CHAPTER

Issues in DNS security

2

INFORMATION IN THIS CHAPTER

- A Brief History of DNS Security Breaches
- Why is DNS Security Important?
- Common DNS Security Problems
- Developing a DNS Security Plan

INTRODUCTION

DNS is a core component of everyone's daily lives on the Internet, but very few people understand how it works, or how fragile the underlying infrastructure can be. Even security professionals, who are charged with protecting an organization, often do not have a full grasp of the potential security pitfalls in DNS.

Part of this lack of knowledge stems from the fact that DNS is something that is often "set and forget." DNS infrastructure is set up and other than a few zone changes here and there it is rarely considered. DNS is also a long established protocol, many companies registered their domains 20 years or more earlier and the team that set up the original DNS infrastructure has long since moved on to other roles. As long as DNS for the organization is working why make changes? Even worse, there may not be anyone who knows how to make changes.

AN OLD PROBLEM

The "set and forget" DNS problem has been around for years. In the mid-1990s I worked for a major Internet Service Provider (ISP) that had significant turnover within the DNS team. The ISP domain name, which was also used to manage our backbone infrastructure, expired and no one knew. Fortunately, a manager at Verisign knew one of our managers and she called before letting the domain expire. The manager put the $100 renewal on his personal credit card because he knew he could not get an invoice paid in time to keep the domain live and prevent the ISP from effectively shutting down. Lesson one in DNS Security: Make sure domain renewal notices go to an alias, not an individual person.

The fact that DNS is so resilient, combined with domain registrations being done for years at a time and too few security teams that have DNS experience and too few

DNS Security. DOI: http://dx.doi.org/10.1016/B978-0-12-803306-7.00002-4

DNS administrators that have security experience, creates a unique challenge in securing DNS infrastructure. Combine the internal challenges with the external DNS security threats that face an organization: DNS-based Distributed Denial of Service (DDoS) attacks, cache poisoning, malware that uses DNS for command and control purposes, and DNS security is a potential nightmare for any team.

The goal of this chapter is to provide a quick history of the some of the best known attacks against or taking advantage of flaws in DNS. The chapter will also discuss some of the threats and how to put together a plan to better protect an organization from these threats.

A BRIEF HISTORY OF DNS SECURITY BREACHES

A listing of all security breaches that were either attacks against DNS infrastructure or took advantage of flaws in DNS security would fill several books. Rather the purpose of this section is to provide an overview of the different types of breaches that have occurred over the years and to demonstrate how DNS attacks have changed over time.

In 1996 Eugene Kashpureff used a DNS cache poisoning exploit to redirect traffic from the InterNIC's web site to his own web site, AlterNIC, an alternative registry. The exploit went on for several days before Kashpureff returned service to the InterNIC.

In February of 2000 an attacker changed the authoritative name servers listed with the InterNIC for RSA Security's domain. The attacker also set up a spoof RSA Security web site and directed users to that site—giving the mistaken impression that the web site had been compromised.

On January 29, 2001 access to all of Microsoft's sites, including its MSN sites, was disrupted for almost a day because of an attack launched against Microsoft's name servers. Microsoft's DNS administrators made the attack easier by placing all of their name servers on the same network segment, which gave the attacker a single target.

An attack was launched against the root name servers on October 21, 2002. The attack was an ICMP-based DDoS attack that rendered several of the root name servers unreachable. Because of recursiveness and the redundancy in the root servers virtually no one noticed the attack, which lasted about an hour. Had the attack continued for a longer period of time the impact would undoubtedly have been much greater.

In June of 2008 a Turkish hacker group calling itself NetDevilz used social engineering to convince the domain registrar for the Internet Corporation for Assigned Names and Numbers (ICANN) and the Internet Assigned Numbers Authority (IANA) to hand over control the icann.org and iana.org domains to NetDevilz. The record change only lasted 20 minutes or so before it was corrected, but many users were redirected to the wrong web sites for up to 24 hours.

Also in 2008 Dan Kaminsky released details about the "Kaminsky Bug" which would allow an attacker to send authoritative responses to domains for which the server was not authoritative. For example, a user could visit reallyfunwebgames.com and the authoritative name server for reallyfunwebgames.com would also send an authoritative response for americanexpress.com. Thus, every user who relied on the same recursive server as the original user would now be sent to the wrong page when they tried to go to americanexpress.com. Kaminsky was able to engineer this by combining a flaw in which DNS servers managed query IDs with a cache poisoning technique. Unlike other attacks on this list, Kaminsky is a responsible researcher and reported the bug to the appropriate vendors so it could be patched before he released details of the exploit to the general public.

In 2013 web hosting company, CyberBunker, launched what was, at the time the world's largest DDoS attack against the DNS servers of Spamhaus, a volunteer organization that tracks spammers and provides a blacklist other organizations can subscribe to reduce the amount of spam they received, because Spamhaus added CyberBunker's IP address space to the Spamhaus Black List.[1] Other organizations had attempted unsuccessful DDoS attacks against Spamhaus servers before. By targeting the Spamhaus DNS servers, which were hosted by a third party, CyberBunker was able to bypass the DDoS mitigation capabilities that Spamhaus had in place. Those outsourced DNS servers also served other customers around the world, so the DDoS attack not only made Spamhaus servers unreachable it also degraded service for customers around the world.

In 2010 Verisign was the victim of multiple successful attacks by unknown attackers. Verisign manages the .com and the .net root name servers as well as the root name servers for several other Generic Top Level Domains and many Country Code Top Level Domains (ccTLDs). According to Verisign, no data related to the root servers managed by the company was compromised during the attacks.

On March 31, 2012 Anonymous attempted to take the entire Internet off-line with Operation Blackout. The goal with Operation Blackout was to take out the 13 root servers using a DNS Amplification attack (described later in this chapter). DNS Amplification attacks are remarkably easy to launch and have been used effective in a number of DDoS attacks. Fortunately, the crew at Anonymous had very little understanding of the how DNS operates, how the root name servers are configured, how major ISPs deal with the root name servers. Not to mention that they are, for the most part, incompetent. The attack had very little chance of success. In the end the attack either did not happen or simply had no effect on the performance of the root name servers.

A much more effective attack was lunched against Turkey in December of 2015. This DDoS attack was targeted the .tr ccTLD root name servers and effectively isolated Turkey from the rest of the world, Internet-wise. The attack had the side benefit of degrading service throughout Europe because the Reseaux IP Europeens Network Coordination Centre provided secondary authoritative DNS services to the .tr domain.

By attacking the .tr root name servers with a relatively modest 40 Gps DDoS attack the attackers were able to make about 400,000 domains unreachable. Which meant users were not able to reach company web sites or send email to users with the .tr email addresses. In order to block the attack the Turkish government had to temporarily block all Internet traffic originating from outside of Turkey. That allowed people within Turkey to start communicating with .tr domains again, but blocked the rest of the Internet.

These attacks all serve to illustrate a number of points: The first is that in some cases, the attacks may have been prevented if a stronger DNS security policy had been in place. The second point, and in some ways the more important of the two, is that large companies, security savvy companies, and even companies with extensive DNS experience can still be vulnerable to attacks. A third point to note from these examples is the evolution of attacks over time. Were this book written about DNS security 10 years ago, it would have primarily focused on DNS cache poisoning, DNS hijacking, and vulnerabilities in DNS software. Instead, there will be a lot more focus on the protocol itself, and how to take advantage of weaknesses in the protocol to make service unavailable or exfiltrate stolen data.

WHY IS DNS SECURITY IMPORTANT?

Ask any security professional what keeps her awake at night and you will most likely get a response about protecting the organization against phishing attacks.[2] Dive a little deeper and she might express concerns about security challenges with BYOD (bring your own device) or worry over some of the web applications that network users have access to, or that run on the organization's web site. After a few beers she might express concern about the fact that there are more alerts than she can keep up with, or that she does not have a clear picture of everything that is happening on the network.

It is very rare that a discussion about security issues reaches the point where DNS comes up as a topic. That seems like an odd statement to make in a book about DNS security, but it tends to be true. Unless there has been a recent breach in the news involving DNS, generally DNS does not come up as a topic.

DNS is also one of the most outsourced services. Many organizations recognize that they do not have DNS expertise in-house so let their domain registrar or another third party manage the organization's zones and only run recursive DNS services internally (though, often even that is outsourced to the ISP providing connectivity to the organization). With little or no control of the DNS infrastructure residing within the organization it is easy to see how DNS can become an afterthought in security plans.

But, DNS security needs to be at the forefront of every discussion about network security. DNS attacks are more common than most people realize and

failures in DNS security can be crippling to an organization. How much money does an organization lose every hour that it is unreachable via email? How about when a fully functioning web site is invisible to the Internet, or worse visitors to a web site are redirected to a malicious web site? A 2014 study done by Vanson Bourne found that 75% of organizations in the United States and the United Kingdom had been impacted by a DNS attack and 49% had uncovered some sort of DNS-based attack in the previous 12 months. So, DNS attacks are prevalent, but they are not necessarily getting the attention they deserve.

DNS falls into a category of "utility protocols" that underpin communications on the Internet. These are robust protocols that help keep traffic flowing and servers talking and that most users do not know exist. Protocols like the Border Gateway Protocol, Network Time Protocol, and of course DNS are critical to keeping the Internet up and running, but generally fall well outside the purview of security teams. The administrators who do configure and manage the systems that run these protocols do not usually think about the security concerns inherent in these protocols.

This lack of security insight combined with the relative obscurity of these protocols makes them ripe for potential exploitation and hackers have figured that out. The result of this perceived utility is that within the black hat community there has been a sizeable increase in exploitation and vulnerability research in these protocols. There has also been a lot of research done by the security community into ways to better protect these protocols. Unfortunately, there is a big gap between the work done by researchers and the people who handle the day-to-day administration of these protocols.

A prime example of this is with DNSSEC (discussed in detail in Chapter 10). RFC 3833, which introduced a way to better secure DNS infrastructure, was first released in 2004. Even in 2016 very few domain names have added DNSSEC signing to their zone file and many domain registrars still do not support it.

In the end DNS security is important because a failure in DNS can render an organization completely unreachable and because attackers are actively looking for new ways to exploit the DNS protocol and the DNS infrastructure itself. Understanding key issues in DNS security is critical to maintaining a strong security posture within an organization.

COMMON DNS SECURITY PROBLEMS

Before a security team can effectively protect an organization's DNS infrastructure they must first determine what the risks to its DNS infrastructure are. When performing a risk assessment of a DNS infrastructure it is important to take a very broad view of what constitutes a security risk. The goal of a DNS security plan is to make sure the DNS infrastructure is available as much as possible and that the proper information is propagated to machines making queries.

Based on the definition above, anything that impacts availability or causes faulty data to be disseminated could be considered a security breach. Some would consider this definition problematic because it expands the definition of security beyond its traditional meaning. However, given the importance of DNS to an organization an expanded definition of security is reasonable and, arguably, essential.

One of the reasons an expanded definition of DNS security is essential is that there are so many points of security failure within a DNS framework. In addition to failures traditionally associated with data security such as hardware failure, unauthorized server access, and DDoS attacks, there are also registrar administrative issues, sleazy marketing, and other types of security breaches unique to DNS. The distributed nature of DNS automatically requires a different set of security concerns and adds a layer of complexity to security plans.

Here is an all-too-common example of the unique problems facing anyone attempting to secure a DNS infrastructure: It is Monday, everyone stumbles into the office and realizes that they cannot check mail, the corporate web site is also unreachable. Internet connectivity is fine, and people are able to send mail and access other web sites. The DNS administrator is asked (usually frantically) to fix the DNS problem. But the DNS servers are working fine. Both the primary and secondary servers are responding as expected, data has not been changed and there is no sign of unauthorized access.

The DNS administrator spends all morning attempting to determine the problem. She checks and rechecks system settings, verifies that DNS information has not been altered with the registrar searches various DNS web sites all to no avail. Finally, she posts a description of the problem to a DNS-related mailing list. Within a few minutes someone replies with output of whois data and points out that the domain name has expired. Shaking her head in disbelief the administrator contacts the accounting department to find out if they received a bill from the registrar and if they did, had the bill been paid? The accounting department says that the bill was never received. Further investigation shows that the billing point of contact that the registrar has on file left the company 8 months ago, so the renewal notice was sent to a nonexistent email account and the domain registrar does not have an effective method to deal with bounced emails.

The example above, while somewhat exaggerated is not too far from the truth. Many a large company has been crippled because someone in the accounting department did not pay the registrar bill on time. The example above also does not advise on the possibility that someone is waiting to squat on the domain is a payment is missed and registration expires. Image the embarrassment a company would have to go through if their domain was purchased out from underneath their noses. Ensuring that bills are paid on time would not normally qualify as a security issue, but in this case it certainly could be considered an aspect of availability: If an organization does not make sure the registrar is paid in a timely fashion the domain can be removed from the root servers and no one will be able to access the domain.

Even after the bill has been paid and the registrar has reinstated the domain, it can take up to 48 hours before the domain is again available to the Internet. In other words, this type of mistake can result in an outage that lasts several days—and there is not anything that can be done to speed up the process. This is why it is important to consider all aspects of availability when developing a DNS security plan.

Taking a broad view of security, a DNS security event is anything that impacts the availability of the DNS service, whether that is an internal or an external event. An internal event is one that is caused by an employee or a contractor of the organization, regardless of whether or not the event is accidental or intentional.

This is important to remember: a security breach does not necessarily have to be intentional. An administrator who enters an incorrect IP Address or accidentally deletes an important file has still created a security situation. These type of events need to planned for with as much concern as hostile events.

Internal nonhostile events can include a mistaken entry in a zone file, misconfigured ACLs, firewall rules which prevent access to DNS or grant more access than desired, deleting zone files, and of course not renewing a domain name in a timely fashion.

Internal events can also be hostile. A disgruntled employee might redirect the organization's web site, might attempt to disrupt mail service by removing entries, may change domain contact information so he is listed as the authority over the domain, or may remove a zone file completely, wreaking havoc within the network. Each of these problems can be prevented if the right checks are put in place. Again, once potential attack vectors are known it is easier to prepare for them, and in the case of internal attacks implementing stronger DNS processes goes a long way toward limiting the problem.

External security breaches are another matter; it is very rare that an external breach will be accidental. Most external attacks against DNS servers are either an instance where an organization is specifically targeted or they are random. A random attack occurs when an attacker is scanning a range of IP Addresses and encounters a DNS server with a known vulnerability. The attacker will launch an attack against that server and attempt to gain access not because the attacker has a particular grudge against the organization, but simply because it is possible. Note, an attack can be targeted and still have collateral damage. For example, in 2012 a hacker going by the name AnonymousOwn3r launched a DDoS attack against Domain Registrar. The DDoS attack not only rendered GoDaddy's web site unreachable it also impacted the ability of GoDaddy's authoritative DNS servers to respond to queries. Degrading the service of GoDaddy's customers—who were not the intended target.

Random attacks are relatively easy to defend against. Most script kiddies do not have the depth of knowledge required to launch a serious attack against a well-protected DNS infrastructure, so they will generally bypass those and focus on DNS infrastructures with weaker security measures in place. In many ways it

is the same as car thieves. Someone just looking for a joyride will focus on the easiest car to grab—one that is unlocked or with a weak alarm system. On the other hand, a skilled car thief has a greater knowledge of cars and will know how to defeat the security precautions of the car he wants.

A script kiddie is a lot like a joyriding car thief. Of course as anyone who has had his or her car stolen knows, even a novice car thief can inflict a great deal of damage—especially if it is your car stolen. Likewise, just because a script kiddie is not sophisticated technically does not make the damage inflicted any less painful.

It is important to do everything possible to keep a DNS infrastructure safe from common script kiddie attacks. At the same time DNS administrators must remain watchful for more skilled attackers.

A skilled attacker is more likely to target a specific organization for attack. The attacker may have a grudge against a company, hope to gain access to sensitive data for personal gain, or even be paid by a rival organization.

Two important qualities that good DNS administrators share are vigilance and paranoia; actually, all security administrators share those qualities. As the saying goes, "Just because you are paranoid doesn't mean they are not out to get you." Initially, it is often difficult to distinguish between an attack launched by a skilled attacker and one launched by a novice, an experienced administrator will be able to quickly determine the difference and act appropriately.

A targeted DNS attack can take many forms, depending on the intention of the attacker. If the intention of the attacker is to redirect DNS services away from an organization, then the attacker may not even target that organization's DNS servers directly. In fact, if an attacker wants to take over a domain—also known as domain hijacking—a direct attack against an organization's DNS servers is often the last resort.

A domain hijacker will take advantage of weak DNS security practices within an organization or that organization's registrar to assume ownership of a domain name. Generally, this involves some sort of social engineering. Social engineering is a form of attack that involves manipulating people rather than data. An attacker will take advantage of the willingness of people to share information, even if that information is sensitive.

There are several types of domain hijacking scenarios, and again, these scenarios may not even involve dealing directly with the organization whose domain the hijacker is trying to take over. One way to do hijack a domain is to look for one that was registered using a now-defunct mailing address from a free-mail account. The hijacker reactivates the defunct address and uses it to change the password and contact information for domain. In effect, the hijacker assumes ownership of the domain.

A second type of hijacking revolves around getting information from the domain registrar directly, and this is where social engineering really comes into play. A hijacker calls up a registrar and claims to be the administrator for a domain. The hijacker presents the registrar with a plausible crisis. Perhaps she

explains that the company that is hosting her organization's mail servers has abruptly shut down, leaving them without access to their mail. She has signed up with a new company, but she needs to update her domain information and she cannot remember her password to the registrar's control panel.

She would use the password-reset option, but obviously, with her mail unavailable, she will not receive the new password. This is a real problem, and the president of the company is calling her every 5 minutes demanding to know what the status is and even threatening to fire her. Is not there any way the registrar can reset the password over the phone—she will happily fax over a signed request on company letterhead?

At this point many support people will acquiesce and change the password "this one time," over the phone. If the hijacker does encounter resistance at this level, she will escalate it to a manager, sounding increasingly upset. Eventually, she finds someone who is willing to allow her to change the password over the phone and now she has full control over the domain without having to touch the target network.

This ploy does not always work, but remember that the primary role of the customer service person is to help people; therefore, they are naturally inclined to aid a customer in trouble. A registrar that takes security seriously would have other methods of verifying the person's identity. It is important to remember that registrars, like most service companies, depend on happy customers for repeat and new business. If the person on the other end of the phone is really a distraught customer not changing the password may result in a loss of business.

Social engineering attacks are often the most difficult to defend against, especially when an organization has to rely on a vendor to maintain the same level of security. But even within an organization not all staff members will have the same level of urgency when it comes to security, and even the best security plans are useless if people within the organization do not adhere to it.

Other types of attacks involve more traditional, computer-based, methods of aggression. These attacks generally serve to overwhelm a server making it unreachable from the network, exploit weaknesses in the DNS daemon to gain access to the server, or redirect traffic from its intended destination to a server owned by the attacker.

The first type of attack, overwhelming a server with requests making it impossible to serve legitimate requests, is what is commonly referred to as a DoS attack. The requests can be requests for DNS information, but they can also be ICMP requests, or even another service that is housed on the server.

Because DNS uses the UDP as its primary method of communication, it is especially susceptible to attacks. Unlike a Tranmission Control Protocol (TCP) packet, a UDP packet does not require a handshake to ensure that there is good communication between the two hosts. This makes UDP-based protocols especially susceptible to attack, because it is relatively trivial for an attacker to forge UDP packets. More importantly, it is trivial for an attacker to forge hundreds, thousands, or even hundreds of thousands of packets. Forged packets are sent to

the target DNS server, they look like legitimate requests, so the DNS server responds to all of them, filling up all available UDP sockets and preventing the server from responding to legitimate requests.

An Internet Control Message Protocol (ICMP) DDoS attack uses the same methodology. An attacker targets a server, but instead of launching DNS packets against the server, he uses ICMP packets. These packets can all be launched from a single server or from multiple servers. Either way, the goal is the same, overwhelm the DNS server and make it unresponsive to valid requests from other hosts.

If a DNS server has other services running on it then focusing on those other services is also an option. It does not matter what service is targeted, the important thing is to use up all of the available connections on the remote server and make it unresponsive.

A second type of attack is one that takes advantage of a weakness in either the DNS daemon or other software running on the server. The attacker exploits the weakness to gain administrative access to the server, once on the server the attacker can either attempt to make further inroads into the network or redirect DNS requests from users on the network to a rogue server controlled by the attacker.

An administrative compromise on a critical server, such as DNS servers, can be especially insidious because it allows an attacker to control parts of the network and redirect traffic away from its intended destination. Security precautions taken throughout the rest of the network become irrelevant, because the attacker has access to everything.

Attacks involving administrative compromise can sometimes go undetected for months. If an attacker is careful to cover her tracks properly and the server is poorly secured or monitored, then there is a good chance no one will notice there is a problem. At least not until long after it is too late.

A third type of attack is not as common as it used to be, but it is still one that can occur and therefore should be protected against. An attacker will load bogus information about a popular domain into a zone transfer, tricking recursive servers into redirecting queries to the wrong location.

For example, an attacker may own the domain foo.com. When DNS servers request information about foo.com, the attacker's server will also send bad data for www.amazon.com. The information is embedded within the legitimate request, so the receiving DNS server just accepts the data and shares it with users.

Note that the attacker's DNS server does not send a full zone transfer for the targeted domain, instead it generally sends a single record, most often an A record. The idea is to redirect traffic to a server owned by the attacker. So, the attacker would set up a web site that mirrored the one at www.amazon.com, send the bad data along with requests for foo.com. Compromised DNS servers would direct users toward the attacker's site and the attacker would be able to gather credit card numbers and account information from users who visit the bogus web site. Because the site would be a mirror of Amazon's web site, users would not know what happened at first, potentially giving the attacker a few weeks to exploit the gathered data.

New exploits against popular DNS daemons are constantly being discovered and reported. In addition to the exploits, new tools are released all the time that automate the process of exploiting security holes in DNS software. The confluence of these two trends creates a difficult situation for DNS administrators. Just about anyone with a computer and the ability to decompress a program can launch an attack against a poorly protected, or updated, DNS server. Because launching an elementary attack against a DNS server is so easy, the need for a strong DNS security policy is critical to any security plan.

In addition to a strong security policy, or more appropriately included as part of a strong security policy, it is important to be aware of the latest DNS exploits and understand how they impact an organization's DNS infrastructure. It is not enough to be aware of the exploit; DNS administrators must understand how the exploit works, and what it does.

Even if an exploit is not known to affect an existing DNS infrastructure—for example, an exploit is listed as being applicable to Linux servers and your DNS servers are BSD based—it cannot hurt to test the exploit against those DNS servers. Oftentimes, initial details of an exploit will be incomplete, so further research is always warranted.

Of course, even when there are no known exploits it is usually a good idea to upgrade DNS servers as soon as possible after a patch is released. Any patch should be thoroughly tested prior to upgrade, but patches generally are released to either protect against a security exploit or in anticipation of a potential new security exploit.

DEVELOPING A DNS SECURITY PLAN

A solid security plan is the key element of any organization's network and data security. A good security plan helps bring into focus the security goals of an organization, it creates policies to which people within the organization must adhere, it outlines responsibilities for different aspects of security, and it creates escalation procedures in the event of a security breach.

A well-developed DNS security plan is not going to exist in a vacuum. Most likely it will exist as a subset of an organization's larger security plan. However, there are organizations that do not have a security plan in place, in such cases, a DNS security plan should be able to stand on its own. However, even if no organizational security plan exists, a DNS security plan will have to function within the realities of the organization.

This is a problem that network and server administrators often fail to realize: The most technically correct solution is not always the most practical for an organization. Developing a security plan is always a tricky balance between security needs and meeting the needs of an organization. It is precisely for this reason that a good security plan will have broad organizational involvement.

Good security plans generally start at the top, getting senior management to approve the development of a security plan generally ensures the cooperation of all departments. Of course, if a general security plan exists for an organization the person or committee who developed the original plan should authorize a DNS offshoot. If an organization has a long-standing security plan there is generally an oversight committee that can sign off on changes to the plan, including adding a plan specifically for DNS.

The first question generally asked when developing a DNS security plan, and one you may be asking now, is "Why is there a need for a separate DNS security plan?" The short answer to that question is that DNS, more so than anything else, impacts all aspects of a network, and a compromised DNS server can have far reaching consequences. The difference between DNS and other network protocols is that DNS underlies and controls those other protocols, so if an organization's DNS infrastructure is compromised it impacts all other services.

For example, if an attacker manages to gain access to an organization's web servers, only web server access is interrupted, the same holds true with the mail server. On the other hand, if a DNS server is compromised, it can prevent access to the web server and the mail server. The unique position that DNS occupies within an organization justifies special security considerations.

Once support for a DNS security plan has been secured from the appropriate party, the next step is to make a list of people who need to be involved. In large organizations putting together a DNS security plan can stretch across multiple departments and involve a large number of people, in smaller organizations it may be as simple as grabbing the person in the next cubicle and hashing out the plan. Generally, the departments involved in implementing a DNS security plan will include those responsible for managing servers, workstations, the network, the firewalls, and possibly even the accounting department (or whoever is responsible for ensuring that bills get paid on time).

In a smaller organization, the same two or three people may fill these roles, so the planning process will be more informal. However, in larger organizations, where different departments fill these roles, with different reporting structures the planning process will have to be more formal. A formal planning process generally needs to be initiated by someone in senior management—which goes back to the previous point. The chain in larger organizations usually works as follows: An administrator feels that it is necessary to develop a DNS security plan. The administrator makes a presentation to her boss; her boss escalates the idea to the appropriate person. That person explains the idea and arranges a meeting between the appropriate groups. Alternatively, senior management may ask the person who originally came up with the idea to make a presentation to all of the department heads, the department heads will then assign someone to the task.

Once the group who will ultimately develop the DNS security plan has been assembled, it is important to set goals and to designate a clear set of responsibilities. This is a particular challenge with DNS, as the protocol crosses a wide range of areas. Some tasks will be relatively simple and involve a one-time

adjustment—with periodic review, while others will be more complex and involve ongoing maintenance.

The best way to create a set of goals and define responsibility is to assess the current level of DNS security. Any organization that has given even a passing thought to security will have implemented some basic DNS security measures. Using these measures as a foundation to build a stronger security plan adds focus to the project. Developing a chart can facilitate an initial DNS security assessment. The chart should outline potential threats to DNS security, the results if those threats are exploited, the desired DNS security level, current DNS security practices, and current DNS vulnerabilities within the organization. The chart will look similar to Table 2.1.

The security assessment involves all known threats to DNS security. Each threat should be ranked according to the danger it poses to the organization. The more serious the threat is the higher its rank and the stronger the security measures that must be taken to protect against the threat. For example, a buffer overflow attack that would give the attacker root access is serious vulnerability that could result in DNS servers being taken off-line and provide an attacker with an entry point into the network. Obviously, this is a very serious threat, and one that would need to be addressed immediately, if it was not already being addressed. The assessment for root exploits would look something like Table 2.2.

Table 2.1 DNS Security Assessment

Threats	Threat Results	Security Requirements	Current Practices	Vulnerabilities
Outline known threats to the DNS infrastructure	Worst-case scenario if those threats are exploited	Best practice security policy	Security policy currently in place	Areas in which the organization is vulnerable

Table 2.2 DNS Security Assessment: Root Exploits

Threats	Threat Results	Security Requirements	Current Practices	Vulnerabilities
Root Exploits	Could result in the disabling of all DNS functions and allow an attacker access to the network	DNS servers must be regularly patched and tested against known exploits	No set interval for testing or patching of DNS servers	Too long a period may pass between the release of an exploit or security patch and when the servers are actually patched

This systematic approach to DNS security allows the person or group tasked with securing the DNS infrastructure to prioritize security changes and set goals. Goals are important because it allows the person or group to demonstrate progress in achieving DNS security to senior management. Security costs money, even in cases where no hardware or software purchases are required the time devoted to securing a DNS infrastructure takes away from other projects. Status reports demonstrating progress lead to continued management support. The assessment is all about creating a quantifiable measure of security success.

DNS vulnerabilities can generally be placed into one of three categories. These are vulnerabilities in design, implementation, or configuration. Design vulnerabilities are those vulnerabilities that are inherent in the protocol or application. For instance, some might consider the fact that DNS uses UDP for transport a type of design vulnerability. Weaknesses in DNS software, such as root exploits, are also considered design vulnerabilities. Another vulnerability in design is not monitoring DNS traffic properly, which includes both DNS traffic and monitoring for changes the domain registrar level.

Implementation vulnerabilities are those that occur as a result of the way a solution has been deployed. Running authoritative and recursive DNS services on the same server could be thought of as an implementation threat. Placing two authoritative DNS servers on the same network could be another example. Implementation vulnerabilities do not just have to revolve around the DNS servers or software, not enabling two-factor authentication with the domain registrar could also be considered an implementation vulnerability.

Configuration vulnerabilities are the most common. These vulnerabilities are administrative errors that make a solution less secure. For example, allowing unrestricted access to zone data might be considered a configuration threat. Not assigning the correct permissions to zone files would be another example.

As the DNS security group identifies vulnerabilities, they should be classified into one of the three categories. The response to the vulnerability will depend on the category to which the vulnerability is assigned.

A threat may fall into multiple categories. When threats have been identified and classified, the next step is to determine the course of action. Generally, the response to a threat can fall into one of three categories:

1. Create a new security policy
2. Maintain existing policy
3. Address threat, without changing current policy.

Not addressing a potential threat also falls into the realm of maintaining the existing policy. If the cost to benefit ratio for fixing a problem is simply too great to garner management support it is possible that a solution will not be implemented. Fortunately, in most cases, DNS security is very cost effective.

Once the planning stage has been completed, each proposed solution has to be enabled and the security group has to follow up to ensure newly enacted processes are being followed. This means performing periodic audits of the

organization's DNS structure. The audits should be performed at random times, but with regularity.

The DNS security group should determine how often they will perform audits of the DNS infrastructure. The audits should be somewhat random, but still occur regularly. Usually a few months between audits is adequate—though some organizations may require monthly audits.

The process of implementing a strong DNS security plan does not have to be time consuming. A couple of planning sessions can smooth out the whole process and make the initial implementation proceed relatively smoothly and in a coordinated fashion. Regular audits of the system should take less than an hour—again as long as there is a strong process in place.

NOTES

1. CyberBunker has a different version of events, CyberBunker is wrong.
2. Though, it may be phrased more like "How to keep users from clicking on obvious phishing links."