ON DECOY ROUTING
AND BUILDING A PRACTICAL INFRASTRUCTURE
STRUCTURE

• Motivation
• Decoy Routing
• Software-Defined Networking
• Do Decoy Switches Help?
MOTIVATION
• Governments and ISPs censor data.

• What data?

  • Blogs, Political Parties, Individuals, NGOs ...

• Why?

  • National Security, Values, Stability
• Standard Approach: Onion Routing
• Build route of relays
  • Nested encryption
  • Only entry node sees source location
  • Only exit node sees destination

Image by William Hua, McMaster University
• Caveats:
  • Use SOCKS 4a proxy (Else DNS server sees)
  • Use HTTPS anyway
  • Must find entry relays to use Tor
• Entry relay list publicly available from directory
  • Adversary sees, blocks

• Make some secret entry relays not in directory
  • Tor bridges
  • ... still need to be discoverable
  • Adversary sees, blocks

• Winter/Lindskog (2012):
  • China etc. use deep pattern inspection to detect “handshake”
  • Make Tor traffic look like something else
  • Skype etc.
  • Obfsproxy ...
• This is an arms race

• Pluggable transports are not immune to detection
  • obfs, obfs2 deprecated ...
  • now: obfs3, scramblesuit, fte, obfs4

• Can we find another solution?
DECOY ROUTING

- Hosts are easily filtered by IP address.
- Routers, not so.
  - Packets have no router addresses
  - IP network cannot control upstream path
  - Use well-placed router. Block Traceroute.

Decoy Routing, Karlin et al, FOCI 2011
• Basic idea: IP addresses are nonsense
• Just used to get a flow through decoy router
• **Covert signal** to router to hijack
  • Port knocking, Payload lengths …
• TCP session hijacked, sent to decoy proxy
• TCP options (window scale, SACK) passed encrypted (TLS client 28-byte random field)
• Notable implementations
  • Decoy Routing
  • Telex
  • Cirripede
  • TapDance
• ... Problems with Practicality!

• Cirripede: uses a registration server
  • all traffic sent by decoy router to server
  • could not be implemented

• TapDance: let the message through
  • do without inline blocking. It’s too hard.
• What do we need?
  • Smart, controllable router ... complex operations
  • Able to handle large-volume traffic at line speed
  • For example, TapDance implemented on 16-core server attached to mirror port on HP switch
SOFTWARE-DEFINED NETWORK

- Basic idea:
  - general purpose forwarding devices
  - data plane simple, configured remotely
  - controller - switch separation

Image from aryaka.com
- Simple control plane - data plane interface
  - Standard: OpenFlow
- Switch:
  - Flow tables
  - Channel to controller
- Multiple flow tables, visited in order
  - Multiple actions can be applied to a packet
  - Push/pop labels, redirect at will
  - No encrypt/decrypt
• Controller makes decisions ...
  • unknown flow? Send packet to controller
  • Cirripede
• Who IS controller?
  • Assuming ISP as adversary ... isn’t controller under adversary control?

• Can we perhaps build decoy routers using SDN infrastructure?
  • Once out of the censoring domain, we can be the ISP!
  • ... do we need to be given controller access?
- Switch connections can be established with multiple controllers.
  - Default: OFPCR_ROLE_EQUAL
  - Hand-offs handled by ... controllers
  - Switch dumb
    - reports all
    - no arbitration
- Security not great - seems to be getting worse
  - OpenFlow 1.0: TLS
  - OpenFlow 1.4: TCP (or TLS ... but most take the easy road)
  - Pwn switch: dpctl
  - Pwn controller: REST APIs, poor passwords
DO DECOY SWITCHES HELP?

- Simple operations ...
- Switch just does traffic redirection
- inline blocking etc. easy now
- Heck, if we really want, we can do complex stuff
- Controller can detect handshakes using DPI etc.
Decoy Routing Architecture Involving SDN Controlled Switches (Acting as Decoy Routers)
• What are the major wins?

• Speed, for one. L3 (NAT-like) rather than L5 proxy function.

• Choice. We now have multiple decoy routers.

• General SDN wins: administration

• Load balancing, Failover, Error detection
• Blue-sky: use controller to get a directory service?

• Right now - simply redirect client request (covert "give me choices" message) to directory server, to get overt destinations

• Hiding tracks

• Two SDNs ... X decoys the messages between Y’s controller and switches, and vice versa
• How far have we got?
  • Not very - simulating NAT vs proxy performance on Mininet ...
  • Next step : evaluate on real iron (ExoGENI)
  • Long term : cascade routers, detect misbehavior, see resilience to DoS
Ideas, Questions, Todos, ...

-Thanks!