Encrypted Tunnel Through Virtual Network Interface

Safiqul Islam
INF -9090 – Project Presentation
University of Oslo
Outline

- Introduction
- Background
  - Virtual Private Network
  - Virtual Network Interface
  - Link Local Addressing
  - Cryptography
    - Asymmetric Key Cryptography
    - Symmetric Key Cryptography
- Design
- Evaluation
- Conclusion and Future Work
Introduction

- Virtual Private Network (VPN) provides secure communication over the insecure public network.

- Most of the current open source methods do not support *Mobility* - such as: Vtun and OpenVPN

- Some proprietary methods: Cisco VPN, and Netmotion support mobility

- Designing a system that uses a virtual network interface and supports mobility is the primary goal of this system.
Virtual Private Network

- Provides secure communication over the insecure public network via
  - Authentication
  - Encryption
  - Compression
  - Tunneling

- IPSec
  - Tunnel Mode
  - Transport Mode
Virtual Network Interface

- An Ethernet like device
  - Receives packets from the userspace program
  - Sends them to the userspace program before sending it via physical media.

- TUN/TAP driver is used to create Virtual Network Interface
  - TUN is used for reading and writing IP packets
  - TAP is used for reading and writing Ethernet frames

- By using TUN/TAP for making connection with the other end, we can add the support of mobility when the connection is moved to different location.
Cryptography

- An art of science for transforming intelligible text to an unintelligible one and vice versa.
  - Intelligible text is plain text
  - Unintelligible text is cipher text

- Public-key cryptography
  - Have a pair of cryptographic keys
    - Public and private – mathematically linked
Public-key Cryptography

- Public key is publicly known, and private key has to be kept secret.

- Encryption is done using the public key of the user, and decryption is done using the private key.

- Digital signature is also performed using this cryptography.

plaintext $m$→ Encryption Algorithm→ Ciphertext $Key_R^+(m)$→ Decryption Algorithm→ plaintext $m = Key_R^-(Key_R^+(m))$
Link Local Address

- Intended for addressing on a single link or for a Local Area Network
- Routers do not forward such packets
- Both IPV4 and IPV6 have reserved a block for link local addresses.
  - 169.254.0.0/16 for IPV4
  - Fe80::/64 for IPV6
Design

- Provides Server/Client functionality
- Uses TUN for virtual network interface
Design

- IPv4 link local addresses are used for configuring the TUN interfaces.
- To successfully traverse the network packet is encapsulated into an UDP packet.
Design

- Encryption
- Integrity checking
- Mobility

IP  UDP  Signature  VPN  Payload

Signed and Encrypted
Challenges

- Transport Protocols
  - UDP – TCP over TCP problems
  - Simpler methods and higher success rates

- Kernel Space vs User Space
  - Portability
  - Efficiency
Evaluation

- Metrics
  - Throughput
  - Latency

- Mobility Test
## Testbed 1

<table>
<thead>
<tr>
<th>Device</th>
<th>Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host 1</td>
<td>Processor: Intel Pentium 4, 1.60GHz</td>
</tr>
<tr>
<td></td>
<td>RAM : 0.5 GB</td>
</tr>
<tr>
<td></td>
<td>OS : lubuntu</td>
</tr>
<tr>
<td></td>
<td>Network interface: Realtek RTL-8169 Gigabit Ethernet</td>
</tr>
<tr>
<td>Host 2</td>
<td>Processor: Intel Pentium 4, 1.60GHz</td>
</tr>
<tr>
<td></td>
<td>RAM : 0.5 GB</td>
</tr>
<tr>
<td></td>
<td>OS : lubuntu</td>
</tr>
<tr>
<td></td>
<td>Network interface: Realtek RTL-8169 Gigabit Ethernet</td>
</tr>
<tr>
<td>Switch</td>
<td>8 port Gigabit desktop switch</td>
</tr>
<tr>
<td></td>
<td>Vendor: DLINK</td>
</tr>
<tr>
<td></td>
<td>Model : DGS -1008D</td>
</tr>
</tbody>
</table>
## Testbed 2

<table>
<thead>
<tr>
<th>Device</th>
<th>Configurations</th>
</tr>
</thead>
</table>
| Host 1 | Processor: Intel core i7, 2.93 GHz  
RAM : 8 GB  
OS : Fedora 18  
Network interface: Intel 82578DM Gigabit Network Connection |
| Host 2 | Processor: Intel core i7, 2.93 GHz  
RAM : 8 GB  
OS : Fedora 18  
Network interface: Intel 82578DM Gigabit Network Connection |
| Switch | 8 port Gigabit desktop switch  
Vendor: DLINK  
Model : DGS -1008D |
File Transfers over SSH

Table: File Transfers over SSH for testbed 1

<table>
<thead>
<tr>
<th>File Size (KiB)</th>
<th>Throughput (KBps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>1.1</td>
</tr>
<tr>
<td>64</td>
<td>1.5</td>
</tr>
<tr>
<td>128</td>
<td>1.4</td>
</tr>
</tbody>
</table>

Table: File Transfers over SSH for testbed 2

<table>
<thead>
<tr>
<th>File Size (KiB)</th>
<th>Throughput (KBps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>64</td>
<td>21.3</td>
</tr>
<tr>
<td>1024</td>
<td>18.6</td>
</tr>
<tr>
<td>10240</td>
<td>17.7</td>
</tr>
</tbody>
</table>
Latency

<table>
<thead>
<tr>
<th></th>
<th>Total Packets</th>
<th>Minimum</th>
<th>Average</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>100</td>
<td>0.126</td>
<td>0.270</td>
<td>0.341</td>
</tr>
<tr>
<td>Our System</td>
<td>100</td>
<td>58.141</td>
<td>66.908</td>
<td>82.157</td>
</tr>
</tbody>
</table>
Throughput

TCP Throughput with VPN using iperf

TCP Throught without VPN using iperf
Mobility

TCP Throughput over VPN - Mobility Test

Throughput (Kbits/s) vs Time(s)
Conclusion

- Implemented and evaluated an encrypted tunnel where we used virtual network interface.
- Supports mobility
- However, regular system outperforms our system
- There are some future works:
  - Symmetric key cryptography.
  - CPU performance.
  - IP address derivation from the public key
Acknowledgement

- We would like to thank Hans for helpful discussion and valuable feedback.
Thanks and Questions ? 😊