Protecting Insecure Communications with Topology-aware Network Tunnels

Georgios Kontaxis

Angelos D. Keromytis

Department of Computer Science
Columbia University, USA
Clients securing their traffic to a server

- TLS unavailable at the server
- Minimize the unencrypted network path on the Internet
- Without the server’s participation
- Not a substitute for TLS!
Limited adoption of transport layer security

- Top 10K sites (Alexa)
- Only 32% support HTTPS
- Only 15% redirect HTTP to HTTPS

<table>
<thead>
<tr>
<th>HTTPS response</th>
<th>HTTPS?</th>
<th>%</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Error (Conn. refused)</td>
<td>No</td>
<td>21.4</td>
<td>2144</td>
</tr>
<tr>
<td>2 Error (Invalid cert.)</td>
<td>No</td>
<td>22.1</td>
<td>2205</td>
</tr>
<tr>
<td>3 Error (HTTP 4xx 5xx)</td>
<td>No</td>
<td>2.9</td>
<td>292</td>
</tr>
<tr>
<td>4 HTTPS downgraded</td>
<td>No</td>
<td>21.5</td>
<td>2152</td>
</tr>
<tr>
<td>Total</td>
<td>No</td>
<td>67.9</td>
<td>6793</td>
</tr>
<tr>
<td>5 OK</td>
<td>Yes</td>
<td>17.0</td>
<td>1695</td>
</tr>
<tr>
<td>6 OK (HTTP upgraded)</td>
<td>Yes</td>
<td>15.1</td>
<td>1512</td>
</tr>
<tr>
<td>Total</td>
<td>Yes</td>
<td>32.1</td>
<td>3207</td>
</tr>
</tbody>
</table>
Imperfect deployment of TLS

- Implementation vulnerabilities threaten user security
  
  FREAK  POODLE  Heartbleed  RC4
  BEAST  DROWN

- Users cannot rely on websites to patch themselves up
  - 45% of servers affected by FREAK vulnerable 9 months later
  - DROWN affects a TLS 1.2 client because server supports SSLv2
Short network paths minimize the attack surface

- kontaxis@austria$ traceroute www.nytimes.com
  1. EDIS GmbH (AT)
  2. RETN Limited (UK)
  3. NTT America, Inc. (US)
  4. Fastly (US)

- kontaxis@ec2-us-east-1$ traceroute www.nytimes.com
  1. Amazon Inc. (US)
  2. Fastly (US)
Web services are clustered in the cloud

- Cloud networks host the majority of web services
- Excellent vantage point to browse the web
- Users have access to Virtual Machines in the cloud

<table>
<thead>
<tr>
<th>%</th>
<th>Autonomous System Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.1</td>
<td>Akamai Technologies, Inc.</td>
</tr>
<tr>
<td>13.9</td>
<td>Amazon.com, Inc.</td>
</tr>
<tr>
<td>11.4</td>
<td>CloudFlare, Inc.</td>
</tr>
<tr>
<td>9.9</td>
<td>Google Inc.</td>
</tr>
<tr>
<td>3.7</td>
<td>EdgeCast Networks, Inc.</td>
</tr>
<tr>
<td>2.9</td>
<td>SoftLayer Technologies Inc.</td>
</tr>
<tr>
<td>2.1</td>
<td>Fastly</td>
</tr>
<tr>
<td>1.7</td>
<td>Tinet SpA</td>
</tr>
<tr>
<td>1.6</td>
<td>Internap Network Services Corp.</td>
</tr>
<tr>
<td>1.5</td>
<td>Rackspace Hosting</td>
</tr>
<tr>
<td>65.8</td>
<td>Total</td>
</tr>
</tbody>
</table>
Cloud networks are the gateway to Internet services
Proposed overlay of encrypted tunnels with the cloud

- We replace multi-hop plain-text links with encrypted tunnels
Topology-aware Network Tunnels (TNT)

• Routing through the tunnel closest to the server
TNT establishes links to popular cloud networks
Given a destination TNT maps the available paths

• Initially traffic is routed through a tunnel at random

<table>
<thead>
<tr>
<th>Dest</th>
<th>GW</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>Any</td>
<td>0</td>
</tr>
</tbody>
</table>

Client \[\rightarrow\] Topology-aware Tunnel 1 \[\rightarrow\] Cloud Provider 1

Client \[\rightarrow\] Topology-aware Tunnel 2 \[\rightarrow\] Cloud Provider 2

Cloud Provider 1 \[\rightarrow\] Server A

AS z
TNT evaluates the available paths to a destination.

- Active, passive network measurements identify the shortest path.
TNT makes routing decisions to minimize plain-text traffic

- TNT starts routing packets through the shortest path in real time

<table>
<thead>
<tr>
<th>Dest</th>
<th>GW</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>Any</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>Tun 1</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>Tun 2</td>
<td>2</td>
</tr>
</tbody>
</table>

Topologies:
- **Topology-aware Tunnel 1**: Dest = A, GW = Tun 1, Metric = 1
- **Topology-aware Tunnel 2**: Dest = A, GW = Tun 2, Metric = 2
Insecure network paths are minimized

- Tunnel exit in the same network as a web server
  - Zero traffic exposure to the Internet
- Tunnels to AWS, Azure are collocated with 20% of web services
Insecure network paths are minimized

- Tunnel exit in a network near the web server
  - Minimal traffic exchange outside the cloud
- TNT paths are always shorter than the native path

![CDF of the number of ASes on the network path to a web service]

Autonomous systems making up the network path to a web service

kontaxis@cs.columbia.edu
TNT preserves the browsing experience

- Page load time and latency do not deviate from the baseline
- Used the network at Columbia University for comparison
Topology-aware network tunnels

- An overlay of encrypted links to key network infrastructure
- Motivated by the clustering of services in the cloud
- Minimize plain-text traffic on the Internet
- Without the cooperation of individual services
- Put clients in control of their security
- Deployable using existing technologies and resources
Figure 1: In the TNT architecture an overlay of secure topology-aware tunnels is established between the client and a set of network vantage points. The number and placement of secure tunnels is strategically selected to minimize the network distance packets need to travel outside the overlay to reach their destination. Individual network packets are intelligently routed through the tunnel exiting closest to their destination. Tunnel exits within the same network as the destination of a packet (Servers A, B) eliminate the exposure of traffic to network adversaries.

Figure 2: Example of a network path on the Internet. For insecure protocols such as HTTP data are exposed across the path to operators of the underlying infrastructure. With these infrastructure providers. This addresses the first challenge from above. That way we can shorten the insecure network path and essentially bring the client as close to these servers as possible, ideally within the same network. As a result, the traffic of insecure protocols will have minimal or zero exposure on the Internet. Apart from minimizing the overall path length, we also define metrics rewarding or penalizing the presence of a trusted or untrusted intermediate network in the path. The trustworthiness of a network is context specific so in this paper we focus on path length. Figure 2 presents an example of a network path today. The set of links and routers a client's packets must traverse to reach a server is grouped into autonomous systems (ASes) and controlled by distinct organizations. Note that such path might span different countries or continents. This translates to potential passive and active attacks against the user's web browsing or e-mail.

Figure 1 presents the TNT architecture as an overlay on the existing Internet infrastructure. TNT has established secure tunnels between the client and two cloud networks that exhibit high clustering of Internet services. Server A is hosted by Cloud Provider 1 and we can reach it through Topology-aware Tunnel 1 without exposing plain-text traffic to the Internet. Packets towards Server A enter the tunnel before leaving the user's network and are encrypted and signed. Internet routers operated by AS 1 and AS x observe an encrypted flow from the user to Server A. Without TNT these ASes have access to plain-text traffic. Packets exit the tunnel inside the trusted network of Cloud Provider 1 and are authenticated and decrypted. Subsequently packets transit the cloud provider's network and reach Server A which is unaware of the process. To reach Server C without TNT the user's packets will travel in plain text through AS 1, AS y and AS z. With a TNT link to Cloud Provider 2 they travel encrypted and signed through AS 1 and AS y. Server C is an outlier not hosted in a cluster of Internet services. In this case TNT is able to minimize the length of the insecure network path so instead of 3 ASes only AS z will be able to observe plain-text traffic. Next we describe how the TNT router determines the optimal tunnel to route traffic through so as to minimize insecure network paths.

5.2 The TNT Router

The TNT router is a routing software suite managing topology-aware tunnels and directing traffic through them. It is located on the client's system or local network gateway, for instance a home router, and maintains topology-aware tunnels with remote networks based on the placement strategy described earlier. It has a network-mapping and a decision-making component. Given the available tunnels and a specific destination address the mapping component employs a set of probes to discover the network path between each tunnel's exit on the remote network and the destination. The discovery process involves active and passive network measurements described in section 7. The information is passed on to the decision-making component which evaluates it and assigns metrics on each tunnel based on its suitability to carry traffic to the specific destination. Based on the metric the TNT router directs outgoing traffic through the tunnel which minimizes its value. This satisfies...