headers or data payloads, the 3rd party infrastructure or the final server would be able to read any potentially covertly transmitted data from the compromised machine.

But the attacker (who controls the cat-pictures service server), even though receiving some of the user sensitive data, e.g. disk encryption key, might not be able to figure out which user do they belong to. Of course the user might have plenty of identifiable information on their laptop, and the malware might be smart enough to search around for them and include them with the blobs sent over the covert channel. Theoretically, if the user was careful enough this might not be the case, but in reality expecting the user to be so careful with regards to *all* of the activity performed on their laptop, might be unacceptable for most users.

Typically users would be willing to be careful only with regards to some of the *domains*, while would like to "live a normal life" in others. Operating systems such as Qubes OS [16] try to resolve this problem by using Virtual Machine-based compartmentalization. Sadly in case of malware operating in the ME or SMM[7] the Virtual Machine technology (even augmented by technology such as Intel VT-x and VT-d) is of little help.[8]

On the other hand, forcing the attacker's malware to modify only high-level protocol payloads to leak data might already be considered a significant win. The higher protocol the attacker needs to intercept, the higher the complexity of the malware, which increases the probability of getting caught by curious users or administrators.

In addition the attacker has little control over which servers or infrastructure she should control in order to be able to receive stolen data from a given user.

Scenario 3: Unconstrained Internet access?

Not every user would like to forward all their networking through Tor or even a fast VPN gateway. The primary reason not to do that might be the limitation on the bandwidth and latency imposed by such proxies.

[7] Although systems that properly use compartmentalization might make it very hard for the SMM to ever get infected. On the other hand, they can do nothing against the backdoors built in by vendors right from start.

[8] Admittedly Intel VT-x allows for SMM sandboxing using Dual Monitor Mode, although in practice there seem to be lots of problems with this approach, as the author has discussed in [10].