Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Why is securing DNS zone transfer necessary ?

Domain Names System (DNS) is a vital and critical component of the Internet. Users often do not know anything about DNS, but they definitely use it every time they are on the Internet. DNS is the mechanism that translates IP address 192.168.1.200 to a name www.somewhere.com and vice versa. It is much easier for someone to remember a name such as www.somewhere.com than an IP address. Electronic mail, web browsing, ftp, and any other Internet related applications rely on DNS. What can be done to secure your DNS informati...

Copyright SANS Institute
Author Retains Full Rights

AD

Why is securing DNS zone transfer necessary ?
Steven Lau
March 17, 2003
GSEC version 1.4b
GSEC administrivia 2.2
Document version 2.0

Domain Names System (DNS) is a vital and critical component of the Internet. Users often do not know anything about DNS, but they definitely use it every time they are on the Internet. DNS is the mechanism that translates IP address 192.168.1.200 to a name www.somewhere.com and vice versa. It is much easier for someone to remember a name such as www.somewhere.com than an IP address. Electronic mail, web browsing, ftp, and any other Internet related applications rely on DNS.

What can be done to secure your DNS information? DNS queries, zone transfers, and dynamic updates can be secured. This paper will focus on the reason for securing DNS zone transfers between DNS Name Servers. It will concentrate on the use of allow-transfer statement in Berkley Internet Name Domain (BIND) DNS to accomplish the goal of preventing DNS poisoning or spoofing.

Why is DNS a vital and critical component of the Internet? DNS is like a map or an address book. It contains information such as host name and IP addresses of sites located on the Internet. DNS data is constructed in a hierarchical inverted tree structure. It is very similar to a family tree. The very top of the tree is known as the root. Terminology is also similar to a family tree. A parent domain is the domain that is above and responsible for creating the child domain that is immediately below it. The process of creating the link by the parent domain to a child is known as delegation.

The tree is essentially a distributed hierarchical database containing DNS information such as Name Servers, host names, IP addresses, and MX records (mail exchanger for the Domain), CNAME record (aliases), and glue records (delegation from parent to child Domain). The Name Servers consists of Primary and/or Secondary DNS Name Servers that can provide authoritative answers to

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

DNS queries (commonly refer to as DNS lookups) for its Domain. Primary and Secondary DNS Name Servers are the official DNS Name Servers for the Domain. A simple way to describe an authoritative answer is the phrase "this is what the information is". A non-authoritative answer from a Name Server can be phrase as "this is what I think the information is". The Primary and Secondary DNS servers must provide authoritative answers to DNS queries for its Domain. If it provides a non-authoritative answer, the Name Server is broken which is also known as a lame server. Some applications such a DNS zone transfers rely on authoritative answers from the official DNS Servers. The term "zone" is often used interchangeably with Domain. A zone means the same as a Domain.

The DNS information most users rely on are the host names, IP addresses, and MX records. When someone visits a web site, the user most likely type in www.someplace.com on the Internet browser rather than the IP address 192.168.1.80. Another example is with electronic mail, it will be much simpler to send e-mail to gijoe@someplace.com than gijoe@192.168.1.25. It is much easier and practical for individuals to remember names than a collection of IP address. Marketing departments also prefer to use names that can identify with products and brand names rather than some cryptic IP address. Some of the functions of DNS servers are:

1. Perform the task of resolving host name to IP address and vice versa.
2. Inform mail servers which mail servers will accept and process for a particular Domain.
3. Inform which are the official Name Servers for a particular Domain.

DNS is one of the foundation technologies to help make the Internet work. This makes it essential to protect the data integrity of DNS. You rely on it to surf the Internet just as you rely on a map when visiting a new city.

What is DNS poisoning or spoofing? It is someone changing DNS information to something else. It can be accomplished through various methods such as man in the middle attacks (intercepting communication between two parties). How can this be accomplished? One possible method is to perform a Denial of Service attack on the Primary DNS Server rendering it too busy or unable to answer any DNS queries. Another host assuming the identity of the Primary DNS Server provides different DNS information to the Domain's Secondary DNS Servers and the Internet community. The domain is essentially hijacked. False DNS information is then propagated on the Internet. If the attacker is malicious,

1. Web sites can be set up to mimic the look and feel of the original website to avoid suspicion. The fake web site can even be set-up to handle secure web transactions such as SSL to provide users some sense of security. Any online transactions and private information are compromised. The hacked company's customer private information can be stolen.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

2. MX records are changed to some other mail servers. All electronic mail destined to the domain are compromised.
3. The FTP server is redirected to some other server. Any data uploaded to the server is stolen or lost. The company's FTP server is a software repository for software drivers and patches. Now imagine only corrupt software is stored on the fake FTP server. This can wreck havoc if a customer downloaded the software and installed it onto their system.
4. Depending on the security process of the Domain's Internet Registrar, the hacker can request the Internet Registrar to change the delegation for the domain to the fake Primary DNS server. If electronic mail is used for verification for changes and the electronic mail addresses are in the compromised domain, guess who will be processing the electronic mail. Now the delegation can be changed. The fake Primary DNS server becomes the official Primary DNS server known to the Internet! The Domain is hijacked compounding the problem!
5. Good luck undoing the damage. It may take an enormous amount effort to convince the Internet Registrar to fix the delegation. They might even request that you prove who you actually are! You may even have to contact law enforcement. This is just the administration part of correcting the problem. After the problem is fixed, it will still take time for the correct DNS information to propagate on the Internet.

DNS information is taken for granted by the user community. It is imperative the DNS administrator maintains the integrity of the DNS information. It would be extremely damaging if the information were compromised. The above scenario is described without any security practices in place. The default installation of a product or operating system can very well mean no security is configured. The minimum damage can be redirecting a corporation's web site or mail to a competitor, a non-business related web site, or nowhere. The result is still damaging to the company's reputation. The threat is real and not hypothetical. Eugene Kashpureff was able to compromise InterNIC's DNS by exploiting a flaw in the version of the software in July 1997. Web traffic destined to InterNIC was redirected to his web site. When you are performing any type of Internet transaction, you and your customers expect to be exchanging private information with each other and not with someone else!

What exactly is a zone transfer? It is an answer to a DNS query to list all DNS information (such as Name Servers, host names, MX records, CNAME records, glue records (delegation for child Domains), zone serial number, Time To Live (TTL) records, etc) for a Domain. The query can be made from a single host to look up information for the entire Domain. It is also the mechanism Primary and Secondary Name Servers use to update its DNS data. This is one of the vulnerable points where a malicious person can compromise DNS data integrity.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

The default behavior for DNS zone transfer permits any host to request and receive a full zone transfer for a Domain. This is a security issue since DNS data can be used to decipher the topology of a company's network. The information obtained can be used for malicious exploitation such as DNS poisoning/spoofing. This is like an anonymous person calling the receptionist to request and receive the entire company's telephone and address book.

The two common methods of securing zone transfer between DNS Name Servers can be accomplished by utilizing BIND DNS access control list and DNS Transaction Signatures (TSIG). The allow-transfer feature can be used as an options substatement or as a zone substatement in named.conf. The options substatement is a global definition. The zone substatement is a local definition pertaining only to the zone/domain. The zone substatement will override the options substatement.

The BIND DNS access control list (acl) consists of BIND DNS address match lists or a group of IP addresses. It is not like a Cisco Access Control List that performs packet level filtering. When used in conjunction with some of the BIND DNS security features, it can be used to restrict or deny services such as zone transfers or answer queries.

If you decide to use the BIND address match list, there are four predefined options. n*one* is no IP addresses allowed. *any* permits any IP addresses. *localhost* is any IP address on the host running named. *localnets* is any network directly connected to host running named.

Restricting zone transfers with the use of BIND access control list can be accomplished with simply utilizing the feature allow-transfer with a set of host or network IP addresses or a match address list. This will permit only hosts approved to request and receive a zone transfer. All other hosts requesting a zone transfer will be refused.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Example :

Extracted information on the Primary Name Server named.conf pertaining to zone transfer :

```
acl ournets { 192.168.1.0.0/22; 172.16.1.0/24; };

acl mynets { 10.1.1.53; };

allow-transfer { ournets; mynets; }

zone cooking.com {
        type master;
        file cooking.zone;
        allow-transfer { ournets; mynets; };
        };
```

This will permit zone transfers between the Primary Name Server and any hosts on the networks 192.168.1.0 to 192.168.3.255, 172.16.1.0 to 172.16.1.255, and host 10.1.1.53 for the domain cooking.com.

```
zone barbecue.cooking.com {
        type master;
        file barbecue.cooking.zone;
        allow-transfer { mynets; };
        };
```

This will permit zone transfers between the Primary Name Server and the host 10.1.1.53 for the domain barbecue.cooking.com.

```
zone deepfry.cooking.com {
        type master;
        file deepfry.cooking.zone;
        allow-transfer { ournets; };
        };
```

This will permit zone transfers between the Primary Name Server and any hosts located on the 192.168.1.0 to 192.168.3.255 and 172.16.1.0 to 172.16.1.255 networks for deepfry.cooking.com Domain.

All other hosts requesting a zone transfer from the Name Server oven.cooking.com will be refused.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Extracted information on respective Secondary Name Servers' named.conf pertaining to zone transfer :

```
// hosts on 192.168.1.0 to 192.168.3.255,  172.16.1.0 to 172.16.1.255,
// and 10.1.1.53 host
zone cooking.com {
        slave;
        master { 10.1.1.10; };
        file cooking.zone.cache;
        allow-transfer { none; };
        };


// host on 10.1.1.53 network
zone barbecue.cooking.com {
        type slave;
        master { 10.1.1.10; };
        file barbecue.cooking.zone.cache;
        allow-transfer { none; };
        };


// host on 192.168.1.0 to 192.168.3.255 and 172.16.1.0 to 172.16.1.255 networks
zone deepfry.cooking.com {
        type slave;
        master { 10.1.1.10; };
        file deepfry.cooking.zone.cache;
        allow-transfer { none; };
        };
```

The benefits of using BIND acl to secure DNS zone transfers are:

1. Restricting zone transfer to only permitted hosts
2. Configuration management is simple to understand and maintain.

The pitfall for this type of configuration is that it does not address IP spoofing. Authentication is based solely on the source IP address. Security experts will argue this practice is relatively insecure. There is still a potential that some host can pretend to be the Primary DNS Server for the Domain and trick the Secondary DNS Servers. False DNS information can still be polluted on the Internet. This configuration should only be used on internal networks with security practices in place.

The next level of securing transfer zones is the deployment of DNS Transaction Signatures (TSIG) between the DNS name servers. DNS TSIG is defined by

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

RFC 2845. DNS TSIG provides a level of security that ensures the information from the primary name server is actually from the primary name server.
What is DNS TSIG ? DNS TSIG "uses shared secrets and a one way hash function to authenticate DNS messages, particularly responses and updates"[1]. A DNS server configured to use DNS TSIG adds an additional record to the DNS data section of a message. The DNS TSIG record is added to the DNS transaction. The record is not visible. It does not appear on the zone data or cached by the DNS Name Server. The DNS TSIG record is basically a signature. MD5 (HMAC-MD5) is the current one way hash function used by DNS TSIG. Other one way hash functions might be available in the future. A one way hash function is a mathematical algorithm that takes the entire input value and creates a fixed size result. The hash value is computed over the entire DNS message. Any changes to the input data will alter the result significantly. The hash value is computed with a shared secret between the two DNS servers. One DNS Server is the signer and the other the verifier per transaction. Once the hash value is verified, it proves that some host with the shared secret signs the DNS data and the hash value was not altered after it was signed. Thereby, the DNS data can be trusted. The hash value provides a point to point authentication and verification for DNS transactions. A trust is basically established between the two communicating DNS Servers.

The DNS TSIG RFC recommends for ease of readability and identification the following format for naming the TSIG key primary name server, secondary name server, and zone. The DNS TSIG key can be generated by the dnssec-keygen for BIND 9 and dnskeygen for BIND 8 program.

It is recommended to define unique key pairs between named servers. This will prevent a single compromise of a key to a broader security breach.

Once DNS TSIG is configured for DNS zones transfers between the Primary and Secondary DNS servers, the basic interaction will be:

1. A server will sign its request with the corresponding key
2. The queried server will verify the key
    - If the key is valid, sign the answer with its corresponding key to the request.
    - If the key is invalid, the server will reject the request.
3. If the requesting server receives an answer, it will verify the key and process the DNS data.

[1] Albitz, Paul and Liu, Cricket. DNS and Bind Fourth Edition, O'Reilly & Associates, April 2001 p309

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Examples for generating DNS TSIG keys:

The first step is to generate DNS TSIG keys for each server pair. The arguments for dnssec-keygen are –a is the name of the algorithm, -b is the key length in bits (RFC recommends 128), -n argument type for DNS TSIG is host, and followed by the name of the key.

```
# dnssec-keygen –a HMAC-MD5 –b 128 –n HOST oven-skillet.cooking.com.

# dnssec-keygen –a HMAC-MD5 –b 128 –n HOST oven-grill.cooking.com.

# dnssec-keygen –a HMAC-MD5 –b 128 –n HOST oven-fryer.cooking.com.
```

The following are the file names and its contents generated by the dnssec-keygen command :

for use between oven.cooking.com and skillet.cooking.com:

```
Koven-skillet.cooking.com.+157+63995.key
  oven-skillet.cooking.com. IN KEY 512 3 157
  bFNvA4Z8uwb3pbBiX2vMMg==

Koven-skillet.cooking.com.+157+63995.private
    Private-key-format: v1.2
    Algorithm: 157 (HMAC_MD5)
    Key: bFNvA4Z8uwb3pbBiX2vMMg==
```

for use between oven.cooking.com and grill.cooking.com:

```
Koven-grill.cooking.com.+157+18693.key
oven-grill.cooking.com. IN KEY 512 3 157 k0My8y+aSmn0UOb6H/MNjA==

Koven-grill.cooking.com.+157+18693.private
    Private-key-format: v1.2
    Algorithm: 157 (HMAC_MD5)
    Key: k0My8y+aSmn0UOb6H/MNjA==
```

for use between oven.cooking.com and fryer.cooking.com:

```
Koven-fryer.cooking.com.+157+50776.key
oven-fryer.cooking.com. IN KEY 512 3 157 FlqaZQECmw2GWAJYcxp9vg==

Koven-fryer.cooking.com.+157+50776.private
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: FlqaZQECmw2GWAJYcxp9vg==
```


© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

There is a meaning for the cryptic key file name. It is put together with the key name, algorithm type (157 is HMAC-MD5), and the key's fingerprint. The key's fingerprint is not used in DNS TSIG. The key value needs to be extracted and shared with the corresponding Secondary Name Server. The key value can be exchanged through various means such as voice communication, encrypted message within an e-mail, scp, etc. The key by itself does not perform any function. It needs to be used in conjunction with one of BIND's security feature such as allow-transfer to be of any use. The keys need to be treated in the same fashion as private keys in a Public Key Infrastructure (PKI) environment. The keys need to be stored in a secured location on each host. If the key is compromised, the DNS vulnerabilities are the essentially the same as using BIND DNS access control list.

Example utilizing DNS TSIG on the Primary Name Server's named.conf:

```
// for communicating with skillet.cooking.com
server 10.1.1.53 {
        keys { oven-skillet.cooking.com.; };
};

// for communicating with grill.cooking.com
server 192.168.1.53 {
        keys { oven-grill.cooking.com.; };
};

// for communicating with fryer.cooking.com
server  172.16.1.53 {
        keys { oven-fryer.cooking.com.; };
        };

key oven-skillet.cooking.com. {
        algorithm hmac-md5;
        secret "bFNvA4Z8uwb3pbBiX2vMMg==";
};

key oven-grill.cooking.com {
        algorithm hmac-md5;
        secret "k0My8y+aSmn0UOb6H/MNjA==";
};

key oven-fryer.cooking.com {
        algorithm hmac-md5;
        secret "FlqaZQECmw2GWAJYcxp9vg==";
};
```


© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
zone “cooking.com” {
        type master;
        file “cooking.zone”;
        allow-transfer { key oven-skillet.cooking.com; };
};

zone barbecue.cooking.com {
        type master;
        file barbecue.cooking.zone;
        allow-transfer { key oven-grill.cooking.com; };
        };

zone deepfry.cooking.com {
        type master;
        file deepfry.cooking.zone;
        allow-transfer { key oven-fryer.cooking.com; };
        };
```

The statement server identifies the respective key to be used with the corresponding DNS server. The keyword “key” in the allow-transfer option invokes the use of DNS TSIG.

The DNS TSIG keys need to be added to corresponding Secondary Name Server’s named.conf :

```
// for communicating with oven.cooking.com
server 10.1.1.10 {
        keys { oven-skillet.cooking.com.; };
};

// key for skillet.cooking.com
key oven-skillet.cooking.com. {
        algorithm hmac-md5;
        secret “bFNvA4Z8uwb3pbBiX2vMMg==”;
};

zone cooking.com {
        slave;
        master { 10.1.1.10; };
        file cooking.zone.cache;
        allow-transfer { none; };
        };
```


© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

The DNS TSIG key bFNvA4Z8uwb3pbBiX2vMMg== is for name server pair oven.cooking.com and skillet.cooking.com. This permits skillet.oven.cooking.com to sign request zone transfers for the domain cooking.com from oven.cooking.com. oven.cooking.com verifies the signed request if approved it signs the zone data with the DNS TSIG key for skillet.cooking.com to verify.

```
// for communicating with oven.cooking.com
server 10.1.1.10 {
        keys { oven-grill.cooking.com.; };
        };

// key for grill.cooking.com
key oven-grill.cooking.com {
        algorithm hmac-md5;
        secret "k0My8y+aSmn0UOb6H/MNjA==";
};

zone barbecue.cooking.com {
        type slave;
        master { 10.1.1.10; };
        file barbecue.cooking.zone.cache;
        allow-transfer { none; };
        };
```

The DNS TSIG key k0My8y+aSmn0UOb6H/MNjA== is for name server pair oven.cooking.com and grill.cooking.com. This permits grill.cooking.com to sign request zone transfers for the domain barbecue.cooking.com from oven.cooking.com. oven.cooking.com verifies the signed request if approved it signs the zone data with the DNS TSIG key for grill.cooking.com to verify.

```
// for communicating with oven.cooking.com
server 10.1.1.10 {
        keys { oven-fryer.cooking.com.; };
        };

// key for fryer.cooking.com
key oven-fryer.cooking.com {
        algorithm hmac-md5;
        secret "FlqaZQECmw2GWAJYcxp9vg==";
};

zone deepfry.cooking.com {
        type slave;
```


© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
master { 10.1.1.10; };
file deepfry.cooking.zone.cache;
allow-transfer { none; };
};
```

The DNS TSIG key FlqaZQECmw2GWAJYcxp9vg== is for name server pair oven.cooking.com and fryer.cooking.com. This permits fryer.cooking.com to sign request zone transfers for the domain deepfry.cooking.com from oven.cooking.com. oven.cooking.com verifies the signed request if approved it signs the zone data with the DNS TSIG key fryer.cooking.com to verify.

The DNS TSIG examples above show the DNS TSIG keys as part of the named.conf file for presentation purposes. It is recommended to use the INCLUDE statement in named.conf for the DNS TSIG keys. This will prevent anyone from obtaining the DNS TSIG key values from named.conf. The Secondary DNS server also will not permit any zone transfer request from any other hosts. If zone transfers are necessary from Secondary DNS Servers, DNS TSIG is certainly an option to be implemented.

The DNS TSIG keys should be changed periodically. This will prevent the keys from being deciphered over time. A formal process should be defined since coordination of key exchanges and implementation of the new keys are critical to the health of utilizing DNS TSIG.

The benefits of using DNS TSIG to secure DNS zone transfers are:

1. Restricts zone transfers to only hosts configured with valid DNS TSIG keys.
2. Each DNS requests and answers can be verified for valid DNS TSIG keys.
3. Configuration management is still straightforward and requires only general understanding of cryptographic terminology.

What are the pitfalls? DNS TSIG is not a true signature defined by cryptography terms because it does not provide nonrepudiation. Nonrepudiation is the ability where one cannot deny or refute its authenticity. With DNS TSIG, there is a potential either holder of the shared key can create the same signature thereby the signature cannot claim to be unique. The recipient can forge the signature; send the DNS message with the forged signature back to itself! The method of the key exchange can be cumbersome. It requires the Domain administrators to securely exchange the key values between two servers. This can be accomplished through various means such as encrypted message within an e-mail, voice communication, sneaker net, etc. Time synchronization is very crucial. A secured Network Time Protocol (ntp) source must be used to avoid rejection of messages due to time expiration. Time is a value in the DNS TSIG record. It is used to prevent replay attacks such as someone capturing, changing, and then sending the corrupted message. When time is not

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

synchronized between the DNS servers, this can lead to false positives or replay attacks. It is prohibitive to use DNS TSIG in a large-scale environment because of scalability issues. It is labor intensive for configuration management. Interoperability with Microsoft DNS is currently not feasible. This is due to Microsoft deviation from the RFC 2845 of using HMAC-MD5 algorithm for one way hash. The good news is ISC BIND ported a version of DNS BIND for Windows NT and Windows 2000.

DNSSEC will hopefully address some of the security and scalability concerns of DNS TSIG in the near future. DNSSEC addresses the scalability issue and nonrepudiation issue with DNS TSIG. DNSSEC is defined in RFC2535. It uses cryptographic digital signatures for authenticating DNS data. This provides the DNS zone administrators the ability to digitally sign and verify DNS data.

BIND DNSSEC uses an asymmetric cryptography algorithm such as RSA. "One key is used to decrypt the data that another encrypts."[2] A host will have a public key and private key. When host A needs to communicate with host B, host A encrypts the data with host A's private key and host B's public key. Only host B will be able to decrypt the data using host A's public key and its private key.

DNS messages can be large in size. This is not ideal for asymmetric cryptography. Encrypting and decrypting the entire DNS messages are not practical due to performance issues. So rather than encrypt the entire DNS message, DNSSEC attaches a digital signature for authentication. The DNS message is processed by an one way hash function that creates a hash value. The hash value is then encrypted. The encrypted hash value is known as the digital signature. DNS data also needs to be publicly available, so authenticating the data is the right approach. The DNS Servers can process transactions such as queries and zone transfers by first verifying the digital signature. The digital signature is generated with the use of host A's private key with host B's public key. Only host B will be able to decrypt the digital signature by using host A's public key and its private key. Since host A cannot deny it generated the digital signature for host B, nonrepudiation is ensured. The process is the same when host B communicates with host A. Once the digital signature is verified for its authenticity, the DNS Server can process the DNS request.

The private key of a host must be stored in a secured location on the host. If the private key is compromised, any host with the private key can encrypt and decrypt messages with a corresponding host's public key. This can possibly lead to a security breach.

It is important to carefully design your deployment of DNSSEC. Important designs are based on whether DNSSEC is deployed and operational at the root

[2] Albitz, Paul and Liu, Cricket. DNS and Bind Fourth Edition, O'Reilly & Associates, April 2001 p350

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

server, and will DNSSEC be deployed on the parent and all or some child domains. These parameters along with implementation specific parameters such as agreement of cryptographic algorithms and key lengths need to finalized ahead of time to ensure a smooth and successful deployment of DNSSEC for your Domain. Since DNSSEC is a not a trivial exercise, not widely deployed, and has technology maturity concerns, it is not recommended for a production environment. DNSSEC requires a much more in depth analysis and explanation, which is beyond the scope of this document.

"Due diligence" on securing, hardening, and maintaining patch levels for the Operating System and applications on the DNS host is expected. Details on how to perform the operation can be found from various practicals submitted to SANS for securing Unix, Windows, and DNS configurations. If the Primary DNS Server is compromised, the DNS information and any other services provided by the host are jeopardized.

The goal is to protect your DNS information and at minimum make it much more difficult for DNS poisoning or spoofing. The task of securing DNS zone transfer is within everyone's capability. The document described methods to secure DNS zone transfers between DNS Name Servers and the possible outcome if no security mechanism is used. This is just one step of many to create a secure computing environment. Operational processes such as reviewing system logs, configuration revision control, verifying software patch levels, and technical reviews are just as important to maintain a secure computing environment. There will always be a curious user on the Internet looking for an adventure or challenge. The intentions may range from a practical joke to the intent of causing financial damages.

A decision has to be made factoring technical staff capability, fiscal cost, and available technology to create a balance and workable secure computing environment. There is no sense to procure the latest technology if the personnel are not sufficiently trained to support and implement the solution. One will not implement DNSSEC in their environment with a novice technical staff with the latest hardware and software. A misconfigured server is detrimental to providing quality service as well as security. Nor will it be wise to implement DNSSEC with antiquated hardware with an expert technical staff. Also keep in mind, in the security field a little paranoia is good.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

References :

Albitz, Paul and Liu, Cricket. DNS and BIND Fourth Edition, O'Reilly & Associates, April 2001

Liu, Cricket. DNS & BIND Cookbook, O'Reilly & Associates, October 2002

Secret Key Transaction Authentication for DNS (TSIG) RFC 2845
http://www.ietf.org/rfc/rfc2845.txt

Secret Key Establishment for DNS (TKEY RR) RFC2930
http://www.ietf.org/rfc/rfc2930.txt

Domain Name System Security (DNSSEC) Signing Authority RFC 3008
http://www.ietf.org/rfc/rfc3008.txt

DNS Security Extension Clarification on Zone Status RFC 3090
http://www.ietf.org/rfc/rfc3090.txt

DNS Security Extensions RFC2535
http://www.ietf.org/rfc/rfc2535.txt

RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS) RFC 3110
http://www.ietf.org/rfc/rfc3110.txt

Indicating Resolver Support of DNSSEC RFC3225
http://www.ietf.org/rfc/rfc3225.txt

Notes from the State-of-The-Technology: DNSSEC
http://www.faqs.org/rfcs/rfc3130.html

DNS Zone Transfer Protocol Clarifications
http://www.ietf.org/internet-drafts/draft-ietf-dnsext-axfr-clarify-05.txt

DNS Security Document Roadmap
http://www.ietf.org/internet-drafts/draft-ietf-dnsext-dnssec-roadmap-06.txt

Threat Analysis of the Domain Name System
http://www.ietf.org/internet-drafts/draft-ietf-dnsext-dns-threats-02.txt

DNS Extensions Work Group
http://www.ietf.org/html.charters/dnsext-charter.html

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Eugene E. Kashpureff Pleaded Guilty to Unleashing Software on the Internet That Interrupted Service for Tens of Thousands of Internet Users Worldwide
http://www.usdoj.gov/criminal/cybercrime/kashpurepr.htm

Defense Department faces hurdles with DNS Security
http://www.nwfusion.com/news/2002/1007doddns.html

Internet Software Consortium for BIND DNS software
http://www.isc.org/products/BIND/

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: June 20th, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

Digital Forensics & Incident Response Summit 2013	**Austin, TXUS**	**Jul 09, 2013 - Jul 16, 2013**	**Live Event**
SANS London Summer 2013	**London, GB**	**Jul 09, 2013 - Jul 16, 2013**	**Live Event**
SANS Rocky Mountain 2013	**Denver, COUS**	**Jul 14, 2013 - Jul 20, 2013**	**Live Event**
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	**Washington, DCUS**	**Jul 22, 2013 - Jul 26, 2013**	**Live Event**
SANS Mumbai 2013	**Mumbai, IN**	**Jul 22, 2013 - Jul 27, 2013**	**Live Event**
SANS San Francisco 2013	**San Francisco, CAUS**	**Jul 29, 2013 - Aug 03, 2013**	**Live Event**
SANS Boston 2013	**Boston, MAUS**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	**Bangalore, IN**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
Critical Security Controls Summit	**Washington, DCUS**	**Aug 12, 2013 - Aug 18, 2013**	**Live Event**
Industrial Control Systems Security Training - DC	**Washington, DCUS**	**Aug 12, 2013 - Aug 16, 2013**	**Live Event**
SANS Virginia Beach 2013	**Virginia Beach, VAUS**	**Aug 19, 2013 - Aug 30, 2013**	**Live Event**
SANS Thailand 2013	**Bangkok, TH**	**Aug 19, 2013 - Aug 31, 2013**	**Live Event**
SANS Melbourne 2013	**Melbourne, AU**	**Sep 02, 2013 - Sep 07, 2013**	**Live Event**
SANS Capital City 2013	**Washington, DCUS**	**Sep 03, 2013 - Sep 08, 2013**	**Live Event**
Network Security 2013	**Las Vegas, NVUS**	**Sep 14, 2013 - Sep 23, 2013**	**Live Event**
SANS Canberra 2013	**OnlineAU**	**Jul 01, 2013 - Jul 13, 2013**	**Live Event**
SANS OnDemand	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**