Choosing a DNSSEC Solution

Beware Dark Zones Ahead

Ron Aitchison
Zytrax, Inc.
Author Pro DNS and BIND

(Apress ISBN 1-59059-494-0)

May 29, 2009

1 Introduction

Without DNSSEC, we live in a world of hope. We hope that our web users get to our web site and not that of a hijacker. We hope that email sent to us actually gets to us, and it is not relayed to a malicious third party. We hope when using IM, FTP, VoIP, VPNs, and most other services that we are going to the right place, getting data from the right place, and sending data to the right place. DNSSEC—if correctly configured—replaces hope with certainty and provides a feature the Internet has long lacked.

As the progress—worldwide—of DNSSEC accelerates, domain owners are starting to look hard at its implications. First, they want to understand what all the fuss is about—just what is DNSSEC and what does it do for me? Second, domain owners want to understand the options--and consequences--of DNSSEC implementation.

In some cases, the need for DNSSEC is driven by compliance (especially U.S.gov domains, which must be DNSSEC ready by the end of 2009). The need for DNSSEC can also be driven by a desire to increase domain security (particularly when using SSL certificates), by questions from auditors or security experts, or even from that essential human characteristic—insatiable curiosity.

This paper does not make product recommendations but rather focuses on a set of criteria by which domain owners and operators can evaluate the already impressive range of DNSSEC solutions.

The slightly whimsical title of this paper also contains a health warning: DNSSEC is not a trivial technology. One of the consequences of a poor implementation choice is that a domain may become unreachable—in the DNS jargon, the zone may go dark.

This paper mentions the word cryptography. For those already suffering palpitations at the mere sight of this word, have no fear. This paper is an entirely math-free zone. Knowledgeable or impatient readers may want to skip the (mercifully) brief overviews and background material and cut to the chase in section 5.

2 DNSSEC Overview

The term DNSSEC is, like many others in the technical world, context sensitive. Strictly speaking, DNSSEC is a generic term that describes DNS security and currently covers three functions:

- **Zone Transfer Security**. Securing—and authorizing—of zone transfers from the master to the slave. This technology has been available for many years, and it is mature and widely implemented. Zone transfer security is not discussed further in this paper, but any DNNSEC implications are noted where appropriate.
- **Dynamic DNS (DDNS) Security**. Securing—and authorizing—dynamic (run-time) updates to zone files. Again, this technology has been available for many years, and it is mature and widely implemented. There are additional and significant implications for users of DDNS when using DNSSEC, which are noted throughout this paper.

- **Zone Integrity**. The term DNSSEC is increasingly being used to describe only this aspect of DNS security. This paper follows this trend; DNSSEC in the context of this paper means zone integrity.

DNSSEC has been available in its current form since approximately 2003. It was formally standardized by the IETF in 2005, with one incremental capability (NSEC3) introduced in early 2008. In this author's opinion, the technology is both stable and mature. There remain operational issues with details that are still being ironed out, but these currently do not have any implications for the underlying technology. Indeed, Sweden—the first operational DNSSEC country in the world—went live with the technology as early as 2005, and it has since been joined by a number of other countries. Most of the major DNS operators and registries have now committed to implementation timescales, the latest of which is scheduled for 2011. The U.S. government is noticeably pro-active in DNSSEC, with initial plans requiring all federal agencies' external zones to be signed by the end of 2009.

3 What is all the DNSSEC Fuss About?

When a PC user or a peer server connected to the Internet wants to access a service or a resource such as a web site, send an email, or create a VoIP call, it starts with the name of the resource or service that it needs to contact, for example, google.com. But networks need IP addresses to function, not names. So the first operation of almost every Internet access is a DNS lookup (a query, in the DNS jargon) to obtain the IP address that corresponds to the name.

The lookup process can be quite involved and complex. Hopefully, the result is the correct IP address from the domain's **authoritative** DNS server or an intermediate caching DNS server. A simplified version of this process is illustrated below.

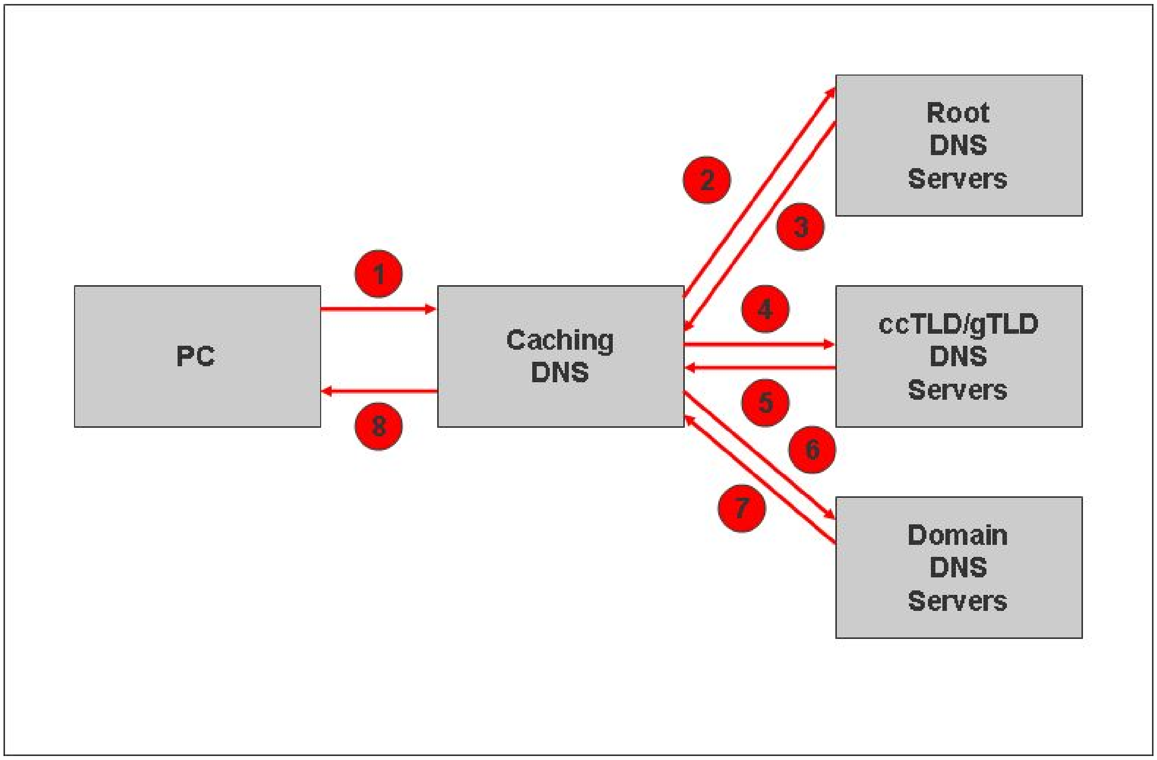

Figure 1 – Simplified DNS Query and Response Process

3.1 There is always Bad News

Every line or system in Figure 1 can be subverted. And without DNSSEC, we have no way of knowing it was subverted. Actually, that last statement is incorrect. When cash is whipped out of our bank account then, for sure, we know something probably went pretty seriously wrong.

But HTTPS solves the problem? SSL certificates solve the problem?

Wrong. Recall that the DNS lookup is the first transaction. If it is corrupted, hijacked, or polluted in some way, then we never get to the real site. All the HTTPS and SSL certificates in the world are simply useless if we don't go to the right place. DNS results really are that important.

3.2 DNSSEC is the Good News

DNSSEC (zone integrity) allows us to be certain we reach the right place—we get the right IP address. It does this by providing three key services:

- **Source Authentication**. We can prove that the DNS data about a domain can only have originated from the domain's authoritative DNS servers.
- **Data Integrity**. We can prove that the data received was the same as the data sent from the domain's authoritative DNS servers, even if it is subsequently obtained from an intermediate DNS caching service.
- **Proof of Non-Existence (PNE)**. In cases where a negative response is obtained (the name does not exist), we can prove that this is correct and that it came from the domain's authoritative DNS servers. The reason for this functionality may not appear all that obvious. But depending on an attacker's motivation, it may be enough to spoof responses indicating, for example, that your web site does not exist. DNSSEC stops this kind of attack through its quaintly named Proof of Non-Existence (PNE) capability.

3.3 How Does This Magic Work?

DNSSEC works using a cryptographic process coupled with (like SSL certificates) a trust process. If your eyes glaze over at even the sight of the word cryptography, fear not. As previously promised, this is a math-free zone.

The Signing Process

A DNSSEC-secured zone is digitally **signed** using one or more private key(s) of an asymmetric cryptographic algorithm. Asymmetric encryption systems, such as RSA, elliptic curves, and others, have both public keys and private keys. The public key (as its name suggests) can be made freely available. It can only be used to decrypt information signed by the private key, thus authenticating the source of the data.

A number of new DNS resource records (RRs) are created during the signing process. The public key(s) used to sign the domain are stored in the form of DNSKEY RRs. Each group of DNS records, including the DNSKEY RRs, is signed using one or more of the private keys, and the results are placed in an RRSIG RR. Finally, an NSEC (or NSEC3) RR is added between most RRsets, which are used to provide Proof of Non-Existence (PNE).

The Trust Process

The trust part of the process enables the receiver of the DNS records to track back the public keys until it reaches what is called a trust anchor. As its name suggests, a trust anchor is simply the public key of a domain in the DNS hierarchy which was obtained from a source and using a method (not defined in the DNSSEC standards) that the operator of the DNS trusts. For example, in the case of Sweden, the trust anchor to cover the entire Swedish domain (.se) can be obtained directly from an HTTPS page at IANA (part of ICANN, the DNS governing body).

The tracking-back process may involve navigating a chain of trust, in which the parent zone (the next one up in the name hierarchy) contains yet another new DNSSEC RR called the DS (Delegated Signer) RR. The DS RR corresponds to, and validates, a particular public DNS key. It is typically supplied by the domain owner, though it can be computed by the parent. When used in this type of chain, the parent zone must also be signed. The receiving DNS that is verifying the authenticity of the zone data will have the trust anchor corresponding to the zone name at some point in this chain.

Figure 2 illustrates the DNSSEC process and chain of trust.

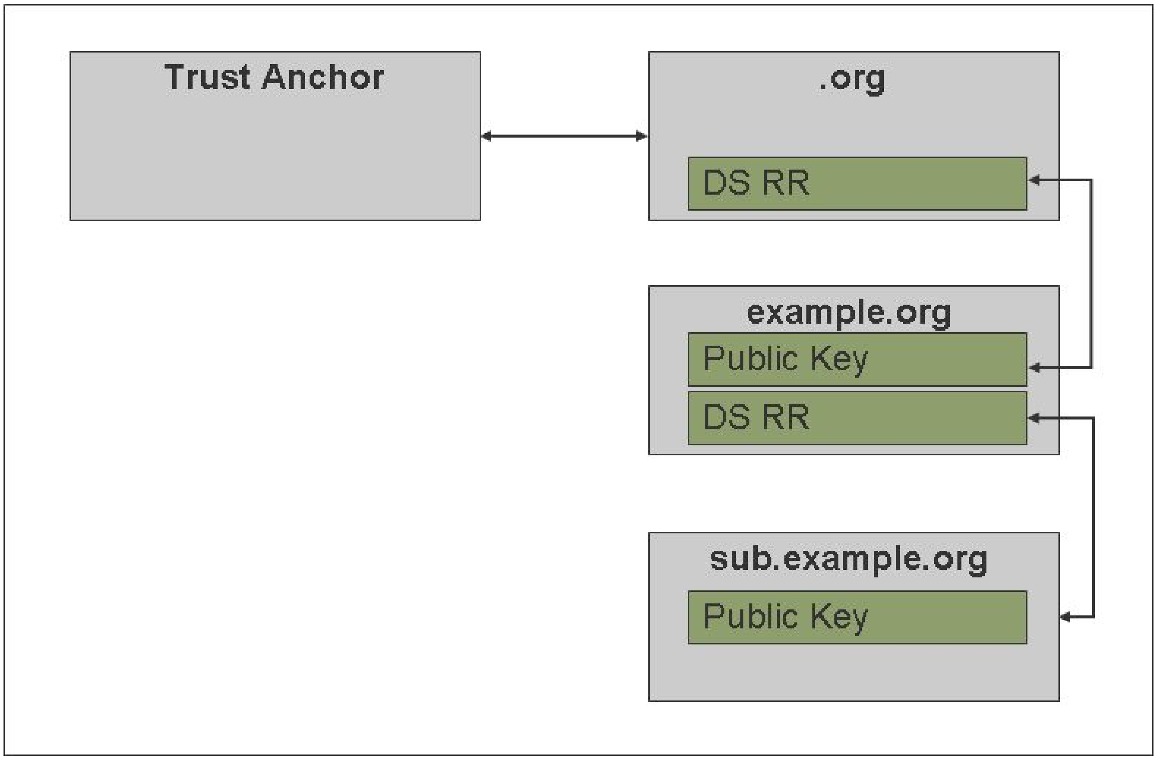

Figure 2 – Chain of Trust

While the specific example of the .org domain is shown in this Figure for illustration purposes (and because it was one of the first gTLDs to have committed to an aggressive implementation schedule), it could equally apply to any other domain, such as .gov, .se, or .br.

4 DNSSEC Complexity

The previous section presented a very simplistic overview of DNSSEC. To give a feel for the underlying complexity of the technology, which is essential when evaluating possible solutions, this section lists and briefly describes some of the procedural issues that are required to make DNSSEC viable.

- **Zone Changes**. Every time the zone changes, it needs to be re-signed. When the zone is re-signed, the private key(s) are required to be on-line. If Dynamic DNS (DDNS) is being used, either re-signing has to be batched, say every 24 hours (which may well defeat the objective of using DDNS), or the private key(s) used to sign the zone must be permanently on-line.
- **Periodic Re-Signing**. The zone's s signatures have time limits. Even if no changes have taken place to the zone's data, the signatures must be updated by re-signing the zone before they expire. Re-signing requires the private keys to be on-line.
- **Multiple Keys**. While it is possible to use DNSSEC with a single operational key, in practice, most systems use at least two keys. Each key has a unique role in the process and requires regular maintenance. The maintenance process for each key may be different.
- **Key Maintenance**. (Also known as key rollover.) The key(s) used to sign the zone should be replaced periodically for safety reasons. Because of the nature of the global DNS system and particularly the presence of intermediate DNS caches, key rollover involves the introduction of new keys some time before they are used to sign the zone. Old keys are retired some time after they have ceased to be used. At any one time, a zone may have a replacement key, a current key, and a retired key.
- **Chain of Trust**. When keys that are visible outside the domain are changed, these changes need to be propagated to the chain of trust—the parent, the receiving DNS systems, or some trusted intermediary.

  Now add multiple domains to the mix, and the fact that, if a key is compromised, some of these operations need to be carried out in pretty short order. It becomes quickly apparent that some form of DNSSEC automation is highly desirable, if not absolutely essential, to avoid a coronary bypass.

  DNSSEC relies on both a procedural process and a trust process. As far as the domain signer is concerned, the trust process means the care and attention devoted to handling of the private key. If one or more of the private keys used to sign a domain are compromised (obtained by a malicious user), the last thing attackers will do is boast about the fact. Instead, the bad guys will simply subvert the domain traffic or use the key to devastating effect. Among the last to know the key was compromised may be the signing organization.

  Secure management of the private keys is crucial to a successful implementation of DNSSEC. Secure key management has many parts, but arguably the single most important aspect is minimal—or better yet—zero exposure.
- **Hidden Master**. Private keys should never be exposed on a public server. Instead, a hidden master or a special DNS zone signer should be used and the resulting signed zones transferred to public DNS slave servers.
- **On-line and off-line keys**. Even in a hidden master configuration, extreme caution must be exercised. Leaving a private key permanently on-line in a normal file system is a ticking bomb. Instead, the private key should only be present when it is required for the signing process, and it should be taken off-line immediately afterward. This simple rule has two major implications for DNSSEC implementation. First, it means that whatever level of DNSSEC

automation is implemented, if the net result is to receive an email that says, “meet me in the machine room in 10 minutes and don’t forget to bring the private key,” then such a system has serious operational limitations. Second, when DNSSEC is used in conjunction with Dynamic DNS, then the problem becomes more acute because the zone must be re-signed, requiring the private key, whenever a change is made. Unless changes are batched in some way, the private key may have to be permanently on-line. The only truly safe way of handling private keys in such an environment is using some form of tamper-proof or tamper-evident hardware module in which private keys are never exposed.

Figure 3 illustrates a hidden master configuration and shows how it may fit into an enterprise environment.

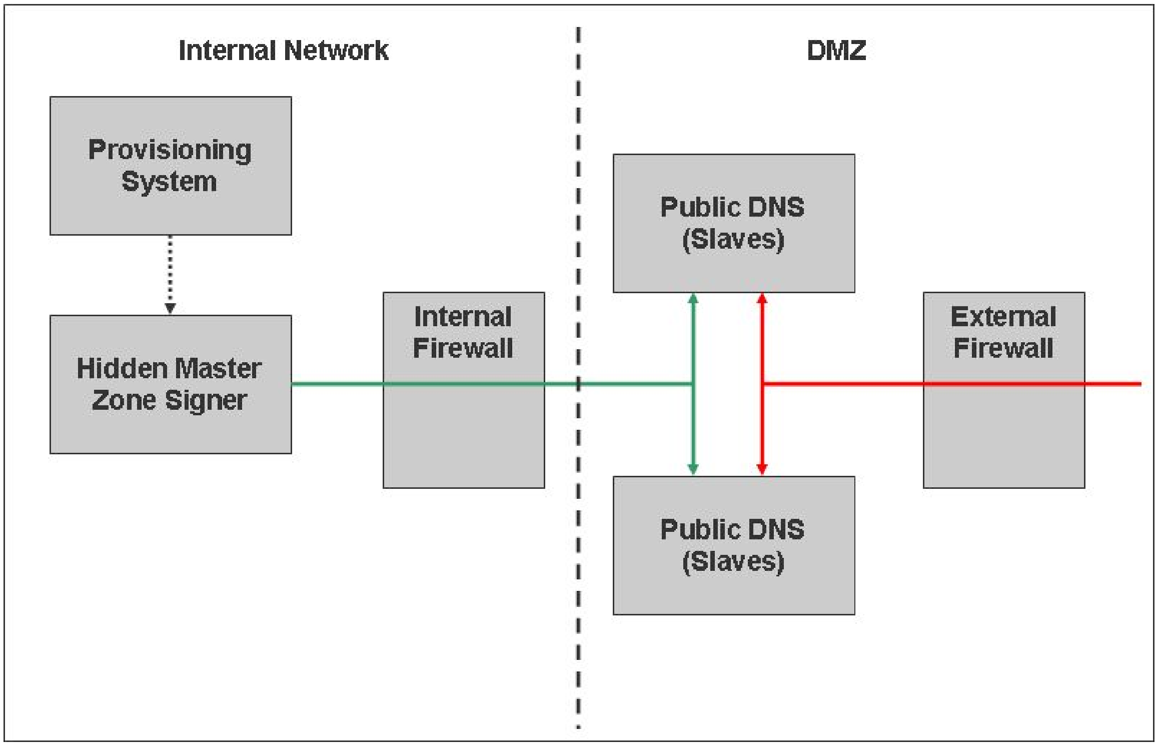

Figure 3 – Hidden Master Configuration

5 Choosing a DNSSEC Solution

There are, already, an impressive number of possible solutions for the implementation of DNSSEC provided by both Open Source and commercial organizations. This paper categorizes possible DNSSEC solutions into only two categories, each of which has a high or low rating. The solution attributes in each of the four quadrants of the resulting matrix are briefly described, together with identifying characteristics to help the reader navigate the confusing minefield while still retaining (hopefully) at least one limb.

5.1 Applicable Standards

Suppliers of all shades—commercial and Open Source—can use the terms “DNSSEC compliant”, “DNSSEC Ready” and “‘DNSSEC enabled” rather loosely. The following is a brief checklist of RFCs and other standards that may apply to DNSSEC zone signing solutions, which should be verified as supported based on implementation requirements:

- **RFCs 4033, 4034, and 4035**. Mandatory. These RFCs define the basic functionality of DNSSEC and must be supported to be DNSSEC compliant.
- **RFC 5155**. Optional. The base DNSSEC system allows simple zone enumeration (find all the names in a zone file) by following the NSEC chain used for PNE (Proof of Non-Existence). This RFC provides an alternative PNE method using a new NSEC3 RR type. If the possibility of zone enumeration is to be minimized (it is always possible), then support of this RFC is essential.
- **RFC 5011**. Mandatory. Defines a method of supporting trust anchor updating using the REVOKE bit set in the DNSKEY RR.
- **SP800-81**. This publication from the U.S. National Institute for Standards and Technology (NIST, www.nist.gov) contains specific recommendations for domain name security, including DNSSEC, that are required for U.S. federal government organizations. However, it contains many details that have much wider applicability.
- **FIPS 140-2 Level 3 or Level 4**. U.S. government standard covering cryptographic modules. The FIPS 140-2 series of standards are recognized in Canada, Europe, and Australia.
- **ISO/IEC 19790:2006 Level 3 or Level 4**. The ISO equivalent of FIPS 140-2 Level 3 and Level 4.

Other compliance standards are outside the scope of this paper, but they should reference one or more of the above RFC standards when defining required functionality.

5.2 Choice Matrix

User requirements can vary enormously, and DNSSEC system capabilities vary enormously. Picking a path through this minefield is a question of balancing many competing demands. It is all too easy to get sidetracked into a ‘Whizzo feature X’ versus ‘Whizzo feature Y’ war.

This paper is not about specific product recommendations but rather evaluation criteria. The following choice matrix is designed to get to the essence of DNSSEC implementation—everything else is simply fluff—perhaps useful fluff! But still fluff.

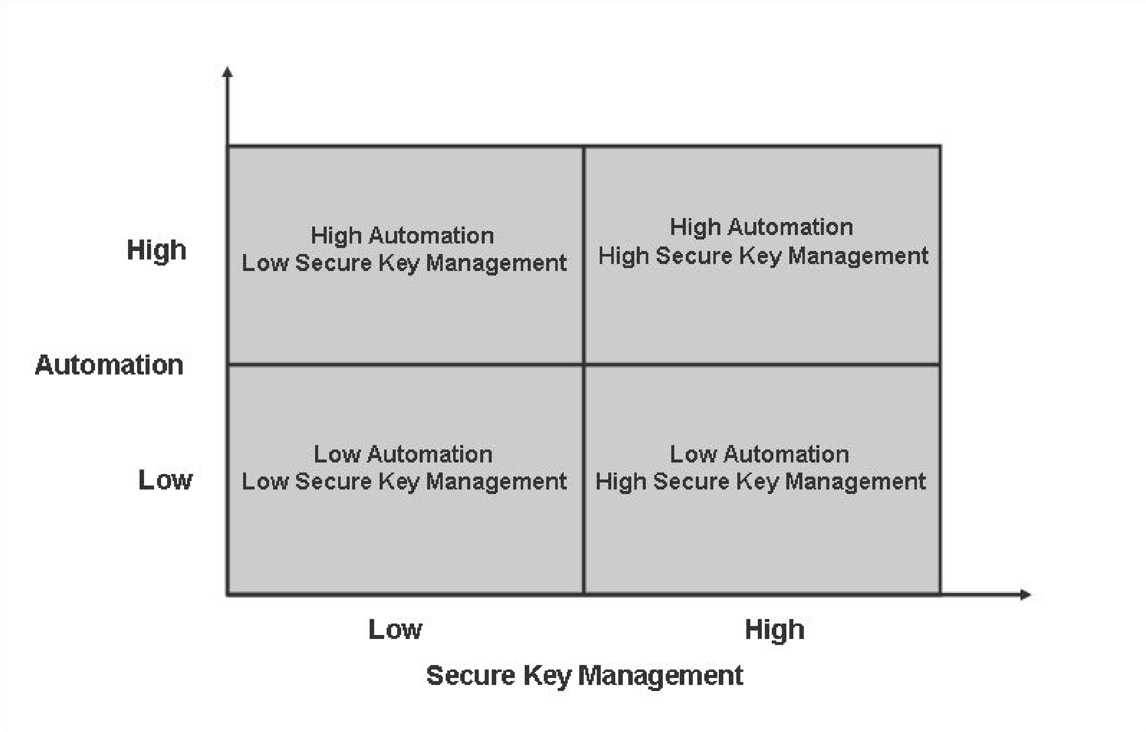

Figure 4 – The DNSSEC Choice Matrix

The following sections discuss system attributes under each of the quadrants. While on its face it may appear obvious that the ideal solution would lie in the quadrant High Automation, High Key Management, that may not always be practical. In which case, the matrix also indicates missing capabilities that may need alternative provisioning.

5.3 Secure Key Management

As noted previously, DNSSEC is about building and maintaining trust relationships. Ultimately, that boils down to security of the private keys. Remote key transactions always use public keys. All the superb cryptographic technology in the world is useless if private keys are simply left lying around or needlessly exposed. This is not a simple problem to solve, especially if Dynamic DNS (DDNS) is being used.

Low Secure Key Management Attributes

Solutions that have any **one** of the following attributes are classified as Low Secure Key Management:

- **Manual Key Handing**. The simple fact of life is that we poor humans make mistakes. If we are required to manually take the private key off-line, and then put it on-line, we are going to make a mistake. The more frequently we have to do it, the more likely the mistake.
- **Uncontrolled Permissions**. Private keys must always have the minimum read permissions to enable them to perform only their required tasks. The software that generates keys must automatically assign the correct permission levels such that the resulting private keys can only be accessed by corresponding applications in the solution, for example, a zone signer. In most systems, root/Administrator can read everything. Typically, the parts of a DNSSEC automation solution that handle private keys typically run at these user/group

levels, thus requiring root/Administrator-only read permissions on all private keys.

- **Visible Zone Signer**. DNSSEC automation solutions that cannot operate in a hidden master environment (and be capable of securely transferring the signed zones to the public servers) are a disaster waiting to happen—and not waiting very long either.

Note: DNS systems which simply **serve** signed zones and do not sign the zones should never need to handle private keys and therefore do not require any special treatment.

High Secure Key Management Attributes

High secure key management is generally only possible using hardware assistance—especially if DDNS is being used and private keys are required to be permanently on-line. Where this is neither feasible nor cost-effective, other mitigating strategies are discussed.

- **Hardware Assisted Key Handling**. High secure key management solutions will use hardware in the form of a tamper-proof or tamper-evident (you are told when it has been compromised) cryptographic module that should be certified to FIPS 140-2 (Level 3 or Level 4) or ISO/IEC 19790:2006 (Level 3 or Level 4). Cryptographic modules come in a variety of forms such as network appliances, plug-in cards, or embedded in systems. All key generation and key processing takes place inside these modules, and private keys are never visible. Period.
- **Secure OS Platform**. It does little good to protect the keys in a cryptographic module if an attacker can install malware on the server and modify your DNS data before it is signed. High secure key management solutions must use a hardened or secure OS platform to host the DNSSEC automation application.
- **Hidden Zone Signer**. Even with the best key management in the world, if your system is publicly exposed—you become like a honey pot. The bad bees of the Internet will swarm around until they get smart or lucky. A hidden master configuration at least minimizes this exposure. Of course, modern security research indicates that there are many internal threats as well. But it is best to at least minimize the fronts you fight on, to brutally paraphrase Von Clauswich, or was it Sun Tsu?

Manual Key Handing Mitigation Strategies

If the only option is to use a non-hardware assisted key management system, then there are couple of strategies that can allow such DNSSEC automation solutions to get to a high'ish (but never a high) secure key management level.

That part of a DNSSEC automation solution suite that handles key generation and signing should be synchronized to use the same user/group permissions. Software that generates keys automatically assigns the required minimum permissions. Software that uses private keys always checks that a key has the right (minimum) permissions, automatically sets them if this is not the case, and issues an alert to inform the organization that they have a weak spot in their key handling processes. Finally, once the key has been used, the software issues an alert to indicate the keys are no longer required and performs a periodic check to make sure it (they) have been removed. Without these minimum safeguards, you may as well start leaving your front door open at home as well.

The important point to consider here is file system accessibility, not the solution packaging. For example, DNS appliances bring multiple benefits by isolating functionality and providing tailored solutions. However, such appliances are normally implemented using Linux or one of the BSD variants, many with unique hardening solutions, and thus have conventional file systems. If the good guys can get into the appliance, then the bad guys can get in as well. If the good guys cannot get into the appliance, the chances are high—but not impossible—that the bad guys will not be able to get in.

The difference between hardware assisted key management and manual key handling apply equally to DNS appliance and non-appliance solutions.

Note: A number of vendors offer thumbprint or other authentication methods to secure access to USB devices. In some case, such devices are even certified to the lower FIPS 140-2 Level 1 or 2 standard. In most cases however, once the authentication process has occurred, the USB file system is freely exposed to the OS. These devices have the characteristics of manual key handing, not hardware assisted key handling.

5.4 DNSSEC Automation

DNSSEC involves a number of regular processes and some pretty tight scheduling issues, both of which suggest that the more automation, the less likelihood of something going wrong. And that is the normal case. If things go wrong—such as key compromise—those things need to happen in a hurry. We have all been through enough fire drills to want to avoid them if possible.

Low DNSSEC Automation

Solutions characterized as having low automation have the following characteristics:

- **Extensive User Intervention**. Many of the DNSSEC processes are purely procedural. Periodic re-signing and key rollovers (ZSK) that do not affect external parties are obvious examples. To even qualify as low DNSSEC automation, software must provide these services as a minimum. Processes that require some form of manual intervention (at least at this stage in DNSSEC evolution) are where the low-high differences start to appear. I would argue that even low automation software should have a model of the DNSSEC process and at least be capable of informing the user what they must do via email, paging alerts, or some other sensible method. Even a “meet me in the machine at 12 sharp, bring the private keys, and have a rose between your teeth so you can be identified” email is better than a dark zone.
- **Lack of Granularity**. There are elements of a DNSSEC automation solution that are required to handle private keys, for example key generation and zone signing, that consequently require appropriate key access permissions. There are many elements of a DNSSEC automation solution that do not require access to private keys, and these should run at the lowest possible system permissions. Thus, in the (hopefully rare) event of a software error in such elements, access permissions which could expose private keys are not compromised.
- **Require Extensive DNSSEC Knowledge**. Low automation solutions will generally require that you have reasonable to high knowledge of the DNSSEC process. You will need this knowledge because if anything goes wrong, you will most likely have to get involved in pretty short order. The cost of DNSSEC knowledge acquisition should not be under estimated.

- **Emphasis on GUI**. There is a prevailing view that a GUI indicates a high level of automation and that conversely the command line is some artifact from computing's Stone Age. While the best GUI's can greatly simplify and decrease the likelihood of configuration errors, many low automation solutions (the word most also springs to mind) simply provide a GUI wrapper to a manual process. For example, a GUI with a prompt of "Please enter public key" is not exactly a game changer. This is also why the command-line guys always retort by saying, "You GUI guys are lucky, you get to type stuff and click the mouse, we just get to type stuff." In this author's opinion, better a highly automated command line solution than a poorly automated GUI solution.

High DNSSEC Automation

Solutions characterized as having high automation have the following characteristics:

- **Little User Intervention**. High DNSSEC automation means that, by default, current best practices are implemented without providing endless configuration data or baby-sitting the process. Equally, where the user is required to intervene, it should not require a three-week vendor course. There are still parts of the DNSSEC process that do require manual intervention—these are always concerned with external communications. Apart from making keys available and importantly removing them at certain times (depending on the secure key management strategy), the user should not be required to ever run or execute a specific job or even intervene manually at a GUI or command line to perform a particular task—in the normal course of events.

  However, even in the most highly automated systems, the user should certainly expect to have to carry out manual processes, such as sending DS records or trust anchors by a secure process, such as secure email, to notify an externally visible key-rollover. Where a manual process is needed, the best systems will provide all the material required (in some cases in a number of formats) with explicit instructions or advice as to the next step in the process. Explicit confirmation of the action should be required, with appropriate escalation if not received within a reasonable timeframe.

  As an example, if a new external key (a KSK in the jargon) is made available, any signed parent or key repository must be updated. The best solutions, having provided the user with the means and data by which this can be done, should then check periodically that it has been done, by automated inspection of the parent or key repository. This is simple but essential stuff that can be accomplished now. Clearly, there are always emergency cases—such as key exposure—that will require, perhaps a lot of, manual intervention. Here again, a high level DNSSEC automation solution should at least have a model of best practice recovery procedures.

- **Require Moderate DNSSEC Knowledge**. While good quality automation solutions can do a lot and in the normal case should (depending on the secure key management strategy) require little if any intervention, the downside of an emergency is a dark zone that may be cataclysmic. At least reasonable knowledge of the entirety of DNSSEC and its attendant processes will be required for some time. Whether this knowledge is maintained in-house or through external experts is a matter or local policy and budget.

- **Failover Architecture**. Hardware and networks can fail, and if they are down long enough, signatures can expire and zones can go dark. High automation solutions should offer the ability for a backup zone signer to take over from the primary signer in such cases. This has serious implications for the secure exchange of keys between such systems, or if keys are not transferred (the most secure option) then immediate zone resigning with a new set of keys is essential.

6 Recommendations

This paper is not about explicit product recommendations but rather objective evaluation criteria. However, for what they are worth, here are some more subjective considerations and observations.

Many DNS appliance, provisioning system, and even OS suppliers are adding DNSSEC to their product lines. While the DNSSEC technology is mature, the operational practices are still evolving. What to send to who, in what format, when, etc., are all still open questions. Vendors need to keep track of what is still a moving target and be committed to keep abreast of developments. DNSSEC cannot be an afterthought or buzzword to flush out a product line. The consequence of failure could be a dark zone—you need a committed DNSSEC vendor.

If you are forced to make a trade-off within the choice matrix, in this author's opinion, good secure key management trumps good automation every time. You can always learn more about DNSSEC or carry out some manual process, but you might not even know about stolen or compromised keys—until it is too late. Better the known than the threat of the unknown.

6.1 Questions to Ask Your DNSSEC Vendor

Hopefully, the criteria offered in section 5 have already provided lots of vendor questions and areas for detailed follow-up; however, here are a few explicit questions you might want to ask your potential DNSSEC vendors in order to determine which type of solution they offer:

- Are private keys stored on-line?
- Is the zone signer visible to the Internet?
- Are permissions on the keys and the zones set to the minimum possible level?
- Is the solution FIPS 140-2 or ISO/IEC 19790:2006 certified? To what level (1-4)?
- Does the system automatically sign and re-sign the zones without any user intervention required?
- Does the system automatically perform key rollovers without any user intervention required?
- What does the system do when an externally visible key (a KSK) is rolled?
- If zone enumeration (zone walking) is important—does the vendor support NSEC3?
- Does the vendor have a fail-over strategy? If so, does it involve secure key transfer or key rollover?

7 Summary

DNSSEC is a mature technology that can be implemented now with little if any risk. Substantial benefits can be derived from the technology immediately, especially where verified access is required for either intra- or inter- organizational communications—be they government, supplier networks, or any other community of interest or affinity grouping. With the pace of DNSSEC implementation accelerating, as it currently appears to be doing, the benefits will only increase. You may also, as an additional benefit, keep auditors, unhappy users, and regulatory authorities off your back.

Finally, the integrity of the domain within a DNSSEC environment is determined by the care with which an organization handles its private keys. While pre-DNSSEC secure DNS solutions (for zone updates and zone transfers) have always demanded some form of secure key management, these were usually not extreme and the consequences of failure were at worst limited to a small and finite number of users.

DNSSEC places a premium requirement on secure key management. The importance of this point cannot be over-emphasized. The actual process of signing and maintaining DNSSEC signed zones, while complicated and containing some nasty timing criteria, is ultimately procedural. The key (pun intended) requirement of any DNSSEC software automation solution is how it manages private keys. There is almost no alternative, especially if Dynamic DNS (DDNS) is being used, other than to use key management hardware—a tamper-proof or tamper-evident cryptographic module—as part of a total solution. Less than this exposes the user to significant down-side risks, which at best negate all of the substantial advantages DNSSEC offers, and at worst could result in an unreachable—a dark—zone.