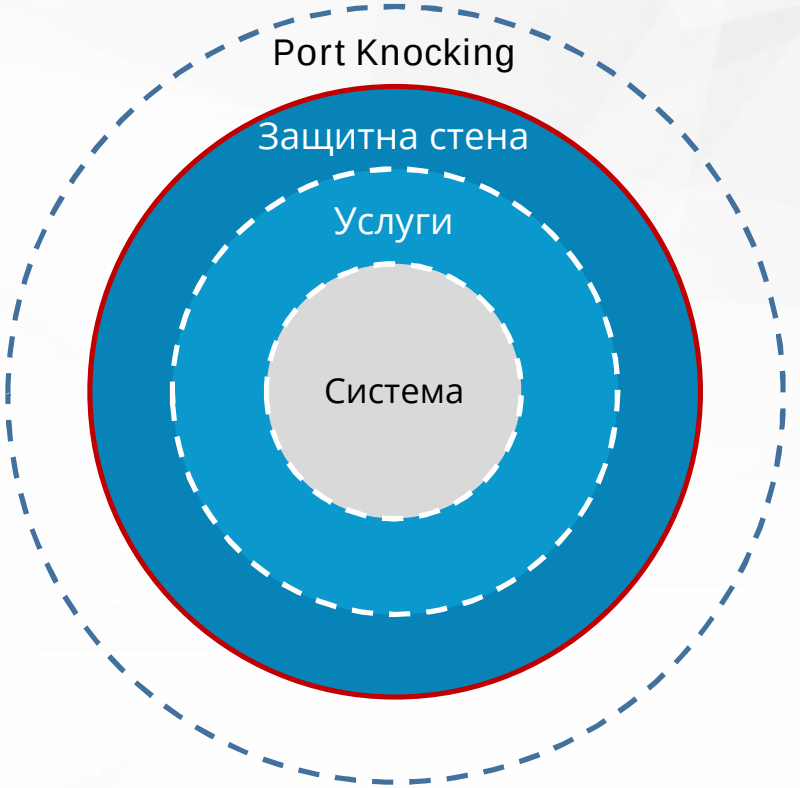


В лесни стъпки (3)



Допълнителен слой

- Важни елементи:
  - Защитна стена
  - Протоколи (TCP/UDP, ICMP и др.)
  - Мрежови портове
- Роли (клиент/сървър)
- Варианти:
  - Строга последователност от пакети
  - Предаване/разпознаване на данни



Предимства (+)

- Удостоверяване чрез защитната стена
- Портовете са затворени при сканиране
- Динамична защита (Stateful firewall)
- ~141 трилиона комбинации (3 правила)
- Гъвкава прозрачна защита
- Опростен дизайн, лесен за анализ и адаптация

Недостатъци (-)

- Почуквания не пристигат в същата последователност (Out-of-order delivery)
- Без криптиране не предотвратява защита от spoofing и man-in-the-middle атаки
- Неподходящо е да се прилага при публични услуги

RouterOS и Port Knocking

MikroTik RouterOS, като сървър и клиент

RouterOS и Port Knocking

- RouterOS може да играе ролята на **сървър**, като проверява последователността на опитите за връзка, както и данните, които могат да се пренасят (payload)
- Може да изпълнява и ролята на **клиент**, като прави предварително дефинирани опити за връзка и пренася информация с определени инструменти

Сървър (роля)

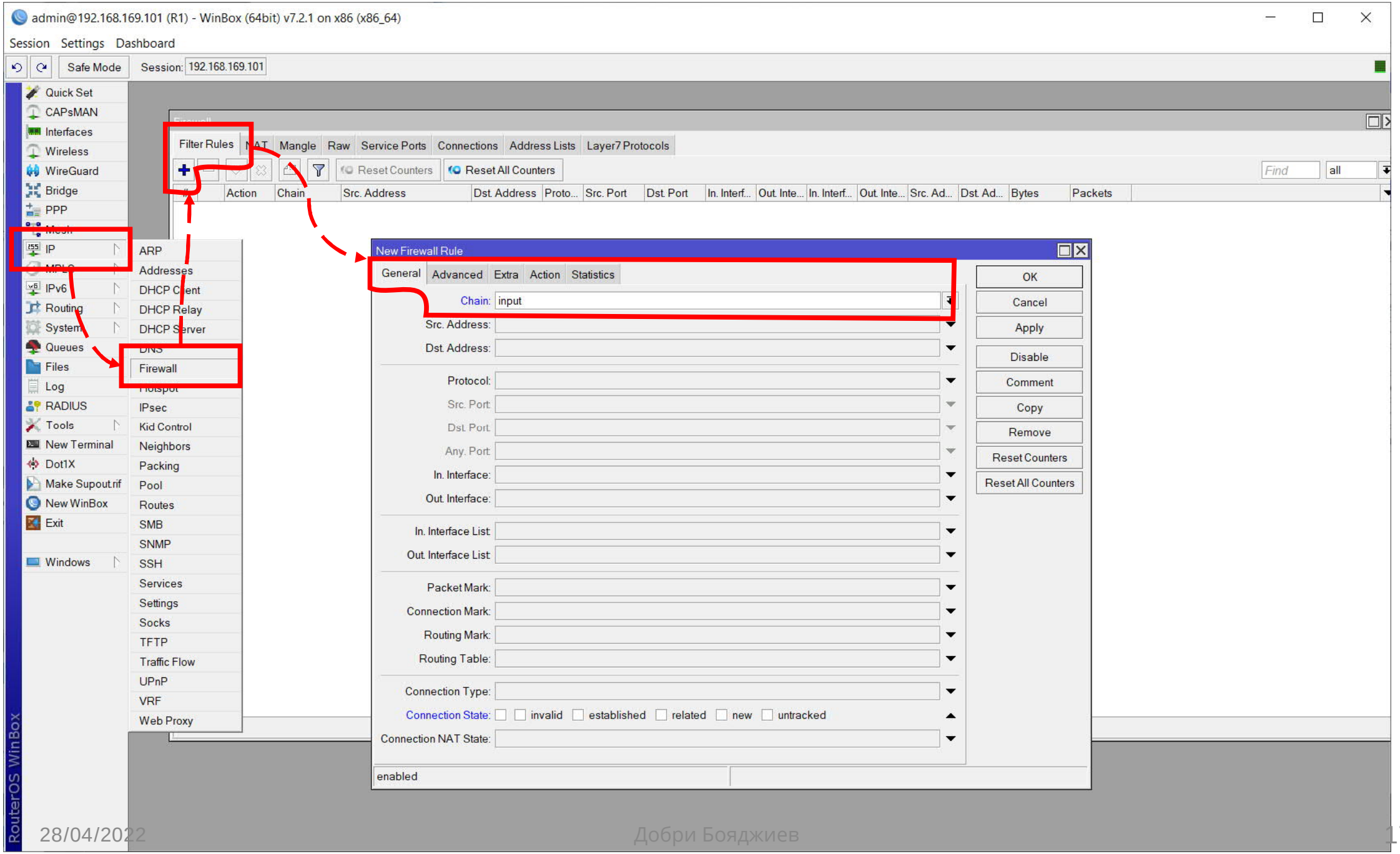
Последователност от връзки и/или пренасяне на информация

Сървър (в ролята на)

- Проследяване на строга последователност от опити за връзка към портове (TCP, UDP)
- Пренасяне и откриване на данни чрез различни протоколи
  - TCP, UDP, HTTP и др.
  - ICMP / Ping Knocking
- Могат да бъдат използвани в комбинация

Елементи в RouterOS

- /ip/firewall/filter:
  - **В условия**: c*hain*; protocol; c*onnection-state*; src-address-list; content; layer7-protocol; packet-size и др.
  - **В действие**: add-src-to-address-list, accept, drop
- /ip/firewall/raw
- /ip/firewall/connection
- /ip/firewall/address-list
- /ip/firewall/layer7-protocol
- /ip/firewall/nat


admin@192.168.169.101 (R1) - WinBox (64bit) v7.2.1 on x86 (x86_64)
Session Settings Dashboard
Safe Mode
Session: 192.168.169.101
Quick Set
CAPsMAN
Interfaces
Wireless
WireGuard
Bridge
PPP
IP
IPv6
Routing
System
Queues
Files
Log
RADIUS
Tools
New Terminal
Dot1X
Make Supout.rif
New WinBox
Exit
Windows
ARP
Addresses
DHCP Client
DHCP Relay
DHCP Server
DNS
Firewall
IPsec
Kid Control
Neighbors
Packing
Pool
Routes
SMB
SNMP
SSH
Services
Settings
Socks
TFTP
Traffic Flow
UPnP
VRF
Web Proxy
Filter Rules
NAT
Mangle
Raw
Service Ports
Connections
Address Lists
Layer7 Protocols
Reset Counters
Reset All Counters
Find
all
Action
Chain
Src. Address
Dst. Address
Proto...
Src. Port
Dst. Port
In. Interf...
Out. Inte...
In. Interf...
Out. Inte...
Src. Ad...
Dst. Ad...
Bytes
Packets
New Firewall Rule
General
Advanced
Extra
Action
Statistics
Chain: input
Src. Address:
Dst. Address:
Protocol:
Src. Port:
Dst. Port:
Any. Port:
In. Interface:
Out. Interface:
In. Interface List:
Out. Interface List:
Packet Mark:
Connection Mark:
Routing Mark:
Routing Table:
Connection Type:
Connection State:
invalid
established
related
new
untracked
Connection NAT State:
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Reset Counters
Reset All Counters
enabled
RouterOS WinBox

TCP/UDP port knocking

- Създаваме (3) правила при следните условия, за всяко:
  - Да проверява трафика, предназначен за рутера
  - Да следи определен порт (22)
  - При опит за връзка към него, записва временно IP адреса на подателя в определен (според логиката) списък
  - Всяко следващо правило включва и списъка от предишното
- Добавяме (1) правило, което да разрешава достъпа до порта (22), само за IP адресите, попаднали в последния списък
- Какво още е необходимо?

TCP/UDP Port Knocking

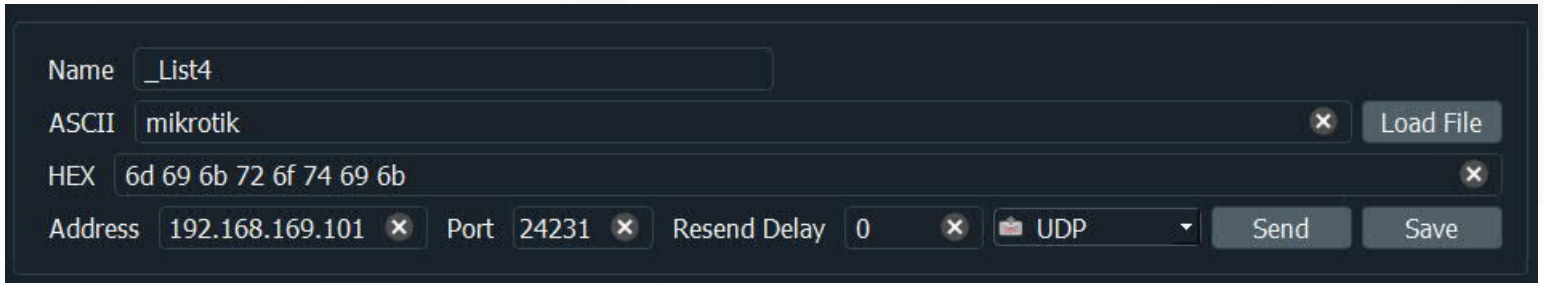
- Необходимо е да има още две правила:
  - Едното да приема вече установените и свързани връзки
  - Последното да изхвърля останалия трафик – да защитава устройството и неговите услуги

Демонстрация (1)

Последователност от връзки

Данни в TCP или UDP сегмента

- Можем да подаваме данни в TCP или UDP сегмента



- И да ги откриваме:

```
/ip firewall filter
add action=accept chain=input comment="Allow TCP port 22/SSH after PN" dst-port=22 \
    protocol=tcp src-address-list=List4 24231 mikrotik
```

Демонстрация (+)

Подаване/откриване на данни

Данни чрез HTTP GET заявка

- Възможно* е да се извършва и чрез HTTP GET заявка
  - http://192.168.169.1/mikrotik
- Проверката се прави чрез правило по същата логика, както на предишния слайд

* Изисква допълнителна логика

Сървър (роля)

Да разширим рамката с други протоколи и услуги

Ping (ICMP)

- (Linux / Windows) ping 192.168.169.101 –s (-l) 692
- /ip/firewall/filter:
  - Ще откриваме по: icmp; packet-size
- Да не забравим:
  - При ping с payload X=692 байта, пакетът е с размер X+28=720 байта
  - 20 байта за IP хедър + 8 байта за ICMP хедър

Ping (ICMP)

- Можем да пренасяме данни* в ICMP хедъра:
  - ping 192.168.169.1 -p 6d696b726f74696b
  - 6d696b726f74696b е думата mikrotik шестнадесетично (Hex)
- И да ги откриваме:
  - /ip firewall layer7-protocol \
    add name="ICMP pattern" regexp=mikrotik
  - /ip firewall filter add action=accept chain=input \
    layer7-protocol="ICMP pattern" protocol=icmp

* Изисква допълнителна логика

ICMP (~~Port~~ Ping Knocking)

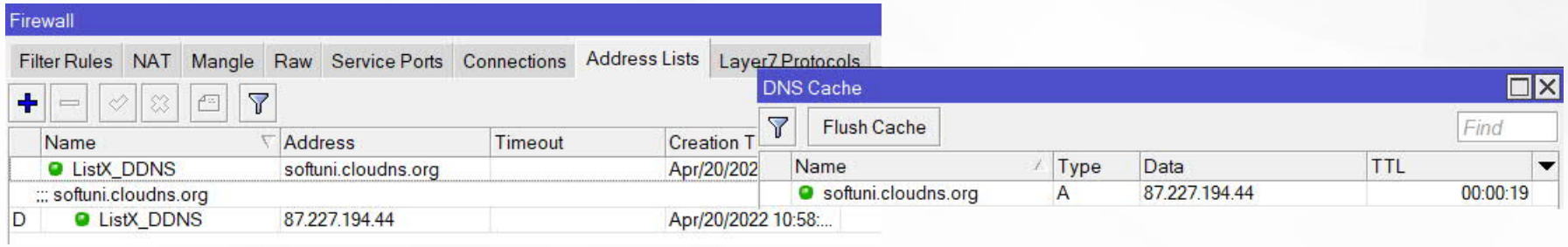
- Добавяме правило, което да:
  - Проверява трафика, предназначен за рутера
  - Следи само icmp
  - Търси в съдържанието думата *mikrotik*
  - Записва временно IP адреса на подателя в определен списък
- Да спазим логиката: всяко следващо правило включва и списъка от предишното

Демонстрация (+)

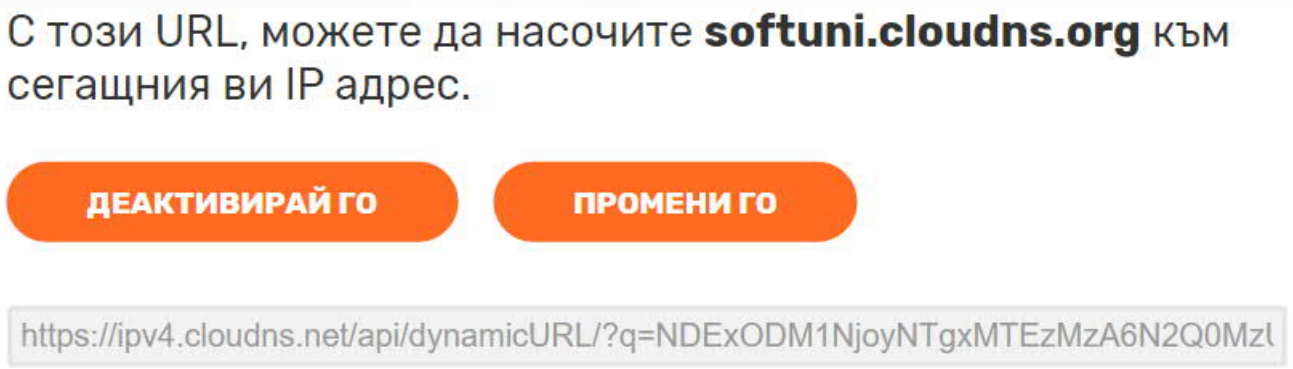
Пренасяне на данни

Разширяване (Address list knocking)

- Имена на хостове в защитната стена



- Услугата [динамичен DNS]



Клиент (роля)

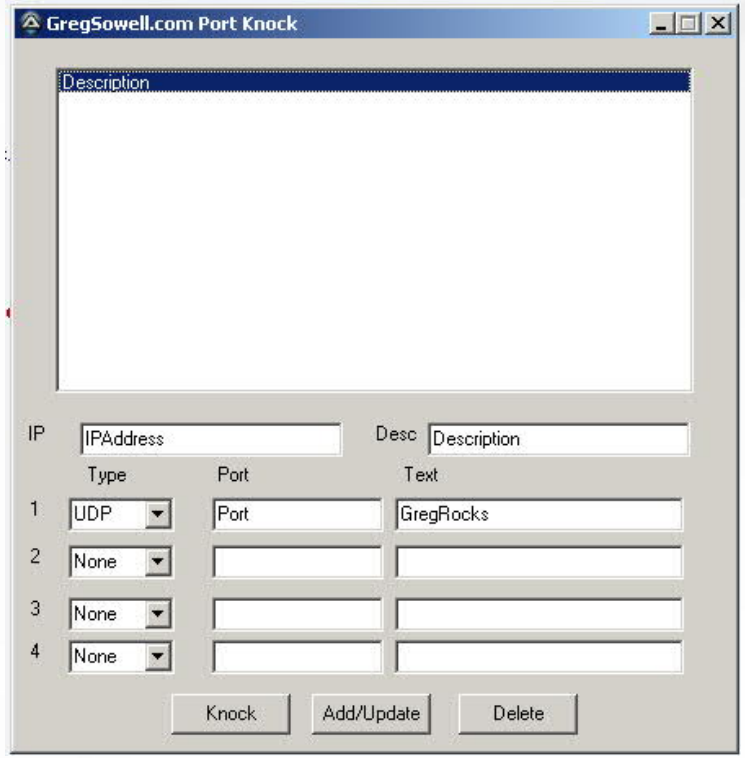
Последователност от връзки и/или пренасяне на информация

Клиент (CLI)

- **Тестване**: telnet, браузър, Winbox, curl, (cmd) ping
- Windows:
  - PortQry -n 192.168.88.1 -o 62521,9851,29819 -p both
  - knock 1.1.1.1 62521:tcp 9851:udp 29819:tcp
  - It's me (IM)
- Linux:
  - netcat -u 192.168.88.1 53
  - hping3 -s 192.168.88.1 -c 1 -p 1520
  - knock 192.168.88.1 2022:tcp 9090:udp 4022:tcp
  - socat - TCP:192.168.88.1:6756

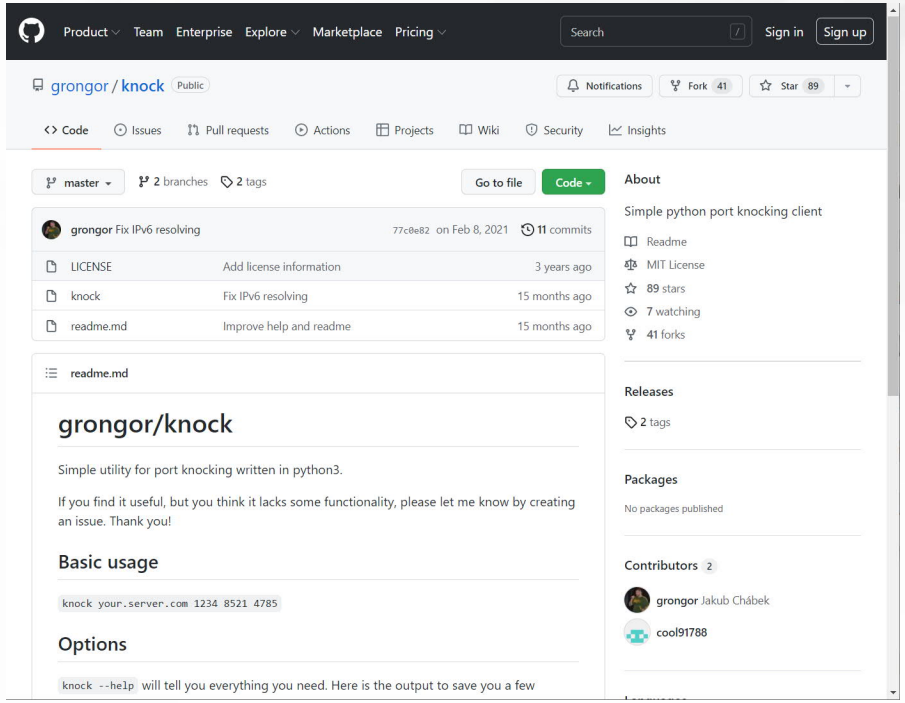
Клиент (GUI)

- Windows:
  - Packet Sender
  - Windows Port Knock Application
  - KnockKnock - Port Knocking for Windows
  - PortQryUI
- Android:
  - Port Knocker
  - Knock on Ports
  - Knock the Port



Клиент (скрипт)

- github.com/grongor/knock
  - knock your.server.com 1234 8521
  - usage: knock [-h] [-t TIMEOUT]
    [-d DELAY] [-u]
    host port[:protocol] [port[:protocol] ...]



Клиент MikroTik RouterOS

- /system/telnet
  - /system/telnet address=192.168.169.101 port=23
- /tools/fetch
  - /tool/fetch address=192.168.169.101 src-path=mikrotik mode=https
- /ping
  - ping 192.168. 169.101 size=1000
- /tools/traffic-generator
  - Каквито пакети са ви необходими 😂

Демонстрация (2)

MikroTik RouterOS, като port knocking клиент

Препращане на портове

- Port Knocking може да се използва и при достъп до препратени портове:
  - Контролът се извършва с помощта на адреси списъци
  - Може да се приложи подход за конфигурация в:
    - /ip/firewall/filter или
    - /ip/firewall/nat

Демонстрация (3)

Port Knocking и достъп до портове, конфигурирани с Port Forwarding

Добри практики

При изпълнение на Port Knocking

Добри практики (10)

1. **Приемете** връзки от тип established (и related)
2. Вашата **уникална комбинация** (TCP/UDP/ICMP, данни)
3. Използвайте custom chains (ICMP, L7)
4. Още по-сигурно (Address list knocking)
5. dstnat (NAT) или forward (Filter)
6. /ip/firewall/raw – с някои ограничения
7. Комбинирайте със **защита от сканиране на портове**
8. Автоматизирайте ☺
9. Не започвайте с ICMP (ping)
10. Не използвайте, като **единствен слой на защита**

БЛАГОДАРЯ ЗА ВНИМАНИЕТО

За връзка с мен | linkedin.com/in/dobri-boyadzhiev/

Въпроси и отговори

PORT KNOCKING FOR SECURITY

https://www.sli.do/ | Code: #PortKnocking