next step: perform the tests
usually tedious, error prone work
aided by a tool
easily repeatable
enter ft6
ft6 – Agenda

1. overview
2. info on design and implementation
3. live demo
4. writing your own tests (optionally)
ft6 – Design Goals

- easy to configure
- graphical user interface
- browse tests and results
- visual representation
ft6 – Design Goals

- open-source (Creative Commons BY-NC-SA 3.0)
- can act as a framework for new tests
- easy to implement new tests
ft6 – Details

- powered by python, PyQt and scapy
- works with Linux, Windows 7, OS X
- python: rapid development, easily understandable
- PyQt: GUI-framework, available cross-platform
  - [http://www.riverbankcomputing.com/software/pyqt/intro](http://www.riverbankcomputing.com/software/pyqt/intro)
- scapy: great framework for network packet creation
ft6 is a client-server application
- requires machines on both sides of your firewall
- one open port
- place machines not more than one hop away from firewall
ft6 – Running ft6

- Client and Server exchange control messages
  - Start / End / Results
ft6 – Running ft6

- Client sends packets
- Server sniffs
ft6 – Running ft6

- Client sends packets
- Server sniffs
ft6 – Running ft6

- Server sends back list of packets it received
- Client figures out what went missing and displays result
Handling Network Packets
ft6 – packet creation with scapy

- Handling network packets is usually messy
  - Binary protocols
  - Accessing individual flags involves bitshifting or bitmasking
- Sending and receiving is error-prone, too
- Scapy does all that for you and is human readable.
- Great TAB-completion
ft6 – scapy demo

```
>>> mypacket = IP
IP
IP6Field
IP6ListField
IP6PrefixField
IPField
IPID_count
IPOption
IPOption_Address_Extension
IPOption_EOL
IPOption_LSRR
IPOption_MTU_Probe
```

```
IPOption_MTU_Reply
IPOption_NOP
IPOption_RR
IPOption_Router_Alert
IPOption_SDBM
IPOption_SSRR
IPOption_Security
IPOption_Stream_Id
IPOption_Traceroute
IPPROTO_SCTP
IPPROTO_VRRP
```

```
IPTools
IPv6_ADDR_6TO4
IPv6_ADDR_CAST_MASK
IPv6_ADDR_GLOBAL
IPv6_ADDR_LINKLOCAL
IPv6_ADDR_LOOPBACK
IPv6_ADDR_MULTICAST
IPv6_ADDR_SCOPE_MASK
IPv6_ADDR_SITELOCAL
IPv6_ADDR_UNICAST
IPv6_ADDR_UNSPECIFIED
```

```
IP_PROTOS
IPerror
IPerror6
IPv6
IPv6ExtHdrDestOpt
IPv6ExtHdrFragment
IPv6ExtHdrHopByHop
IPv6ExtHdrRouting
IPv6inIP
```
ft6 – scapy demo

```python
>>> mypacket = IPv6()
>>> mypacket.show()
###[ IPv6 ]###
    version= 6
tc= 0
fl= 0
plen= None
nh= No Next Header
hlim= 64
src= ::1
dst= ::1
>>> 
```
ft6 – scapy demo

```python
>>> mypacket.src = "2001:db8::abcd"
>>> mypacket.show()
###[ IPv6 ]###
    version= 6
tc= 0
fl= 0
plen= None
nh= No Next Header
hlim= 64
src= 2001:db8::abcd
dst= ::1
```
ft6 – scapy demo

```python
>>> tcp = TCP(dport=80)
>>> payload = "GET index.html HTTP/1.1"
>>> result = mypacket/tcp/payload
```
ft6 – scapy demo

```
>>> result.show2()
### [ IPv6 ]###
  version= 6L
  tc= 0L
  fl= 0L
  plen= 43
  nh= TCP
  hlim= 64
  src= 2001:db8::abcd
  dst= ::1
### [ TCP ]###
  sport= ftp_data
dport= www
  seq= 0
  ack= 0
  dataofs= 5L
  reserved= 0L
  flags= S
  window= 8192
  chksum= 0xd79d
  urgptr= 0
  options= []
### [ Raw ]###
  load= 'GET index.html HTTP/1.1'
```
Live Demo
Example: build own test, to see if packets containing the string "randomword" can traverse the firewall. Requires four steps:

1. create a class for your test
2. implement the `execute` method
3. implement the `evaluate` method
4. register your test with the application

(More detailed in ft6’s documentation)
Step 1: Create a class for your test

```python
class TestRandomWord(Test):
    def __init__(self, id, name, description, test_settings, app):
        super(TestRandomWord, self).__init__(id, name, description,
                                             test_settings, app)
```
def execute(self):
    e = Ether(dst=self.test_settings.router_mac)
    ip = IPv6(dst=self.test_settings.dst, src=self.test_settings.src)
    udp = UDP(dport=self.test_settings.open_port, sport=12345)
    payload = "ipv6-qab"*128

    packet = e/ip/udp/(payload + "randomword")
    sendp(packet)

    packet = e/ip/udp(paylload + "someotherword")
    sendp(packet)
ft6 – Writing your own tests

Step 3: implement the \texttt{evaluate} method

```python
def evaluate(self, packets):
    results = []
    found_random = False
    found_otherword = False

    # iterate over the packets, filter those that belong to the test
    for p in packets:
        tag = str(p.lastlayer())
        if not "ipv6-qab" in tag:
            continue

        if "randomword" in tag:
            found_random = True

        if "someotherword" in tag:
            found_otherword = True
```

Oliver Eggert (Universität Potsdam)
Step 3: implement the `evaluate` method

```python
# evaluate the flags
if found_random:
    results.append("Success", "Your firewall forwarded a packet with a random word!")
else:
    results.append("Failure", "Your firewall dropped a packet with a random word!")

if found_otherword:
    results.append("Warning", "Your firewall forwarded a packet with some other word. That’s very weird!")
else:
    results.append("Success", "Your firewall dropped a packet with some other word. Well done firewall!")

return results
```
ft6 – Writing your own tests

Step 4: register your test

# create test classes, store them in the dictionary
# so they can later be called by their id

```
tICMP = TestICMP(1, "ICMPv6 Filtering", "The ICMP Test",
    self.test_settings, app)
...

tRandomWord = TestRandomWord(42, "My Random Word Test",
    "Tests for Random Words", self.test_settings, app)

```

```
self.tests = dict([  
    (tICMP.id, tICMP), ..., (tRandomWord.id, tRandomWord)])
```

ft6 – future work

- ft6 is a work in progress
- lots of improvement could be done
- reducing the number of text fields
- better results
- more tests

**your thoughts:** contact@idsv6.de

- ft6 is available at
  