Advanced Topics on Privacy Enhancing Technologies
CS-523
Censorship Resistance Exercises

1 My Directory

Alice decided to build a new anonymity network similar to Tor, but with no distinction on where can a node operate, i.e. all nodes can be guard, middle, or exit nodes at the same time. Alice has not figured out how to inform users about these node's addresses handles them. By analyzing and comparing the trade-off of the following approaches, help Alice to decide how to publish nodes information. You need to consider privacy and censorship resistance (availability).

1. Alice signs the list of all nodes and ships it with the application. We assume that all users verify the checksum of the application.

   Solution:
   The censor downloads the programs, extracts the list of all nodes, and blocks them. There is no privacy issue in this approach.

2. Alice runs a mail-server and automatically responds to each mail with a list of 10 random nodes.

   Solution:
   If Alice is malicious then she can partition the network. In other words, she gives a disjoint set of nodes to each user and exploits the fact that users can only access the nodes in their email to identify them. For example, Alice sends $\{s_1, s_2, ..s_{10}\}$ to Bob and $\{s_{11}, s_{12}, ..s_{20}\}$ to Eve, and a random subset of $\{s_{21}, s_{22}, ..s_{100}\}$ to anyone else. Just by checking the exit node, Alice can determine the sender. The censor can send a large number of emails to Alice to extract all nodes. Extracting nodes by emails is harder than having a list, but it's still feasible for the censor.

3. Alice runs the mail-server as before but instead of using fresh randomness for each email, she uses the sender's address as a seed.

Solution:
The fixed seed does not change the privacy implications of the system if Alice is malicious. However, it makes the node extraction harder. If Alice does not allow emails providers which are under the control of the server, like only accepting Gmail accounts, then the effort necessary to create emails limits the censor extraction.

4. Alice asks Bob and Charlie to run identical mail-servers with their respective secret key. Every user has to mail all three and check their response's signature. Only if all three emails contain the same set of servers, the user will trust them.

   Solution:
   This has no impact on availability. However, the presence of Bob and Charlie prevents Alice from partitioning the network and changes the trust to any trust setting. It's important to note that even honest but curious directories have an advantage in identifying users when users receive a partial view of nodes.

   Any trust model states that as long as one of Alice, Bob, and Charlie is not malicious or does not collude with the others, then the approach is secure. The user does not need to know which party is honest.

2 Domain Fronting

Consider the following setup in a censored country:
`software-download.microsoft.com` is allowed
`https://blocked.azureedge.net` is censored
ISPs in the country will block HTTPS packets that declare `blocked.azureedge.net` in their server name indication (SNI) field. Both services, however, are hosted on the same infrastucture (Microsoft Azure). In order to evade the censorship and access `blocked.azureedge.net`, an internet user in the country can try to obtain the IP address of `software-download.microsoft.com` through a DNS query, and then send the following HTTPS packet to this IP:

```
TLS
SNI: software-download.microsoft.com
...

    HTTP
    GET / HTTP/1.1
    Host:  blocked.azureedge.net
    ...
```

Per the externally visible SNI the packet appears as if it is directed to `software-download.microsoft.com`, but the hope is that Microsoft Azure redirects the packet to `blocked.azureedge.net` within the same connection. This censorship circumvention technique is called domain fronting.

1. What server-side conditions enable domain fronting? How can Microsoft Azure prevent people from using domain fronting through their websites?

   Solution:
   First, both the unblocked and blocked website have to be part of the same multi-tenant host. Second, the server needs to support requests where the SNI is different from HTTP host. In general, the servers might not support such requests, and deny them. Google and Amazon used to support non-matching SNI and HTTP host values until 2018, when they decided to disable this functionality. This crippled the censorship circumvention capabilities of multiple pieces of critical software such as Signal and Tor. As of 2020, Microsoft Azure is the only big CDN provider that still supports domain fronting. You can try it yourself in a Linux terminal:

   ```
   wget -q -O - https://software-download.microsoft.com --header
   "Host:  meek.azureedge.net"
   ```

   This will access `meek.azureedge.net`, but the request's SNI will be declared as `https://software-download.microsoft.com`.

2. How can a censor prevent domain fronting? At what cost?

   Solution:
   The censor needs to block `microsoft.com` for everyone. Domain fronting as a tool for censorship circumvension relies on the fact that blocking the "front" website (the one declared in the SNI field) will result in too much of collateral damage for the censor.

3 Decoy Routing

Alice wants to access a censored site `blocked.com`, in the presence of a state-level adversary, Eve, that is monitoring her traffic. She makes use of a decoy routing system, which routes her connections to the uncensored site `notblocked.com` to `blocked.com`. Bob is another user who wants to access `notblocked.com`. He doesn't use the decoy routing system.

Consider Eve as a passive adversary – such an adversary only monitors client traffic and does not attempt to inject or modify traffic.

1. How can Eve use traffic analysis to determine if Alice was using a decoy router?

   Solution:
   Eve could potentially use timing attacks to determine whether Alice was using a decoy router. At the decoy router, the paths to the censored and uncensored destination would diverge and lead to differences in the network latency. Eve can first measure the latency of a normal connection to

notblocked.com. She compares this latency to the latency she observes when Alice accesses notblocked.com. If there is a significantly large discrepancy in the latency values, Eve can infer that Alice is actually not connected to notblocked.com, but is using a decoy router.

2. Would it be possible for Alice to reduce the impact of timing analysis performed by Eve?

   Solution:
   The choice of (uncensored, censored) site pairs can have an impact on the timing analysis. If the latency of a client accessing a censored site via a decoy router and the uncensored sites are similar, it is harder for an adversary to determine if the client is using the decoy router. Note that the decoy routing process itself can add latency overhead to the process, which also needs to be taken into account. However, trying to find appropriate censored, uncensored site pairs that have similar latency distributions is not trivial.

3. Consider Eve as an active adversary now. Eve has recorded Alice's and Bob's TCP packets sent to notblocked.com. She decides to replay this connection over a route that does not contain decoy routers. Does she see a difference in the response for Alice's and Bob's connections? Why?

   Solution:
   Bob had a legitimate visit to notblocked.com. Thus, when the website sees Eve's replay, it considers it as a duplicate and sends back a TCP duplicate acknowledgment. However, Alice actually visited blocked.com. Hence, she does not have a connection with notblocked.com. Thus, the website sends back a TCP reset packet. This helps Eve determine whether Alice was using a decoy routing.

4. Consider Eve as an active adversary that can switch the first hop of the paths that Alice and Bob's messages take. Alice and Bob have established connections to unblocked.com, when Eve decides to implement a path switch. The new path does not contain decoy routers. How are their connections impacted? Would Eve be able to determine whether Alice is using decoy routing?

   Solution:
   Eve is implementing a crazy Ivan attack. Since Bob had a legitimate visit to unblocked.com, the path switch does not impact him. However, in Alice's case, the website can send back TCP reset packets as before. Not only this, if Alice decides to retry setting up a connection with a new path, this retry can be seen as further evidence that Alice is attempting to use decoy routing.