Global Information Assurance Certification Paper

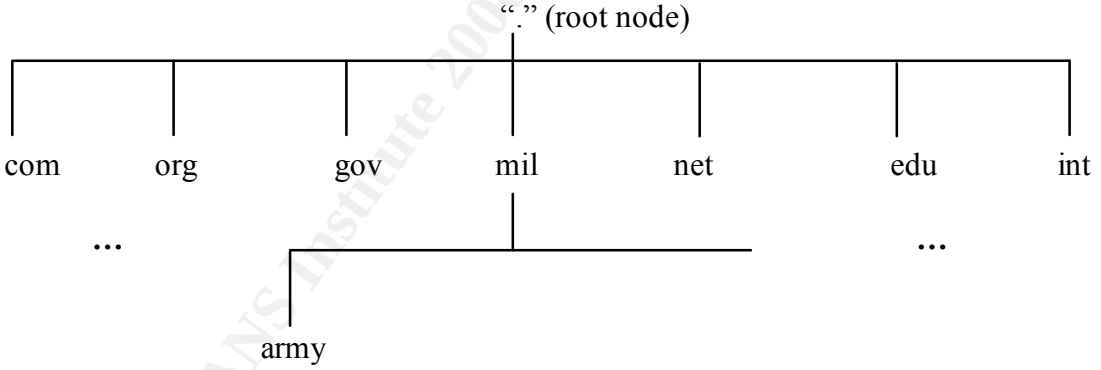
DNS Security
Jeff Holland
July 23, 2000

This paper will address security issues involved with the DNS client/server architecture within a UNIX environment. Suggestions on securing DNS by preventing unauthorized zone transfers will also be discussed.

Background

The name service DNS (Domain Name Service) is a distributed database that allows for the translation of domain names into IP addresses. Inverse queries that map IP addresses to domain names are also possible, but are not entirely accurate [1]. To determine IP address to host-name mappings, consult the IN-ADDR.ARPA domain, which was created for this very purpose [1].

DNS has a hierarchical inverted tree structure, with a root node and seven immediate subdomain nodes below the root [2]. These subdomain nodes, which are domains themselves, are the top-level domains and are controlled by InterNIC (Internet Network Information Center) [3]. For example, given the small sample name space below, the "army" subdomain is a child domain of the parent domain "mil", which in turn is a subdomain of the root node. Combinations of these domains and subdomains form unique domains, such as "army.mil". A domain such as this will make use of the DNS architecture.



The client piece of the DNS architecture is known as a "resolver", and the server piece is known as a "name server" [1]. Resolvers retrieve information associated with a domain name, and domain name servers store various information about the domain space and return information to the resolvers upon request [1]. Name servers may be a primary or a secondary name server for its particular domain [1]. There are also name servers called "caching-only" name servers [4]. These servers will resolve name queries, but do not own or maintain any DNS database files. Changes to primary domain name servers must be propagated to secondary name servers, as primary domain name servers are authoritative and own the database records. This is accomplished via a "zone transfer",

which copies a complete DNS database for a particular subdomain of the Internet domain space to another primary domain or secondary domain name server [1,2].

Zone Transfers

Zone transfers pose a significant risk for organizations running DNS. While there are legitimate and necessary reasons for why zone transfers occur, an attacker may attempt a zone transfer request from any domain name server on the Internet. The reason attackers do this is to gather intimate details of an organization's network, and use this information for further reconnaissance or as a launch pad for an attack. For instance, suppose the name server for the army.mil domain returned DNS entries for machines on the internal network named "intel", "bases", or "troops". Armed with this information, an attacker now has the addresses and names of potential targets [5]. Using this information, the attacker could then attempt to use automated attack scripts to exploit vulnerabilities in various UNIX services [6].

For example, assume an attacker was able to obtain the IP addresses and host names of machines in the victim's DMZ (Demilitarized Zone) via a zone transfer. The attacker could then telnet to port 25 on a mail server if the external router was not configured to prevent unauthorized Telnet connections. If the line referencing the version number of Sendmail was not commented out or falsified in /etc/mail/sendmail.conf, the attacker would know what version of Sendmail the mail server is running. They could then lookup Sendmail exploits for that version on one of many "black-hat" websites.

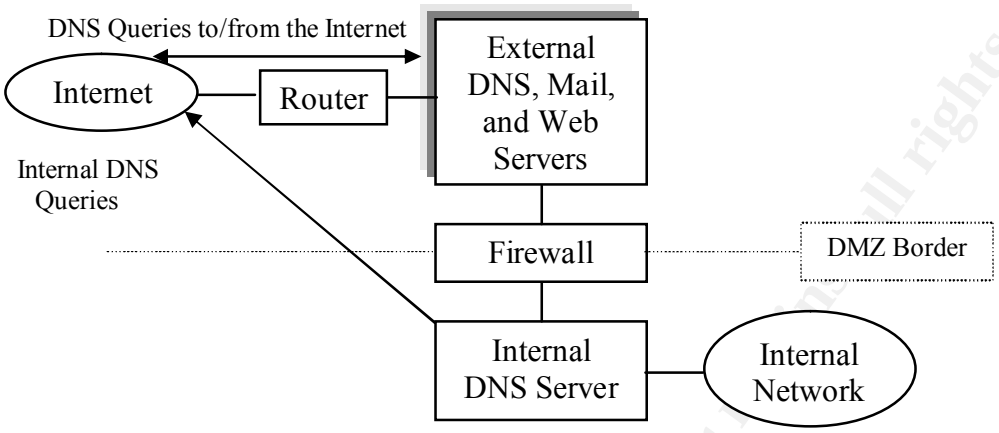
The attacker's job is simplified by the existence of legitimate websites that host DNS tools. One such site is http://samspade.org. The SamSpade.org site provides automated, web-based services such as DNS queries, reverse DNS queries, and Who Is lookups.

Blocking Malicious Zone Transfers

Incorporating a split-DNS architecture may reduce the risk that zone transfers pose. Split-DNS uses a DNS domain server for publicly reachable services within the DMZ, and a DNS domain server for the private internal network [7,8]. The public DNS server, and usually public www and mail servers, are the only servers defined in the public DNS server's database [9]. While the internal DNS server contains all the private server and workstation information for the internal network, the /etc/resolv.conf file on each internal client should be updated so that entries point to the internal DNS server [4]. In addition, the /etc/nsswitch.conf file on each client should be updated to first look at the /etc/inet/hosts file before querying the DNS server [10]. To accomplish this, the following line should be added to /etc/nsswitch.conf: "`hosts: files dns`".

If internal users are allowed to access the Internet, the firewall should allow the internal DNS server to query the Internet. Internal users should not query the external DNS server, as sensitive internal host information would then be stored in the external DNS database records. All DNS queries from the Internet should use the external DNS server. Outbound DNS queries from the external DNS server should also be permitted.

A split-DNS architecture using an unprotected DMZ is illustrated below [9,10].



Another method by which unauthorized zone transfers may be blocked is by installing the most recent version of BIND (Berkeley Internet Name Domain), an implementation of DNS [5]. BIND version 4.9.3, and up to version 4.9.6, has a directive called "`xfernets`". This directive, which should be set in /etc/named.boot, will apply an access list to zone transfers by IP address [11]. BIND version 8.1 (and higher) uses the "`allow-query`" directive, which is also set in /etc/named.boot, to impose access list controls on DNS queries [12].

The BIND 8 tar file includes the tool "dig" (Domain Information Groper), which may be used to debug DNS servers and test security by generating queries that run against the server. For instance, the command "`dig -t txt -c chaos VERSION.BIND @intel.army.mil`" will query for the version of BIND running on the DNS name server [13]. BIND also comes with the "nslookup" tool. This is useful for doing inverse IP address to host-name DNS queries. For instance, the command "`nslookup 172.16.10.20`" will actually perform a regular DNS query on the domain name 20.10.16.172.in-addr.arpa. Note that the IP address is reversed due to the hierarchical structure of DNS [14]. While dig performs many of the same functions as nslookup, nslookup continues to be useful as it can be used in either batch or interactive mode [18].

If your firewall is stateful, enforce packet filtering for UDP and TCP port 53 (DNS) [15]. By doing so, IP packets bound for UDP port 53 from the Internet are limited to authorized replies from queries made from the internal network [12]. If such a packet were not replying to a request from the internal DNS server, the firewall would deny it. The firewall should also deny IP packets bound for TCP port 53 on the internal DNS server, except for those from authorized external secondary DNS servers, to prevent unauthorized zone transfers [19,20]. This is redundant functionality if access control has been established using BIND, but it does provide "defense in depth".

Finally, if using BIND version 8, add the following line to the options block of /etc/named.conf: "`query-source address * port 53;`". This is due to the fact that BIND version 8 chooses random port numbers above 1023 for server to server queries, which most firewalls cannot handle [16].

Recommendations

1. Obtain the most recent version of BIND, version 8.2.2-P5 at the time of this writing, from http://www.isc.org.

2. Configure access control via the "`allow-query`" directive in the /etc/named.boot file. Add the line "`query-source address * port 53;`" under the options block of the /etc/named.conf file to force both servers to use UDP port 53 in a server to server DNS query.

3. Use the dig tool provided in the BIND distribution to debug DNS problems. Other DNS tools are available here: http://www.dns.net/dnsrd/tools.html#star.

4. Subscribe to BIND security mailing lists to stay current on DNS vulnerabilities, such as the "bind-announce" list maintained by ISC. Go to: http://www.isc.org/services/public/lists/bind-lists.html to subscribe.

5. Use a split-DNS architecture.

6. Use a state-based firewall, such as CheckPoint's FireWall-1, Cisco's PIX, or Network Associate Inc.'s Gauntlet [12,17].

7. Enable state-based packet filtering on the firewall for the DNS protocol.

8. See the SANS "Top Ten Internet Security Threats" page for further recommendations on securing BIND. Go to: http://www.sans.org/topten.htm for details.

9. Consider purchasing a reference book on DNS, and read it religiously. An excellent reference is "*DNS and BIND*" by Paul Albitz and Cricket Liu, available at the following URL:
http://www.amazon.com/exec/obidos/ASIN/1565925122/qid=964397720/sr=1-1/104-8896135-4926332

References

[1] Mockapetris, P. "Domain Names – Implementation and Specification." RFC 1035. November, 1987. URL: http://www.cis.ohio-state.edu/htbin/rfc/rfc1035.html (21 July, 2000).

[2] Mockapetris, P. "Domain Names – Concepts and Facilities." RFC 1034. November, 1987. URL: http://www.cis.ohio-state.edu/htbin/rfc/rfc1034.html (21 July, 2000).

[3] Berg, Al. "Take the magic out of mapping: How does Domain Naming Service map user-friendly Internet addresses to computer-friendly numeric addresses?" URL: http://www.wcmh.com/handson/97/706a107a.html (21 July, 2000).

[4] SunSolve Publication. "Product Support Document (PSD) for Sun DNS." Infodoc ID 11975. 19 February, 1997.
URL: http://sunsolve.Sun.COM/pub-cgi/retrieve.pl?type=0&doc=infodoc/11975 (22 July, 2000).

[5] Network Ice Corporation. "DNS Zone Transfer." 2000. URL: http://www.netice.com/advice/intrusions/2000401 (22 July, 2000).

[6] Spitzner, Lance. "Know Your Enemy." 21 July 2000. URL: http://www.enteract.com/~lspitz/enemy.html (21 July, 2000).

[7] Network Ice Corporation. "Split-DNS" URL: http://www.netice.com/advice/Services/Directory/DNS/split-DNS/default.htm (22 July, 2000).

[8] Berkowitz, H. "Router Renumbering Guide." January, 1997. RFC 2072. URL: http://www.cis.ohio-state.edu/htbin/rfc/rfc2072.html (22 July, 2000).

[9] Spitzner, Lance. "Building Your Firewall Rulebase." 26 January 2000. URL: http://www.enteract.com/~lspitz/rules.html (21 July, 2000).

[10] Gregory, Peter H. "Solaris Security." New Jersey, Prentice Hall PTR, 2000. URL: http://www.amazon.com/exec/obidos/ASIN/0130960535/qid%3D964378453/103-2327575-3879020

[11] Mr. DNS. "Restricting zone transfers in BIND 4.9.x with the xfernets directive." URL: http://acmebw.com/askmrdns/00031.htm (22 July, 2000).

[12] Matt Larson and Cricket Liu. "Using BIND: Don't get spoofed again: Learn how to secure your Internet domain name servers." June 5, 2000.
URL: http://www.sunworld.com/swol-11-1997/swol-11-bind.html (22 July, 2000).

[13] Network Ice Corporation. "dig." 2000. URL: http://www.netice.com/advice/Reference/Tools/dig/default.htm (22 July, 2000).

[14] Gray, Damon. "The "IN-ADDR.ARPA" domain and it's relation to DNS." URL: http://www.wednet.edu/network/whitepapers/in-addr.arpa.domain-whitepaper.html (23 July, 2000).

[15] Reynolds, J. and Postel, J. "Assigned Numbers." RFC 1010. May 1987. URL: http://www.cis.ohio-state.edu/htbin/rfc/rfc1010.html (21 July, 2000).

[16] Pomeranz, Hal and Deer Run Associates. "Deploying DNS and Sendmail." Sun-I-4, October 3, 1999. URL: http://www.deer-run.com/dns-send.html

[17] Network Associates, Inc. "Gauntlet Firewall 5.5 for UNIX." 2000 URL: http://www.pgp.com/asp_set/products/tns/jump_page_11_1.asp (22 July, 2000).

[18] Salamon, András. “Tools to manage DNS.” 1999. URL: http://www.dns.net/dnsrd/tools.html (23 July, 2000).

[19] Spitzner, Lance. "DNS Access." 26 January 2000. URL: http://www.enteract.com/~lspitz/rules/rule6.html (21 July, 2000).

[20] The SANS Institute. “How To Eliminate The Ten Most Critical Internet Security Threats: The Experts’ Consensus.” 12 July, 2000. Version 1.25. URL: http://www.sans.org/topten.htm (23 July, 2000).