

Anonymization

- Two types of data we want to anonymize
 - Traffic traces – so that we could share them and protect our privacy
 - Routes that traffic takes to get from us to destinations

1

Trace Anonymization

- Contents of packets are not released
- People obscure IP address information to preserve their privacy
- Sometimes they will obscure port information too, but frequently not
- What is left are timestamps, packet protocol and packet size

2

Obscuring IP Addresses

- Usually just a random address is picked for a given IP address in the trace
- Problem arises when we want to anonymize several traces – addresses will not match
- We can save this information in a dictionary and invoke it later on

3

Obscuring IP Addresses

- For some research it is important to preserve prefix relationship, i.e. addresses A and B share the same prefix
- Can we do this without divulging too much information?

4

TCPdpriv

- A free program TCPdpriv performs prefix-preserving tcpdump trace anonymization
- For the first address in trace – A – it generates the anonymized pair $anon(A)$ randomly
- For any other address – C
 - If C and A have k first bits in common, $anon(C)$ and $anon(A)$ will also have first k bits in common
 - The rest will be randomized

5

TCPdpriv

- Problem:
 - Different traces will map same addresses to different numbers
 - Even with the dictionary, the mapping depends on the processing order of the traces
 - We cannot parallelize the process

6

Cryptographic Anonymization

$$f_i(a_1 a_2 \dots a_i) = L(E(a_1 a_2 \dots a_i, k))$$

Where E is a block encryption, such as DES or AES and L is a function returning the least significant bit

J. Hu, J. Fan, M. Amar, S. Moon, "Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme", Proceedings of ICNP 2002

7

Possible Attacks

> Cryptographic

- > Attacker obtains several mappings (raw address, anonymized address) and tries to infer key k
- > For all anonymized addresses, if they match the prefix of one of the pairs the attacker has, he will know this part of the raw prefix – we want to be sure the rest remains unknown
- > Paper proves that this only depends on the security of encryption protocol E

8

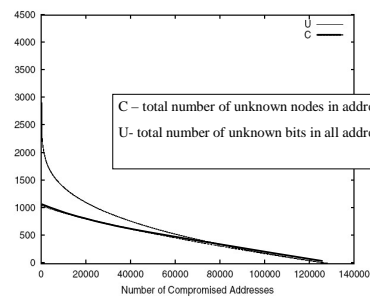
Possible Attacks

> Semantic

- > Attacker infers raw addresses from the anonymized trace by performing frequency analysis
- > He will discover some prefix mappings, and some will remain unknown

9

Random Attacks



10

Random Attacks

- > As the attacker gains more and more addresses, he learns a bit more nodes in the address tree
- > Each address has high probability that some of its prefix bits will be revealed but not very many

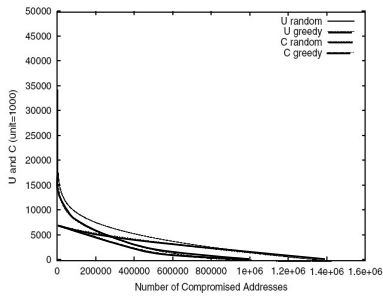
11

Greedy Attacks

- > Attacker chooses an address at each step that gives him the greatest gain with regard to C or U
- > This is the optimal technique

12

Greedy Attacks



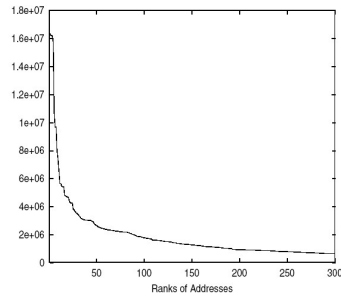
13

Frequency Analysis

- Attacker infers popular addresses from their frequency in the trace
 - DNS servers
 - Popular Web servers
- Uses this information to learn some other address mappings

14

Frequency Analysis



15

Frequency Analysis

- Worse than random compromise – the attacker does not gain from frequency analysis

16

Open Problems

- If the attacker can send some packets in order to have them anonymized, he can retrieve address mappings

17

Anonymizing the Route

- Standard uses of encryption can keep the contents of data private
- Privacy concerning location/identity of users is usually ignored
- Inherently a difficult problem, since location and identity are core to routing and delivery

18

Infranet

- Some countries and providers censor their Internet service
 - They don't allow access to certain sites
- Infranet is a way around this
 - Users send requests to inconspicuous third-party server that sends them to the real server and returns responses

N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger
 "Infranet: Circumventing Censorship and Surveillance",
 Proceedings of 11th USENIX Security Symposium

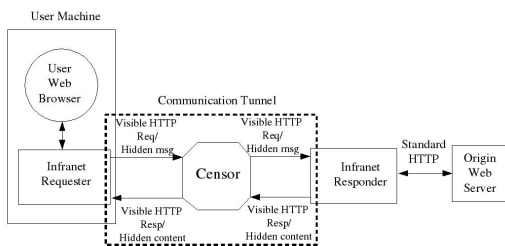
19

Infranet

- Communication between an Infranet server and its clients is hidden in ordinary-looking HTTP requests
 - Clients hide their requests in HTTP requests for Infranet server Web page – censor mustn't know the identity of an Infranet server
 - Servers send replies hidden in images using steganography
 - To ensure deniability server always sends hidden data, regardless of whether the request was for a forbidden page or not

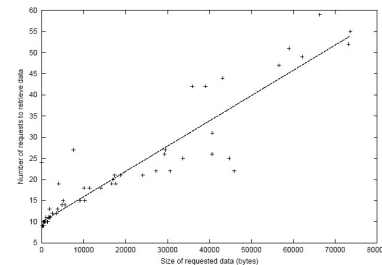
20

Infranet



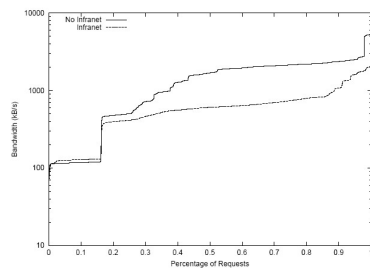
21

Infranet



22

Infranet



23

Onion Routing

- Hide the entire route that the packets take, not just the destination
- Do this with help of several other routers

P. F. Syverson, D.M. Goldschlag, and M.G. Reed,
 "Anonymous Connections and Onion Routing," Proc. IEEE Symp. on Security and Privacy

24

Onion Routing

- Users send requests to one of the cooperating servers – a proxy/onion router that is securely managed
- Proxy router generates a routing path to the destination over the overlay of other participating routers
 - Encapsulates the data for each node in the path with next-hop information cryptographically.
- Each time a node is traversed, one of these “layers” of encryption is removed.

25

Onion Routing

- To avoid revealing end points, onion routers will send the message a few hops further
- Onion routers have fixed connections to their peers, each knows only its neighbors
- Data changes between hops so it cannot be tracked
- Routers can also emit “noise” at all times to hide communication activity

26

Threat Model

- All traffic is visible
- All traffic can be modified
- Onion routers may be compromised
- Compromised routers may cooperate

27

Acknowledged Attacks

- Modifying or replaying onions will result in the end plaintext either not being delivered or not being readable.
- It does not result in sensitive information being disclosed or made obvious.
- But, this implies denial of service vulnerability.

28

Replay Attacks

- To combat replay attacks, onion routers drop duplicate onions
- Each router keeps a hash of every onion it passes along

29

Cost

- The number of asymmetric encryption applications is equal to twice the number of hops throughout the path for *each* packet
- Route is suboptimal

30