From: "Joe Baptista" <baptista@publicroot.org>
To: <DNSSEC@ntia.doc.gov>
Date: Mon, Nov 24, 2008 12:55 PM
Subject: Public Comments Regarding the Deployment of DNSSEC

Fiona Alexander, Associate Administrator,
Office of International Affairs,
National Telecommunications and Information Administration,
U.S. Department of Commerce,
1401 Constitution Avenue, N.W., Room 4701,
Washington, DC 20230.

I am making this public comment under the Notice of Inquiry published in the Federal Register on October 9, 2008. A confirmation of receipt would be appreciated.

I make these comments as an Internet user, root, and TLD (Top-Level Domain) operator. I am and have been involved in the launching and operation of commercial root and TLD services in the Netherlands and Turkey.

This year Dan Kaminsky, an Internet security researcher, found a bug that can be used by an attacker to poison DNS (Domain Name System) records. The publicity surrounding the discovery of the Kaminsky bug has been used improperly to create urgency in the promotion of DNSSEC (Domain Name System Security Extensions) deployment.

The NTIA (National Telecommunications and Information Administration) in a letter dated September 9 2008 to ICANN President Dr. Paul Twomey stated correctly that "deployment of DNSSEC at the root represents the most significant changes in the architecture of the DNS in the past decade." I would further add that deployment of DNSSEC will radically alter not only DNS but also Internet architecture as we know it today.

Who controls the Internet? (before and after DNSSEC):

Today the answer to this question is the end user. A user can elect to switch to any DNS provider he or she wishes to use. Today there are many root systems outside the IANA function (legacy DOC root). These system are operational in China, a number of Arab countries, Turkey has used a separate commercial system to replace the legacy DOC root, and Russia, India and other countries are planning their own national infrastructure deployment.

With DNSSEC fully deployed and operational the answer would change. The user would no longer have a choice. The U.S. Government would be in full control. Essentially DNSSEC is an underhanded smoke and mirrors game being played on the global public in the name of Internet security.

DNSSEC takes over control of the Internet through the deployment of a public key signature encryption system that uses the DNS hierarchy to establish trust anchors and points of authority. To operate properly it requires that the root be signed. Domain keys must be signed by the TLD key which is in turn signed by the root key. The root effectively controls the master key for all domains.

DNSSEC radically alters the end-to-end design principles widely used on the Internet today by creating a technical monopoly in the DNS under the control of the DOC (Department of Commerce), it's contractors ICANN (Internet Corporation for Assigned Name and Numbers), Verisign, and the thirteen root operators. I remind the DOC that many of these root operators have no contract with either your contractors or the DOC. They are volunteers.

I agree with the statement made to the DOC in prior submission that "it is simply not acceptable that any single government hold all the keys" to the Internet. To use a protocol as a means of creating a technical monopoly in the DNS via the root is a ghastly underhanded ploy by your contractors to take over control of the Internet.

I warn the DOC that to allow DNSSEC into the root is a bold move that will result in explosive commercial and political repercussions. DNSSEC if fully deployed creates a real technical dependence on the IANA function (the legacy DOC root) for the navigation of all user end points. Essentially a take over of the DNS and all navigational functions by the United States Government. Folks - this is not a good idea.

I remind the DOC that it is committed to preserving the stability and security of the DNS. To allow the creation of a technical monopoly at the root goes counter to the NTIA's commitment. I feel confident DNSSEC will not be deployed by the DOC and that it will not sanction the re-engineering of Internet Protocol to attempt a technical experiment to take over a core function of the Internet - the DNS.

I strongly advise the DOC get competent independent technical council to advise on the potential political implications of this technical monopoly being proposed under the guise of an Internet security enhancement. Governments would also be advised to fully investigate the legal, technical, commercial and political impact of U.S. Government control of national infrastructure.

Other Issues:

In addition to the main issues of control detailed above there are other considerations with respect to DNSSEC I will list here in point form.

- Under DNSSEC an administrator no longer just puts data into a database. Every change in the DNS must be signed by the domain owner. Any changes to the public key must in turn be signed by the registry key. The added administration required by DNSSEC will significantly increase the cost of operating domains. This will negatively impact business, industry and governments world wide.

- There will be an increase in Internet traffic. DNSSEC requires many more packets than the existing DNS. As a result users will experience slower response times when resolving domain names. Higher latency.

- Standard DNSSEC key signatures use 1024-bit encryption. 1024-bit encryption can be easily broken by script kiddies via bot nets or large organizations and governments having access to similar technology (computer resources). DNSSEC provides the world with a false sense of security.

- DNSSEC like current DNS traffic transmits information across the Internet in the clear. DNS packets are not encrypted and subject to MitM (Man in the Middle) attacks (sniffing).

There are better alternative solutions available in terms of addressing cache poisoning and similar attacks on the DNS that do not require the creation of a monopoly under U.S. Government control. To understand these solutions we must first understand the problem identified by the Kaminsky bug.

The Problem and the Current Solution:

It is important to stress that the DNS is not the issue here. DNS is an application program. Cache poisoning of the DNS is in this case a problem with the transport layer being the UDP (User Datagram Protocol). Kaminsky designed a number of attack vectors that use the UDP transport layer to trick the DNS into thinking it is getting a valid answer. DNS is not what is broken here. The problem is strictly an issue related to the UDP transport layer protocol used by DNS applications. The DNS bug discovered by Kaminsky is in fact a UDP bug.

UDP is one of the core protocols of the Internet Protocol Suite. Using UDP, application programs on networked computers can send short messages known as datagrams to one another. UDP does not guarantee reliability. Datagrams may arrive out of order, appear duplicated, or go missing without notice. UDP has no handshaking capabilities. This makes UDP faster and more efficient but also easy to trick.

Because UDP lacks any handshaking capabilities it uses the port and ID's fields in a UDP packet to track application-to-application communication. These fields are a 16 bit value and that makes it easy to attack since you only have to guess or predict a value between 0 and 65,535.

The Kaminsky bug is a specific problem associated only with recursive DNS applications. These are the DNS servers at Internet service providers or large corporations that look up information so users can resolve domain names by contacting other DNS servers that are authoritative for that information. This problem does not effect authoritative servers. So root and TLD servers are not vulnerable.

The Kaminsky bug is also not new. The UDP Cache poisoning issue has been a problem in the DNS for the last ten years. Amit Klein back in 2007 proved that the source UDP port and DNS transaction ID can be effectively predicted and described an algorithm to do just that.

But the current Kaminsky bug issue was first identified and solved by Professor D. J. Bernstein of the University of Illinois at Chicago in 1999. Bernstein was the first to propose and deploy modern cryptographic random-number generators to randomize the 16-bit ID and UDP source port numbers used by the DNS application layer. He was also the first to implement his solution in the DNS server application program called djbdns back in 1999. The Bernstein solution is what was used to fix the current UDP issues identified by Kaminsky and Klein.

It is not a perfect solution but it is the best solution available to us at this time. An attacker who makes a few billion random guesses is likely to succeed at least once. Russian researcher Evgeniy Polyakov managing a proof-of-concept cache pollution hack in 10-hours using equipment that bombarded a patched BIND DNS server with fake DNS requests using a GigE lan.

What Polyakov managed to prove is that in a controlled test with powerful systems on a fast network the Kaminsky attack-vector window still exists, but instead of the attack taking a few seconds to succeed it now takes about 10 hours. Kaminsky agrees the attack vector has not been eliminated, but only temporarily moved. The Bernstein solution however does make it much easier to protect the DNS using existing methods such as firewalls and IDS (intrusion detection systems) as an effective attack deterrent until a robust fix is developed.

Better Solutions (The Future):

A better solution that will solve the problem immediately is to simply use another protocol at the transport layer. The solution has been proposed on a number of occasions over the years to drop UDP as a transport and use TCP (Transmission Control Protocol) instead. The DNS already uses TCP as a transport for some transactions such as zone transfers.

TCP is a connection-oriented protocol, which means that upon communication it requires handshaking to set up an end-to-end connection. Unlike UDP, which assumes that the data received is from a trusted source if the port and ID numbers match, TCP is more reliable in that it ensures that the data received is in fact from the host system contacted and not an attacking server.

However TCP does require at least three packets just to set up a socket before any actual data is sent. It will take more time to resolve domain names using TCP and the data transmitted is not encrypted so it is subject to MitM attacks. Some consideration should be paid to data encryption over the TCP transport if that solution is implemented.

A much more elegant solution called DNSCurve was proposed recently by Bernstein. DNSCurve adds heavy-duty integrity and confidentiality to the DNS using the existing UDP transport. DNSCurve is easy for software authors to implement and administrators to deploy. Much like the TCP solution an administrator using DNSCurve does not need to change database software, does not need to store signatures, and does not need new procedures for updating DNS records.

DNSCurve uses a public key to encrypt and authenticate UDP DNS packets. No extra packets are generated and forged packets are very easily discarded and denial of service becomes much more difficult. It is essentially the best solution that ensures the long term stability and security of the DNS without the need for users to hand over control of their DNS to the U.S. Government.

In both cases the TCP and DNSCurve solution can be easily implemented allowing for backward compatibility with existing servers that only support standard DNS over UDP.

Regards
Joe Baptista
PublicRoot Consortium
www.publicroot.org
Tel: 416-912-6551