Why Bother Securing DNS? (Transcript of Discussion)

Dieter Gollmann(✉)

Hamburg University of Technology, Hamburg, Germany
diego@tu-harburg.de

As Bruce so kindly said, I was volunteered to give the first talk after he had successfully extracted more than two lines of a position paper from me. I will talk about what I see currently happening in and around the domain name system. I will start with a story, some of you might have heard about it, although I don't know how far it reached beyond Germany.

In Germany you have the Chaos Computer Club, which they quite proudly will tell you, has more or less the status of an NGO in Germany, they're not the evil hackers, they are the good guys. And because they're the good guys, and you can't trust the enemy running the entire infrastructure, they run their own DNS resolver, so you can have a proper trusted resolver. The software version they're running, Dan Bernstein's DJBDNS, which is Open Source, and therefore good.

Back in 2009 someone had a look at DJBDNS and found that there were some features in it that made it particularly susceptible to cache poisoning attacks. They contacted the author who was no longer interested in maintaining DJBDNS, they published their own patch, but the Chaos Computer Club is above installing patches, so they were hacked. That as a starting point, and here you can read on, and I guess it also has an English version where if you don't trust the automatic translation tools from German will tell you what had happened.

So I'm talking about DNS, a distributed directory system, mapping host names to IP addresses, authoritative name servers in charge of making statements about their domain, resolvers, caching whatever the authoritative name servers have told them. Authenticating the authoritative name server in a non-cryptographic fashion. Sending a query ID, maybe at a random port number, anything that comes back on that port contains the query ID, contains the host name, is the authoritative answer. To seasoned cryptographers this is of course ridiculous, and we are not surprised that there is a long history of cache poisoning attacks working on the simple principle of guessing this 16-bit number. Guessing is easy if the 16-bit number is generated by a counter.

It is moderately easy if the resolver runs several queries for the same host and at the same time, then you can use the Birthday Paradox to go down from a search space of 2 to the 16 roughly to a search space of square root of 2 to the 16, that was published by Vagner Sacramento in 2002 on BIND 4 and BIND 8. Strange enough it took another seven years to figure out that DJBDNS had the same problem, and even more strangely if you look at the world, it took another

B. Christianson et al. (Eds.): Security Protocols 2014, LNCS 8809, pp. 9–17, 2014.
DOI: 10.1007/978-3-319-12400-1_2

five years before someone really exploited this known vulnerability. 2008 Dan Kaminsky's famous paper explaining that if your cache poisoning attack failed you could immediately restart it, you did not have to wait for cache entries to expire, by querying for a random host name in the bailiwick of your target host, and eventually you would win the game using additional resource records. At that time some faces in the community, running the global Domain Name System, went very pale. This was Armageddon. They finally realised this type of non-cryptographic authentication doesn't work. We have been talking about DNSSEC since the late 1990s, and now we have to get serious about it.

So, a habit in our community, we see a problem, we run for cryptography. We have an authentication problem, we run for digital signatures. If we do that we need verification keys, we need a public key infrastructure for verification keys, and the public key infrastructure for verification keys that is emerging, as far as I can see, more or less mirrors the hierarchical structure of the Domain Name System, and the top level domains, generic top level domains, country code top level domains, take on the role of routes of trust, which is OK, which is perfectly OK because there are anyway roots of trust. They will say where to find the next authoritative server in the DNS hierarchy. If they want to cheat they could cheat before DNSSEC.

What have we achieved? We have protection against outsiders, proper cryptographic protection against outsiders. We do not have protection against insiders, insiders meaning authoritative name servers making false statements. They're allowed to make statements about hosts in their domain, and then they can claim that an IP address that does not belong to a host in their domain belongs to them. There is a range of papers on this topic starting from the mid 1990s. I think Drew Dean and his Princeton colleagues were one were one of the first to discover such a DNS rebinding attack.

One of the defences, potential defences, would be to write to the IP address and ask, I think you are in this domain, is this true? In past papers I have suggested, so instead of using authentication, we might treat this as authorization to connect. We ask the host at that address, are you happy to accept traffic for this domain?

Frank Stajano: In the case of this possible defence you talk about, if the malicious DNS server is saying, this is the guy you wanted, presumably it's for some attack, at which the thing that responds is colluding with the wrong DNS, so if you ask them, are you the host of this domain, they would say yes wouldn't they?

Reply: It's a different type of attack, it's an attack in the world of same origin policies, where the attacker's script according to the same origin policy would only be allowed to connect back to the domain it came from. But now the bad guy tells it, this IP address is in my domain, and then the browser gets an IP address to connect to, yes, permitted according to my policy, it's in the same domain, and defeats the same origin policy. So that is the background of the DNS rebinding attacks currently. Mike.

Michael Roe: So there's something similar in mobile IPv6 per return routability, where they are not talking about binding domain names by IP addresses, but the long lived IP address for the host relative to its mobile location, there you have precisely this check asking the mobile, sending a message to where the mobile node currently is saying, technically, are you at this house, which is the way against various binding attacks. There are similar sorts of things in DNS where you're careful about doing the reverse lookup.

Reply: Yes, indeed, a general principle, and yes, I could also have talked about return routability in this context. Now, tweet by Mikko Hyppönen, I have found this root certificate, country US, organisation US government, organisational unit DoD, on my iPhone. I can't even get rid of it, what does it do, what is the purpose? I mentioned this story at a workshop and there was a Taiwanese post-doc, and she said very cheerfully, yes, my government also has one of those certificates. And I remember an email from Ross saying, I have been to this conference mentioning one of the Turkish CAs is run by the Turkish secret services, and someone from Turkey violently opposed this view, working with the enemy, yes please.

Micah Sherr: Quick comment. OK, at least it's labelled, I mean, at least it's labelled as the US government.

Reply: Yes, there can be perfectly innocent reasons.

Bruce Christianson: Well it's labelled as the US government.

Michael Roe: So this is potentially an attack because the X.509 certification authority, certification hierarchy doesn't work quite the same way as the hierarchy does in DNS. So I think DNS say the authority for .mil has a key that says they can sign stuff for .mil, you don't care about that because you're not in .mil, they're not going to forge DNS entries for you. But a certificate that might have been intended just for signing certificates for .mil entries, because of the way X.509 does not bind the hierarchies together, that certificate could be used to sign anything, so you might be worried they might sign things outside their domain.

Reply: That is indeed my next point. All these root certificates, and I have produced stories like DigiNotar, who were compromised. Once a trusted CA is trusted it can issue certificates for anyone. I've called it a global point of failure, not a single point of failure, because there are two or three dozen of those points of failure on my machine. In the spirit of this Cambridge Protocols Workshop we had Robert Morris Senior around and hed said, trusted, remember, it meant it can hurt you. So all these trusted route certificates can hurt us. And there are incidents, I've mentioned DigiNotar, where it happened.

Natural response, can we restrict impact, can we make those trusted CAs less trusted? I'm walking back to the Domain Name System where the authoritative name servers can make statements about their domain only. Idea, can't we use this same infrastructure also for TLS certificates? So it would be some point in the domain name hierarchy that issues also certificates, not only for IP addresses, or signatures for IP addresses, but also certificates for public keys, this is known as DANE, DNS based authentication of named identities, RFC6698. I had a student in Hamburg doing his Master thesis whilst working for DFNCert, and the task was to establish, is this more or less secure than the traditional approach. My comment, this is something I see quite often. We have an infrastructure, we have a service, we ask it to do something else on top. DNS was here to resolve domain names to IP addresses, or host names to IP addresses. Now we also want DNS to tell us the public key of the host.

What have we achieved? We have restricted impact, as we wanted. We have achieved that the same entity can lie about your IP address and your public key. Separation of duties, I haven't done my homework, Saltzer and Schroeder, roughly from that time, 1970s, classic, ancient principle in security. And we have thrown it out of the window. Are we asking too much?

So coming to my proposal's questions. What do we need? We need a rendezvous service. Addresses change. With mobility, nodes move round in the network. In time a host might change its IP address. Does it matter when such a service is not available? Yes, because I can't look up your current address. Does it matter when I'm given wrong information? Not yet. I might do the checking independently of the rendezvous service. Cryptography might have a role at the network level below. I don't see it having a role in the construction of the rendezvous service itself, I do not want to trust, as the people who know me know very well.

Interlude. Pekka Nikander in his PhD thesis and in many other places, I have taken this from a draft IETF document as identity. Where does it come from? It comes from Latin identitas, which stems from early Latin, idem et idem, same and same, again, again, identidem, repeatedly. And he goes on in this draft to say that this implies in our understanding, unique ability, to uniquely identify, blah, blah, blah. I have not given you the entire blurb because it then goes on saying, aahh, I think we're moving in the wrong direction, it means something else. So what could one mean with identification?

In our traditional explanation computer security, you login, enter a user name, a password, user name, is identification, you tell who you are, authentication then checks it's really you. If you go to biometrics, they tell you, we have identification, we have verification. Identification means we have a database of fingerprints, and then we check, is this fingerprint from the crime site in our database. Given that we have already overloaded the term, I keep at it, and give it yet another meaning. It tells that it's the same as last time. That is the Pekka Nikander interpretation of identity, identification.

Now fortunately I'm not the only one to be old enough to say, remember around early 1990 s this very agitated discussion about X.509, about, does it make any sense to have globally unique names for access control. Isn't there a fundamental misunderstanding of the concept of names, aren't names meant to identify entities you already know? Is it the case that you only need local names for access control? And all of this comes back here again as far as I can see. So to continue my requirement analysis, I might need an identification service in this last sense, telling, I'm going to the host and check, are you who I believe you are, are you the one I wanted to contact? So the difference to authentication, authentication somebody else tells me who he or she is, and then I check. Now I tell you who you are and you have to confirm it by knowing some secret, for example. So bottom line here, it's easy if I already knew you, it's easy if we have common context, it's easy if we have common context we do not share with anybody else, like a secret key. It's tricky if we've never met before, we need someone to introduce us. So maybe we need in our world introduction services. And same as with the rendezvous services, maybe we have more than one introduction service. If you have independent introduction services, again, we reduce trust on a single service.

Conclusions. When one looks at DNSSEC, when one looks at DANE, are we taking the wrong turn. Are we expecting too much? In particular on DANE, digital signatures protect against outsiders, not against insiders. What are we doing? We've turned the entire DNS hierarchy into insiders we now have to trust. Madness really if you think in security terms, but that is what's going on, and as I've said before, trust is bad for security, we would like to reduce it. Last slide. I'm hearing a lot about critical infrastructures, but I don't think they need security. You should ask, why is the infrastructure critical, because of the services running on the infrastructure, and the services are critical. Secure the services, divide and conquer. And as a final word, I was at the talk by Scott Charney, Vice-President Microsoft, at the Führungsakademie der Bundeswehr in Hamburg, so the Academy of the Officers of the Germany army, and some German army major said, I've just been to the department of defence and the minister said the internet is a critical infrastructure, can we secure the internet? And Scott Charney said, you cannot boil the ocean. The internet was designed to be highly available in the case of a nuclear strike, and we have done quite well maintaining availability, and that is what I'd expect in the main from a critical infrastructure.

So with that I will shut up, up to you to throw your views at me.

Yvo Desmedt: So in your previous slide you mentioned, so the obligation does not protect against live insiders. In 1996 there were two schools that were actually looking at that problem, and so there was a paper by Mike Burmester, and myself, and Kabatianskii, at a workshop organised by Rebecca Wright and Peter Newman, where we exactly said that, but then in the context of certifying authorities. And then Reiter and Stubblebine also wanting to deal with untrusted CA. And later in 2004 Mike and myself published a paper in the Communications of the ACM, Is hierarchical public-key certification the next target for hackers? Obviously what can be said for CAs can also be said for DNS.

Reply: Absolutely.

Tim Goh: The thing is, the authentication things you propose seem to already exist at say various levels, you have TLS if you want to do it above, you've got IPSEC if you want to do it below. The key exchange mechanism already exists, ISAKMP will do web trust file, key exchange if you want it to, RFC recommends such a thing, it's not too far-fetched to implement such a thing. Good luck at getting users to actually establish a web of trust. But there is one major concern that I have that might not be mentioned, embedded systems. Do you have, embedded systems made with DNS, is it reasonable to ask another system to compete key exchange in any useful fashion because considering the systems may be on a 8 meg, 8-bit processor with less than 256 bytes of RAM.

Reply: I wasn't proposing any key exchange.

Tim Goh: Say for example, if you're doing, such an authentication mechanism.

Reply: I'm against using authentication mechanisms, I don't want to use them in the first place, they're useless. That's a very over the top remark. Yes, you're perfectly right. All these authentication mechanisms exist. We have them at the IP layer, we have them at the TCP layer, we have them at the application layer, we have them at the application layer above the application layer. They do not solve the problem. If an insider provides authenticated wrong information, then that is the challenge. Like when DigiNotar was hacked, somebody, the attacker, somebody issued certificates for Google Mail, and via real systems, did all the cryptographic properly, and concluded we are talking to Google Mail, only they weren't.

Tim Goh: But then I am separating the authentication from the trust issue here. Authentication is, I see it as something that needs to happen, but your trust issue is separate.

Simon Foley: Yes, so would you see this as something similar to the Perspectives project at CMU? They developed a browser-plugin that consults a network of notary servers to confirm that others have seen the same SSL certificate that is being presented to you when you visit a website.

Reply: It's more in this direction, yes. And it's also, I think in the context of TLS cache pinning, sorry, certificate pinning. I have been using this certificate with that server in the past, now for some reason I'm getting a different certificate, suspicious. It's no longer the same.

Hannan Xiao: I just wondered, in your introduction service, in your approach, for the first time we still have to rely on the introduction service.

Reply: Of course.

Hannan Xiao: But do you use trust in your introduction service? And somebody introduces someone, so do you use trust.

Reply: So the idea has been around again for ages. When did Phil Zimmermann introduce PGP, early 1990s, earlier, it had this idea. And it works in some communities. DFNCert, the Computer Emergency Response Team for the German research network has an annual conference, and part of their annual conference is a PGP signing meeting, where all the system administrators from German research institutions come and if they have not already shared their keys they can do it at that point, and then they can go back to their institution and introduce maybe certificates or keys to others in the organisation. That's where I see this idea being used, and this idea being used reasonably well. I have my doubts to which extent it can be automated and formalised. And again, if you go back to the research literature there are lots of trust evaluation algorithms. If I give you weight point 35, and you give me a certificate, which you have given a different weight, which weight will I now attribute to the certificate. And then if I have a particular transaction how good must the certificate be. There is a lot on paper, I see little in reality that works.

Virgil Gligor: By definition a trusted third party is trusted by both parties, that's why it's a third party that is trusted. An introduction service need not be trusted by both parties, it needs to be trusted by at most one. So there's a difference of trust there, there is a very clear difference, and you can even formalise that, but I know you don't like formalisation.

Reply: Oh no, I'm a mathematician, I like formalisation, but I like genuine formalisation, not bogus formalisation.

Virgil Gligor: You need not notation only.

Reply: Yes. I keep saying, you don't impress me by using the language of set theory, this was first year stuff in a mathematics course.

Virgil Gligor: So there is a difference.

Reply: Yes.

Bruce Christianson: Virgil's point is a very important one because in order for me to introduce Virgil to somebody I don't have to have any control over Virgil at all.

Virgil Gligor: Correct.

Frank Stajano: One of the things you said in one of the last few slides, I can't remember which one, about the point that establishing that you are talking with the same person you have talked with before, reminded me of the Guy Fawkes protocol, where you can have a strong chain, yes, that's the person who sent me the previous messages, I don't know where it starts from, and there was a big discussion at the time with Roger and Ross about whether you ever know the beginning of it. You can say, well how do you know your mother is your mother, you just know it's the one, for years you've called your mother, but how do you know, at the beginning you didn't have a commission. So that just, this continuity seems to be the authentication rather than the real origin. And something similar to that is in this things like in Android you get some signed installation that, you install a program and you have no clue what it is, it's malware, whatever, but you install it and the next time when it updates at least it's signed with the same key they used on the first time. So it may be completely bullshit, but at least it's the same one as before.

Reply: Yes, I think these are not novel ideas, these are ideas that have been around, but I think the wrong ideas, and simple crypto ideas like digital signatures, attract too much attention, and are not solving the problem we are really facing, in particular, if we are collaborating with the enemy, because the enemy is part of the network, or part of the system.

Tim Goh: Is it worth actually considering say RFC 2408, ISAKMP's original separation of the notion of authentication here, which seems to be conflated here with the notion of initial trust establishment. So before we actually suggest authentication between the host, both hosts, but before that, it's got a complete separate phas that you seem to be calling authentication here, but it's really trust establishment, and establishing some sort of may be signature, maybe, any mechanism somehow to verify as a person, instead of actually having a separate mechanism, and actually defined in those terms. ISAKMP is a horribly painful protocol, but it seems to be exactly what is being asked for here. There is a separate phase that is not, that your talk is not quite interested in, which is somehow given a signature, these two hosts are the thing that we established in the previous phase, but the thing they seem to be interested in is the previous phase where you actually somehow establish trust. So they do briefly talk about mechanisms like that, for at least the IPSEC layer, but ISAKMP can be used for other things.

Reply: I think we have to take that offline, because I didn't talk about authentication.

Yvo Desmedt: The solution has been discovered a long time ago in the reliability community, you just vote, they used it in the station poll, every time that you fly an aeroplane it's used there, and the answer is the same in this circumstance, just vote, don't trust a single party.

Reply: Yes.

Yvo Desmedt: I mean your computer votes. Then it talks to many DNS servers, and then basically your computer votes. That is the vote, and then it decides that the majority can be trusted.

Bruce Christianson: But how do you resist a Sybil attack, it seems very hard to prove that two voters are actually different. How do you prove that two computers are not actually the same computer?

Reply: Yes, that is the core word in my argument, it's the argument that what do we do for availability, we replicate, but we need to be guaranteed independence, how do we know that.

Bruce Christianson: Establishing identity is straightforward along the lines you're proposing. Establishing non-identity seems to be an almost intractable problem.

Virgil Gligor: Maybe a crowd source.

Reply: I think again you will find papers that suggest this as a solution to our present dilemma. Don't use a single server that tells you this is the public key of entity X, ask around, and if you have enough agreement then you take that, but then again, there is Bruce's point, if there are not many people interested in you, and we all collaborate, or we multiply ourselves, we can defeat this mechanism.

Yvo Desmedt: So as a solution in our 2004 paper we suggested that we actually cover all the CAs in that case, depending on the platform that they ran, so if they, for example, were Microsoft or they were Unix, all the Microsoft ones are the same colour, all the Unix ones are the same colour, because if you do a replicated attack, and whether you can attack one or all, is the same, so when you basically, and then you do the same in DNS, so if you say, OK all the CISCO ones, we need them the same colour, so that means that it is easy to hack one of those, and the outsider becomes an insider in that case, and that's how we should just deal with it.

Reply: That defends against the outsider becoming an insider, but it doesn't defend against the insider who is sitting there in the first place.

Yvo Desmedt: But if everybody is against you then you will lose, we know that. There's no solution. If the majority is corrupt then there is no solution.

Reply: Yes.

Alastair Beresford: So just coming back to Bruce's point earlier, so one of the things that bitcoin does is use, compute power for, it's sort of, for who gets to vote mechanism. I'm not sure I like that here, but it's something that does exist, at least as a market solution.

Springer

http://www.springer.com/978-3-319-12399-8

Security Protocols XXII
22nd International Workshop, Cambridge, UK, March
19-21, 2014, Revised Selected Papers
Christianson, B.; Malcolm, J.; Matyáš, V.; Švenda, P.; Stajano,
F.; Anderson, J. (Eds.)
2014, XI, 373 p. 40 illus., Softcover
ISBN: 978-3-319-12399-8