Packet Inspection Using Python

Why?

- Access to low-level kernel packet processing via a high-level API.
- Excellent for quick experiments & prototypes.
- Can implement features not easy to implement with existing tooling:
  - Log specific packets or connection attempts for further analysis.
  - Identify overly chatty programs.
  - Develop an outbound firewall to help protect against trojans.

Primary Building Blocks

iptables

Direct packets to a netfilter queue

nfqueue

Hook your program into a netfilter queue.

scapy

Intuitive packet inspection

iptables

Need to send packets to the netfilter queue using iptables. Start with simple rules to minimize negative impacts to your network.

- Only outbound packets to a specific address:

```
$ iptables -A OUTPUT --dst www.orvant.com -j NFQUEUE --queue-bypass
```

- Only initial packets of outbound connections. Otherwise, will process & log all the packets for a connection:

```
$ iptables -A OUTPUT -m state --state NEW -j NFQUEUE --queue-bypass
```

- Send packets to specific queues, allowing for multiple processes:

```
$ iptables -A INPUT -j NFQUEUE --queue-bypass --queue-num 0
$ iptables -A OUTPUT -j NFQUEUE --queue-bypass --queue-num 1
```

nfqueue

Python bindings to *libnetfilter_queue*. Example "Hello, world!":

```
import socket, nfqueue, atexit

def process(packet):                    # callback method
    print("Hello, packet!")
    print(packet.get_data())
    packet.set_verdict(nfqueue.NF_ACCEPT)

def shutdown(q):                        # cleanup method
    q.unbind(socket.AF_INET)
    q.close()
```

```
q = nfqueue.queue()
q.set_callback(process)                 # register callback
```

```
q.fast_open(0, socket.AF_INET)        # bind to queue 0
atexit.register(shutdown, q)          # make sure to clean up
q.try_run()                           # start processing the queue
```

scapy

The Scapy library is a good option for inspecting packets (see dpkt or impacket for alternatives).

```python
from scapy.all import IP, TCP, UDP, Raw

# Easy to add new protocols to scapy [1]
from HTTP import HTTPRequest

def log_http_domains(packet):          # our nfqueue callback
    packet.set_verdict(nfqueue.NF_ACCEPT)
    ip = IP(packet.get_data())
    if HTTPRequest in ip:              # only HTTP request packets
        request = pkt[HTTPRequest]  # how to access layers
        print("HOST:", request.Host)
```

orvant-snitch

Inspired by Little Snitch for OSX. Is a small, headless version for Linux machines. Useful to answer questions like: "What traffic is program X causing?"

```
$ iptables -A OUTPUT -m state --state NEW \
    -j NFQUEUE --queue-bypass --queue-num 1
$ cat /etc/ov-snitch.conf
  rules:
    /usr/bin/curl:
      443: {deny: true}
  queue: 1
$ ov-snitch
$ tail /var/log/syslog | \
    awk -F '[ :|]+' '$0 ~ "ov-snitch" {
      printf("%-10s %15s:%-5s %s\n",
             $(NF-8), $(NF-3), $(NF-2), $(NF-1));
    }' &
$ curl --max-time 1 -I 'http://www.orvant.com' > /dev/null
  allowed    69.160.46.73:80    /usr/bin/curl
$ curl --max-time 1 -I 'https://www.orvant.com' > /dev/null
  denied     69.160.46.73:443   /usr/bin/curl
```

Some Caveats

- Make sure to use *--queue-bypass* in iptables rules. Otherwise, packets will hang if there is no active program processing the queue.
- These libraries are getting stale.
- Working at the packet level. Easy to mess up the connection if you want to mangle a packet (e.g. alter the payload, IP address, or port). Have to worry about checksums, sequence numbers, etc.

Some References

- NFQueue Bindings - https://www.wzdftpd.net/redmine/projects/nfqueue-bindings
- Scapy - http://www.secdev.org/projects/scapy/
- **NetFilter - http://www.netfilter.org/**
  - https://home.regit.org/netfilter-en/using-nfqueue-and-libnetfi
- **Alternatives to nfqueue-bindings:**
  - http://code.google.com/p/python-libnetfilter-queue/ (uses ctypes)
  - https://github.com/kti/python-netfilterqueue (uses cython, in pypi)

Thank You!

- **Erik Stephens**
    - erik@orvant.com
- **code for orvant-snitch:**
    - https://github.com/orvant/orvant-snitch
- **Network Vulnerability Assessments:**
    - https://www.orvant.com

1	scapy-http https://github.com/invernizzi/scapy-http	-