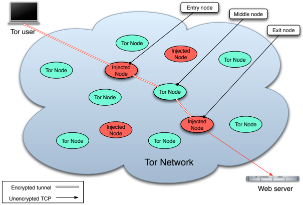
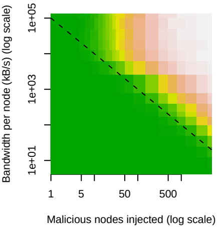
Metrics for Security and Performance in Low-Latency Anonymity Systems





Steven J. Murdoch, Robert N. M. Watson

http://www.cl.cam.ac.uk/users/{sjm217,rnw24}/

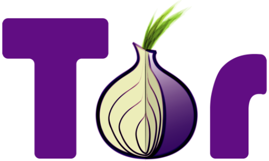
UNIVERSITY OF CAMBRIDGE
Computer Laboratory

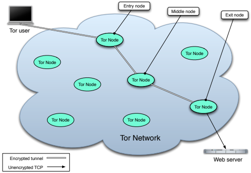
www.torproject.org

Privacy Enhancing Technologies Symposium, 23–25 July 2008, Leuven, Belgium

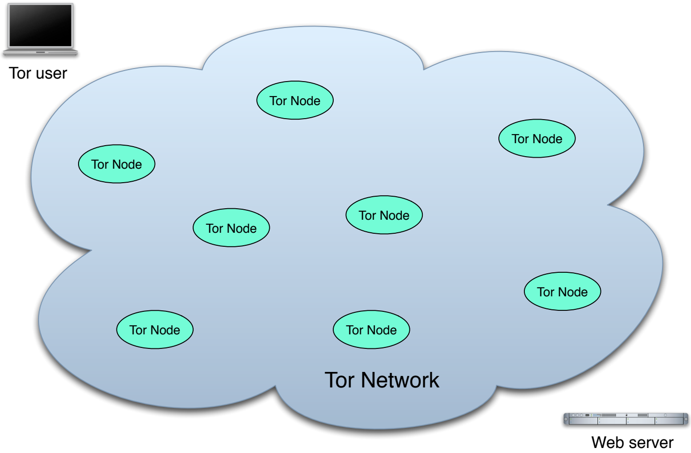
Tor is a low-latency anonymous communication network

- Transports TCP streams: mainly web-browsing (see yesterday's talk)
- Popular: ≈ 2 500 nodes; ≈ 250 000 users
- Limited access control: anyone can run a Tor node and route user's traffic
- Traffic routed through three nodes, with onion routing to give bitwise-unlinkability
- Because of the low latency and limited padding, end-to-end traffic correlation breaks unlinkability

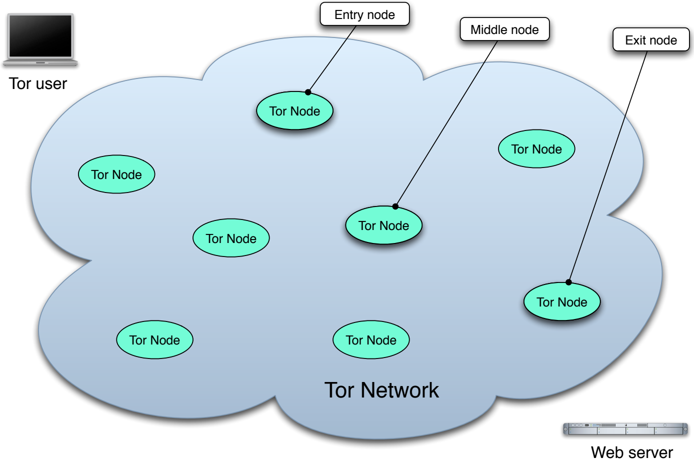




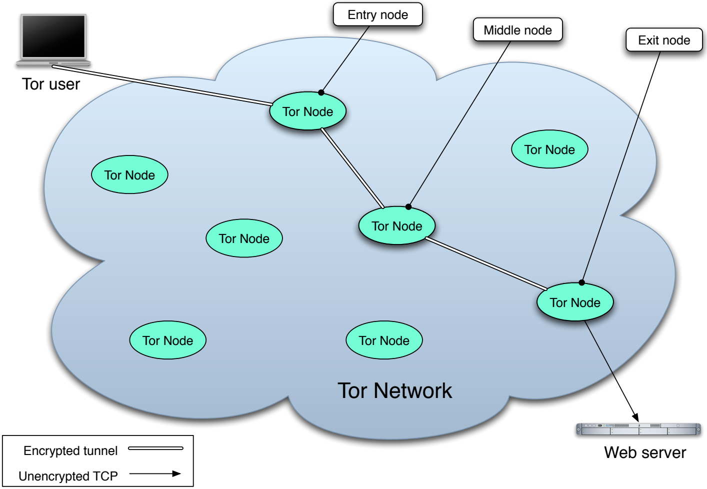
Paths are selected by the Tor client, to resist route capture attacks



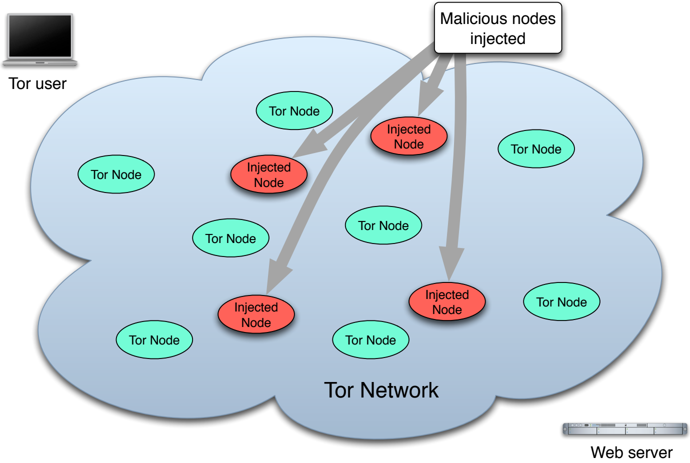
Paths are selected by the Tor client, to resist route capture attacks



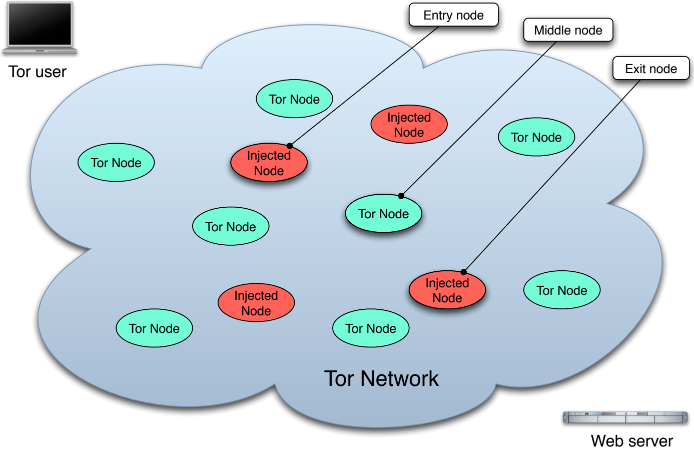
Paths are selected by the Tor client, to resist route capture attacks



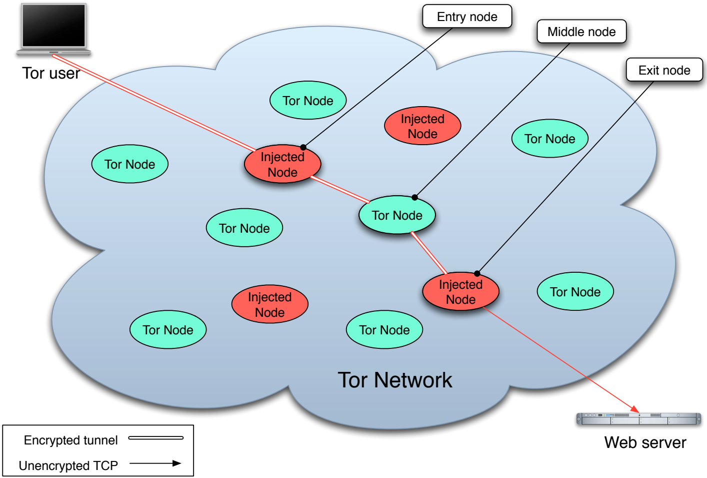
By injecting nodes, an attacker can compromise some proportion of paths



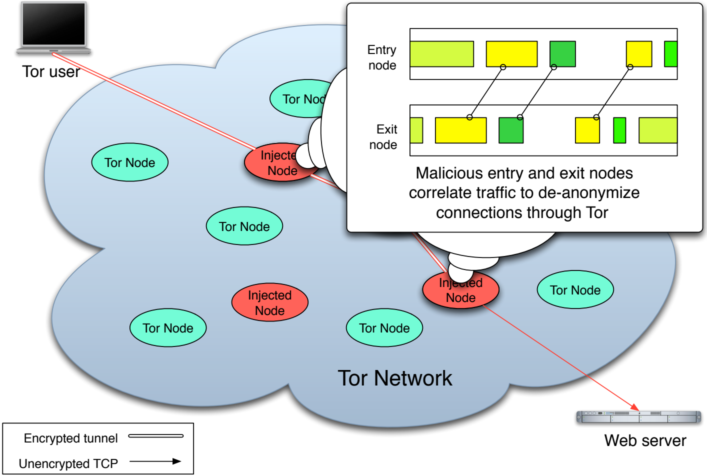
By injecting nodes, an attacker can compromise some proportion of paths



By injecting nodes, an attacker can compromise some proportion of paths



By injecting nodes, an attacker can compromise some proportion of paths



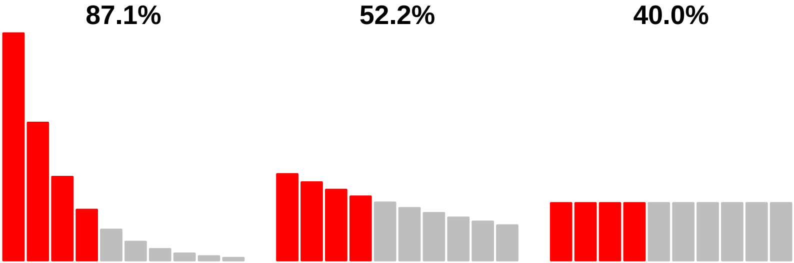
Malicious entry and exit nodes correlate traffic to de-anonymize connections through Tor

The node-injection attack success rate depends on path selection algorithm

- If nodes are selected uniformly at random by the client, the probability that a path is compromised is approximately $\left(\frac{n}{N}\right)^2$, where n is the number of injected nodes and N is the total number of nodes
- Selecting nodes uniformly is bad for performance, so Tor weights the selecting probability by bandwidth
- An alternative selection algorithm, proposed by Snader and Borisov (S-B), weights nodes based on their rank in the bandwidth order
- The S-B variant is also tunable, depending on a client's preference for anonymity vs. performance
- Tor's actual path selection algorithm is more complex (it also takes into account node stability, network location, and history); full details are in the paper

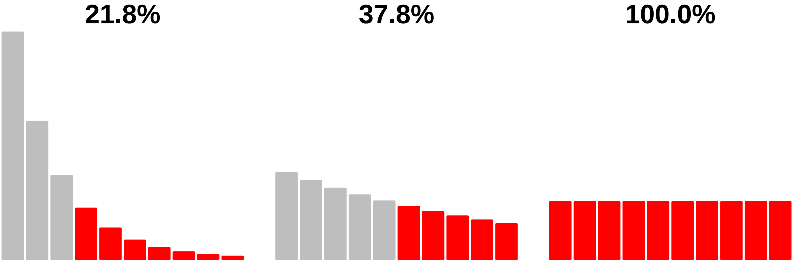
Metrics allow path selection algorithm security to be compared

- One frequent choice is entropy H of the node or path selection probability distribution
- Normalized entropy S effectively measures the skew of the distribution; the more uniform the better (Gini coefficient similar)
- This effectively assumes that an attacker can compromised a fixed number of nodes n, independent of selection algorithm



If *n* depends on the algorithm, entropy no longer measures security

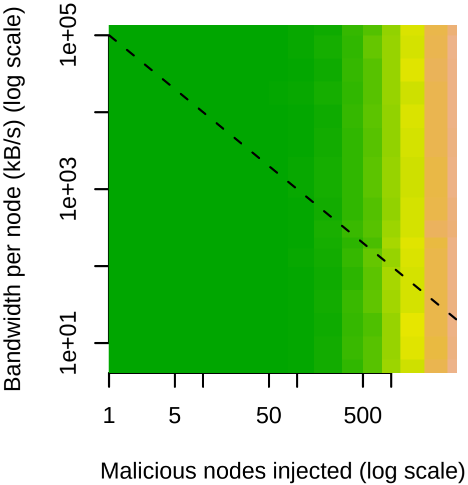
- If the number, and rank, of nodes compromised depends on the path selection algorithm, comparing entropy of algorithms can give misleading results
- This is the case in Tor, because injecting a node in a given position requires an investment of bandwidth



Instead, we look directly at probability of path compromise

- Since the Tor path selection algorithm is very complex, we build our simulation on top of the real network data
- The various selection algorithms use two main properties of nodes when deciding the weight:
    - IP address
    - Bandwidth
- We therefore measure the percentage of paths compromised for a given attacker investment of n nodes and b bandwidth per node
- Adversaries may also be able to pick different values of n and b subject to some constraints (e.g. budget)
- Similar analyses, such as compromising rather than injecting malicious nodes, will lead to similar results.

Vulnerability of uniform selection depends only on number of nodes

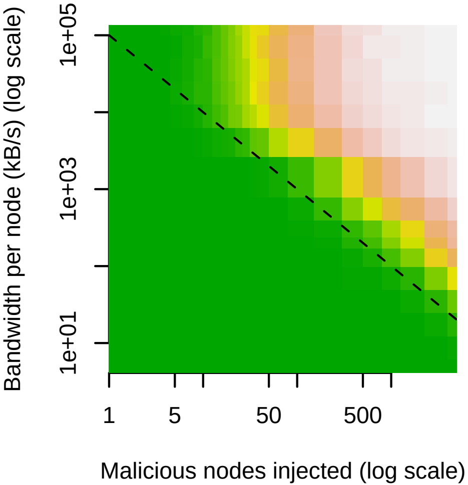


Dashed line shows limit of capabilities for an attacker with fixed total bandwidth budget and unlimited IP addresses

Attacker compromises 80% of paths with investment of 5 000 nodes

One small botnet examined has ≈ 2 000 nodes on ≈ 1 000 distinct /16 networks – would give 40% path compromise

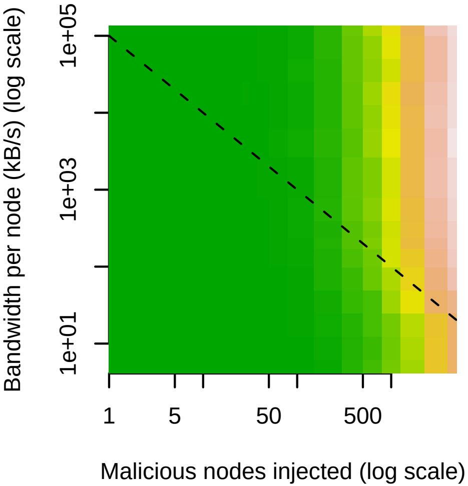
Vulnerability of bandwidth selection depends on bandwidth investment



All points on the line are approximately equivalent (10% compromise rate)

Same botnet would give $<$ 5% compromise if each node could carry 20 kB/s or 40% if each could carry 256 kB/s

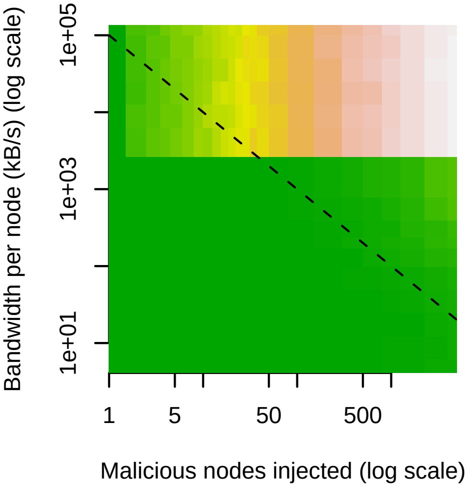
S-B(1) is almost identical in security to uniform selection



S-B(1) gives almost uniform node selection bias

Best strategy for attacker is to generate a large number of nodes

S-B(15) is secure until the attacker occupies the top few positions

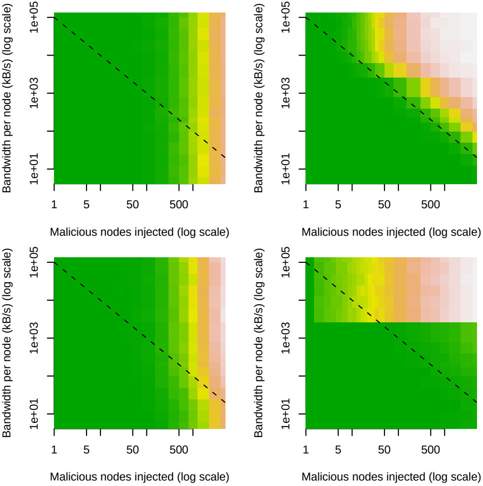


Optimum point is 24 nodes, which gives 50% compromise rate. Above this point the compromise rate drops dramatically

Outside of the top area, security is better than bandwidth weighted – 5% vs. 10% path compromise rate

This model is comparable to cascades, which rely on a small number of trusted nodes.

No one selection algorithm is optimum against all adversaries



Uniform/S-B(1) good against adversaries with few nodes

Bandwidth weighted good against adversaries with limited total bandwidth

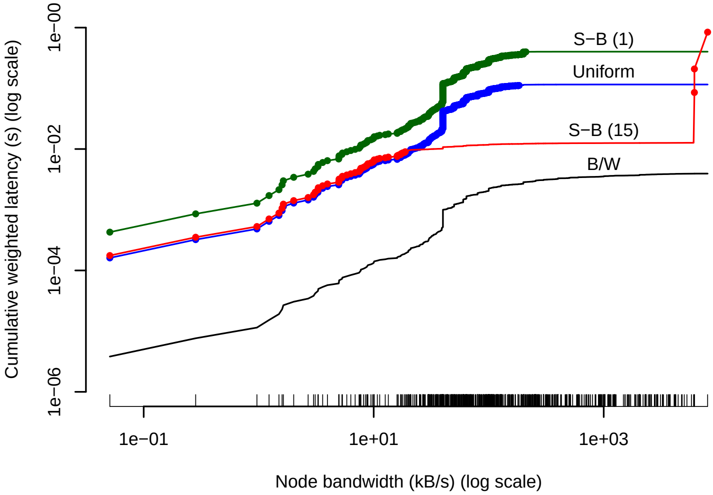
S-B(15) good against adversaries who cannot get a few nodes in the top of the bandwidth ranking

Best path selection algorithm depends on threat model: entropy and Gini coefficient only tell part of the story

Queueing theory model gives latency estimate for simplified network

- Path selection algorithms must give good performance as well as security
- Modifying only the client does not take into account network-level effects
- One-hop Tor network can be modelled as a collection of M/D/1 queues (details in the paper)
- Results depend on distribution of node bandwidth and network utilization ($\approx$ 50% in current Tor network)
- For the bandwidth weighted algorithm, the bandwidth distribution terms disappear, and only the network capacity, utilization and number of nodes affect the result

Bandwidth weighted algorithm is the best in terms of performance



Conclusions

- Security of path selection algorithms cannot be summarized by one number
- The threat model (attacker investment capabilities) radically affects which scheme is most secure
- If attacker is limited by total bandwidth (realistic assumption) the bandwidth weighted scheme, designed for performance, is good for security too
- System-level effects are important in measuring performance of path selection algorithms

Future work

- What is a realistic threat model for Tor – bandwidth costs but so do nodes (botnet figures could be used here)?
- How realistic is the queueing theory model?
- What is the optimum path selection algorithm for performance?