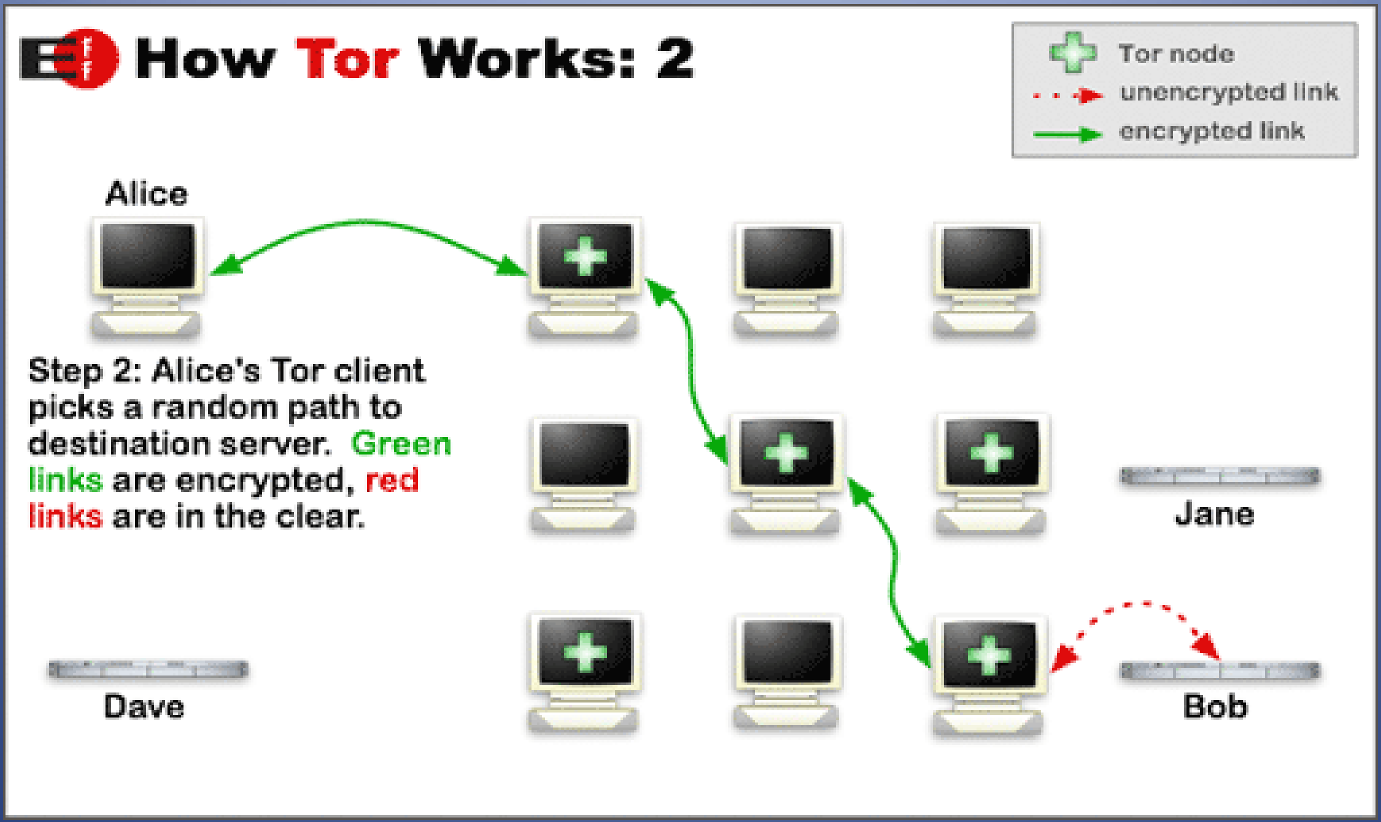
Low Cost Traffic Analysis of Tor

Steven Murdoch and George Danzis

A quick overview of Tor

- It’s an implementation of Onion Routing
- Low-latency
- Attempts to balance between performance and anonymity and it must be used in the real world
- Easy to deploy

How it works



Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.

Image from the Tor project site http://www.torproject.org/overview.html.en

Threat Model

- Limits the scope of the threat model
  - No global adversary present
  - Does not try to conceal who connects to the Tor networks
  - Traffic Analysis as opposed to traffic confirmation (end-to-end attacks)

Attack within the Threat Model

- Limited Resources to be controlled
- Partial view of the network
- A corrupted node
- A corrupted network server
- Probing attacks

The Attack

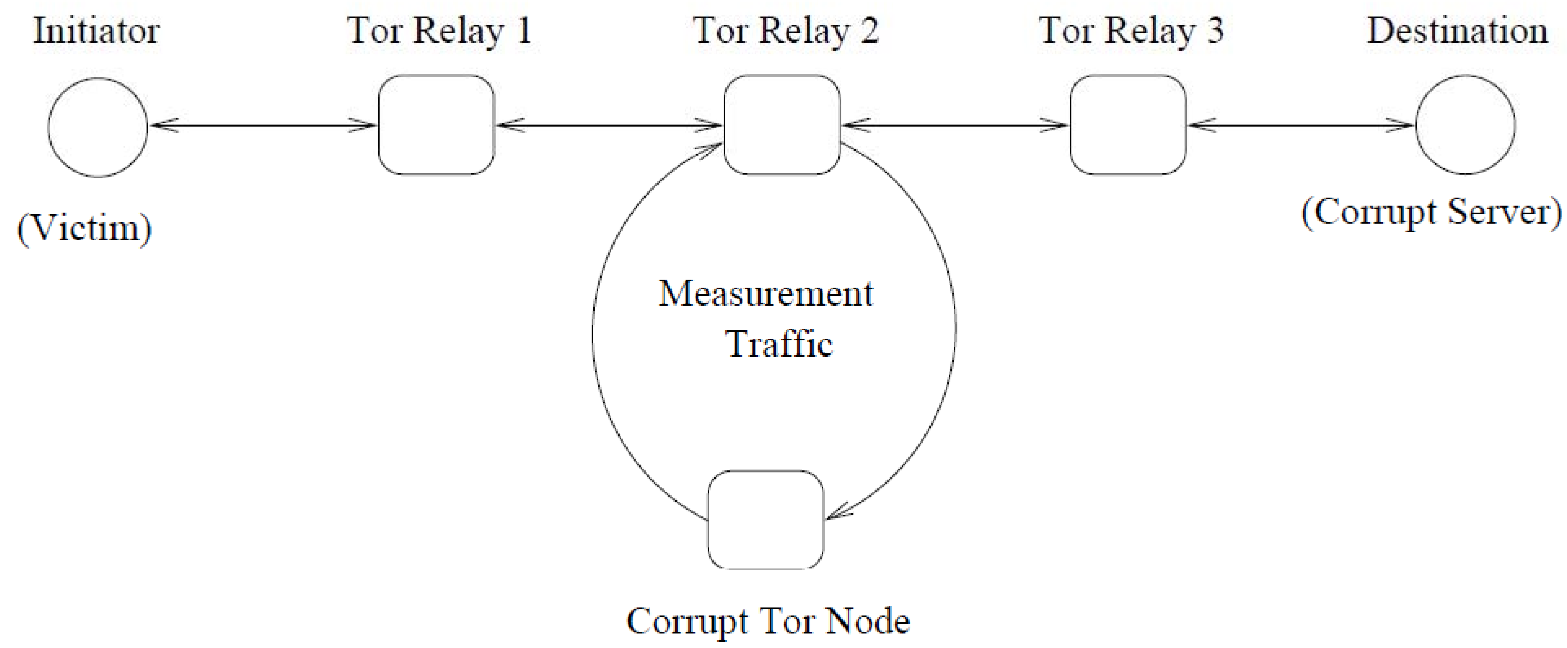


Figure 1. The attack setup

The Attack

- Get the traffic load on the Tor node (load on one node affects the latency of all connections)
- Consider a Tor node corrupt (aim is to measure traffic load of another node)
- Fill the connection with probe Traffic
- Adversary controls a network server connected that the victim is connected
- - bursts of data are sent to the victim via Tor from the network server

Resources Used

- 800 MHZ PC running Debian
- Modified Tor 0.0.9 to select a route of length 1 rather than 3
- Attempting to remove timing properties of runtime services in the code //the corrupted Tor node
- Onion Proxy on the victim was not modified
- Simulated TCP server as network server
- Simulated TCP client to receive data

Experiment

- Probe Client would send data every 0.2 seconds containing the time in ms
- Exit nodes were probed //but this is applicable to all nodes
- Network server to send data between 10 and 25 seconds then stop sending between 30 and 75
- Nodes were targeted in turn //possible with a corrupt node

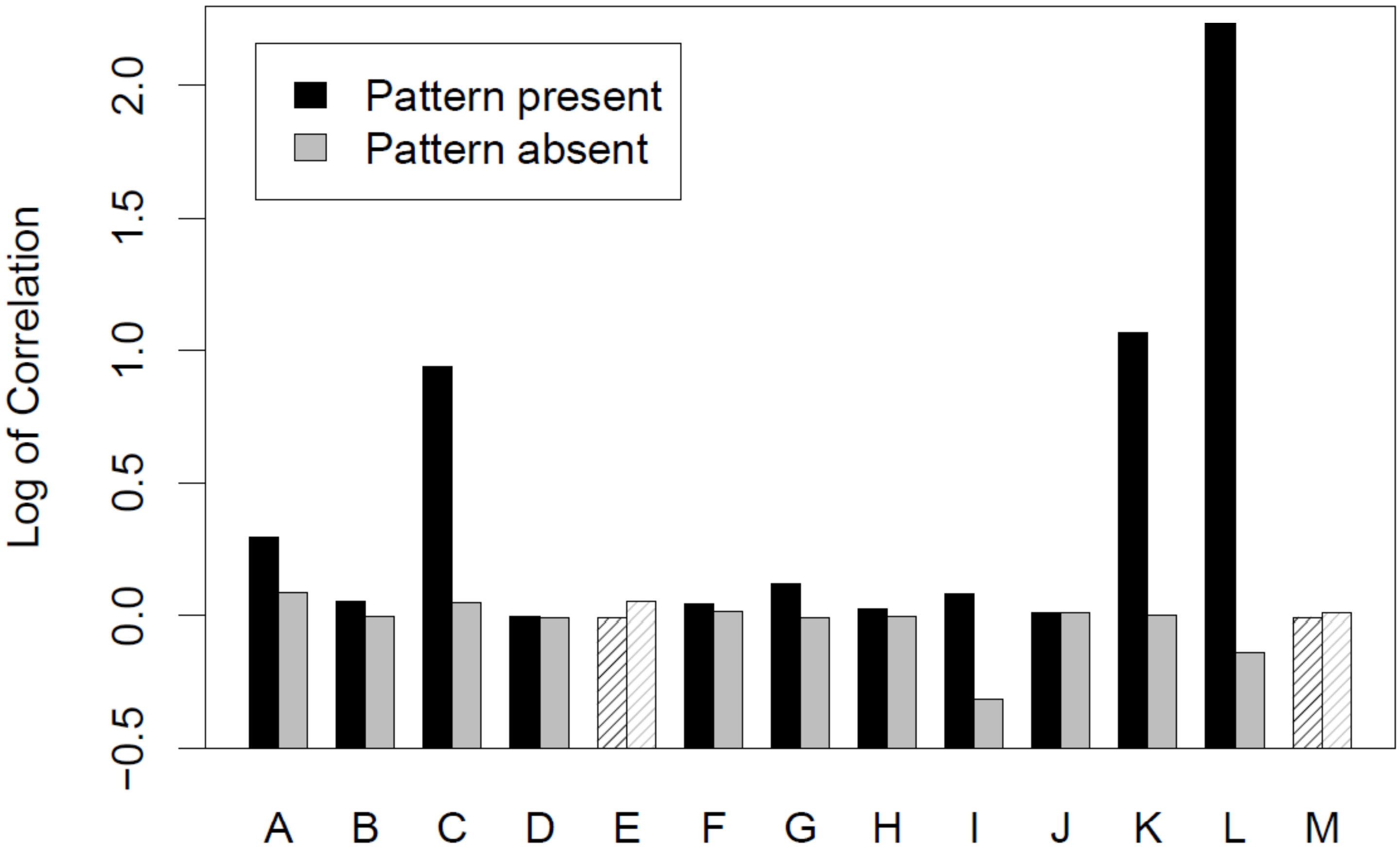
The Cycle

- Probe server would monitor a target node
- Create victim stream to monitor the furthest node away
- Monitor for a while after the stream is closed //in order to prevent false positives
- All data was stored in a file for analysis

Results

- A variation was observed where the target nodes were indeed carrying varying traffic
- Distortion of patterns
- 2 were not correctly identified

Results


Pattern present
Pattern absent
Log of Correlation
−0.5
0.0
0.5
1.0
1.5
2.0
A
B
C
D
E
F
G
H
I
J
K
L
M

Conclusions

- Inexpensive attack
- Adversary did not have full knowledge of the Tor network
- The network itself was used to probe traffic (the corrupted tor node)
- Tor using the same path for multiple stream leaks information

Discussion/Questions

- How Costly is it
- Increase the latency?
- A security discipline?