I am going to talk about some problems with IPSec, especially about how IPSec policies are specified, and whether that matches the kind of security requirements we had in the beginning, and which were why we introduced IPSec. Maybe everyone is familiar with IPSec, but it might be useful to have the same kind of picture in mind, because you can look at these things from various points of view.

Most of the research in the past has been on things like, is this security protocol secure, is this authenticated key exchange protocol secure. In this talk I am just going to assume that you have some authenticated key exchange protocol and that it is secure. Also there’s been lots of work on the cryptographic algorithms: is this encryption (or Mac) the most efficient and the most secure one known, or the best compromise between the two. And that’s another thing that I’m not going to talk about, I shall just assume it happens. I’m more interested in the architectural type of thing, like the security policy and how it is specified.

IPSec is intended for end-to-end authentication and encryption in the Internet, but it’s not quite clear what end-to-end means for IPSec because it’s mostly used for gateway-to-gateway, or sometimes for host-to-gateway, and not really from host-to-host. In this talk I’m trying to look at what happens if it really is used end-to-end, host-to-host, without any security gateways. The goal of IPSec as stated is usually end-to-end security, but in fact the only application where it’s really used in this way is for VPNs, and one might then ask, why is this, and can we actually use IPSec for securing all the communication on the Internet, or in your own home or corporate network.

We’ve done an experiment with this, not me personally, but I’ve seen a deployment of IPSec to a corporate network of tens of thousands of hosts. All traffic basically is protected with IPSec with a very small set of specific exceptions, and it’s first of all difficult to configure it, to get it running, but as we then found out, it doesn’t in fact do what you expected it to do, it doesn’t provide you with the kind of security you wanted. One of the main design principles in IPSec is that it’s a transparent layer, it hides the security mechanisms from you, and that’s part of the problem, or at least causes difficulties.

The structure of IPSec is such that you have these protocols for key exchange, and you have then protocols for protection your session, and then in order to create session keys for the sessions, you run the key exchange authentication protocols, and you have a security policy that guides this process. So it tells whether certain IP packets should be protected in some way, that is, should you create a security association for it, and encrypt, and authenticate it, or
should you discard it, or (what’s important in this talk) should you bypass it, should you let it go without protection. These policies are specified by the system administrator, and then IPSec tries to transparently implement them for all your applications.

The first and main problem that I’m going to present here is that IPSec is somehow tied in with the name resolution. So, here is a PC, and there’s the applications, and the operating system, and the operating system has IPSec inside it (I’m trying to abstract as much as possible). And over there you have the peer computer that this PC wants to talk to. What happens when an application wants to talk to this peer computer, let’s call it server B? First the application says, well, since I want to talk to server B, I must find its IP address, so it asks the operating system to resolve the name into an IP address, and the name service will get a query saying, where is server B, and it will return the address, it’s at this address. The application gets the address and then it will ask the operating system to connect to that address. And there’s an IPSec policy that says, well this kind of packet should be protected therefore I’m going to do a key exchange and then once I have the session keys, I’m going to protect the data with those session keys.

All that looks fine, but obviously I’m pointing out the name resolution there for a reason. IPSec, at least to some extent, depends on the name resolution being secure, so let’s see what happens if the attacker can manipulate the DNS responses. Here is the honest PC trying to connect to a server but unfortunately an attacker is intercepting the traffic, and this is a well-known kind of problem. It’s maybe not the obvious thing you would first think, but it’s not new. Now the application asks for this name to be resolved and it gets the wrong IP address, because it’s the attacker now feeding the response here. The application then asks the OS to connect to this IP address, and now you notice if you look at the details of the IPSec policy that for the true address of the server there is a policy of protecting the traffic, authenticating the server, but for this address the attacker gives, because that’s the attacker’s real address, it’s just a node on the Internet, there is a BYPASS policy saying, no protection is required for all these other nodes, like websites, on the internet.

So in fact, there won’t be any key exchange or IPSec protection taking place, all the traffic is sent plain text between here and there. The application did not get the protection that it was supposed to get based on the policy that the administrator had specified. So there’s a problem in IPSec that your policy selection depends on the secure DNS, and by spoofing DNS responses an attacker can either force you to use a completely insecure connection, or just use different parameters for the authentication.

Now you might say, well we should not have such a default BYPASS policy there, but in reality while you want to authenticate the host in your own organisation, you certainly don’t want to use IPSec with everyone else on the Internet, so you usually have a default BYPASS policy at the end of your policy database. And the policies are resolved in this order, so you first look, do I match this policy, and then if you don’t then you go to the last one. And