6 November 2000
Sinéad Hanley (GSEC practical)

DNS Overview with a discussion of DNS Spoofing

DNS Overview

DNS (Domain Name System) is a distributed database that maps domain names to IP addresses. DNS offers a practical way around the difficulty of maintaining large hosts files in an organisation. On a larger scale, Internet-connected computers use DNS to resolve URLs (Universal Resource Locators). In this way, one doesn't need to know the IP address of a Web server – just its name.

DNS names take the form <domain>.<domain type>, e.g. sans.org. While the list of available DNS types is currently being redesigned by ICANN (Internet Corporation for Assigned Names and Numbers), some popular existing types include .edu (educational establishments), .mil (military organisations), .org (non-profit organisations), and .com (commercial organisations). There are also country-specific domain types, e.g. .ie (Ireland), .jp (Japan) and .de (Germany).

The Internet makes use of a network of DNS servers to form a distributed database of mappings between domain names and IP addresses. At the top of this network of servers sit thirteen root DNS servers [3], and beneath these root DNS server sit the top-level servers.

When a computer (a DNS client) wants to resolve a URL, it queries ('GetHostByName') its DNS server. The DNS client uses a DNS resolver to locate its DNS server. If the DNS server is not authoritative for the destination domain, or if the DNS server doesn't have the information in its cache, it will not be able to answer the client's query immediately. Instead, the DNS server will either act as a DNS forwarder or will issue a recursive query. As an aside, if a DNS client receives a non-authoritative response from a DNS server, that implies that the DNS server has found the answer in its cache, instead of contacting the appropriate authoritative DNS server for that domain.

A DNS forwarder will forward the query to a second DNS server higher up in the tree of servers. This ends the contact between the DNS client and the first DNS server. Alternatively, if recursive querying is allowed, and it usually is, the DNS server will ask a root name server for the IP address of a host that's authoritative for the destination domain, and will then contact the authoritative server and report back to the DNS client. So a recursive query is one where a server issues one or more queries to answer another query.

The DNS server data files can also be queried to provide information about many RRs (Resource Records) in a domain by setting the appropriate type: NS (name server), RP (responsible person), MX (mail exchange), and CNAME (canonical or official name). 'Nslookup' is a DNS tool that is standard across most networks, while 'Dig' is a UNIX utility for querying DNS. MS Windows NT has the Sam Spade tool.

DNS Spoofing involves forcing a DNS client to make a query to an impostor server, and then tricking the client by sending a wrong answer from the impostor server. Three ways to carry out a DNS Spoofing attack are described below: 1. Spoofing the DNS responses 2. DNS cache poisoning 3. Breaking into the platform

1. Spoofing the DNS responses

Attackers can use the recursive mechanism described above to their own advantage, by predicting the request that a DNS server will send out, and replying with false information before the real reply arrives. Each DNS packet has an associated 16-bit ID number that DNS servers use to determine what the original query was. In the case of BIND, the prevalent DNS server software, this number increases by 1 for each query, making the request easier to predict. This has been fixed in the later versions of BIND, where DNS packets are assigned random numbers. This emphasises the importance of running the latest version of BIND (v9).

The impact of providing false host name and mapping information, is that the attacker can then misdirect name resolution mapping, while exposing network data to the threat of capture, inspection, and potential corruption. DNS involves a high trust relationship between client and server, and it is this trust that makes DNS vulnerable to spoofing. A cryptographic authentication mechanism would solve this problem, and one is discussed later in this paper.

To test whether a DNS Server is vulnerable to this DNS Spoofing attack you can send queries to the target name server, assuming that its traffic will flow somewhere over your network link. You can then determine (by analysing the queries), whether or not it is possible to guess the next DNS query ID number of a DNS query packet. If the DNS queries IDs are predictable, you can assume that it is possible to poison the server's cache with invalid data.

2. DNS cache poisoning

After recursive querying, a second DNS vulnerability lies with DNS caching. DNS servers cache all local zone files (hints file, and information for all zones the DNS server is authoritative for) and the results of all recursive queries they've performed since their last start-up, to save time should they receive a similar query again. The length of time that recursive query results are held in the DNS cache (TTL - time to live) is configurable. The default is for a RR (resource record) to inherit the TTL of the zone (Name domain) it is in.

DNS cache poisoning involves sending a DNS server incorrect mapping information with a high TTL. The next time the DNS server is queried, it will reply with the incorrect information. "DNS cache poisoning, where malicious or misleading data received from a remote name server is saved (cached) by another name server. This "bad" data is then made available to programs that request the cached data through the client interface" [5]. It is possible to limit exposure to this DNS cache poisoning attack by reducing the time that information is stored in the cache (the TTL), but this will have a negative impact on the server's performance.

A common implementation of DNS is the open source software BIND (Berkeley Internet Name Daemon) that powers most DNS servers. Many vulnerabilities have been found in

BIND [2], so it is important to ensure that the latest version of BIND is running. BIND 9 was released in September 2000, and its newest features include support for IPv6, DNSSEC and multiprocessor systems.

New DNS standards have closed the DNS cache poisoning hole by establishing discrete cache update security configurations that specify exactly which servers have the authority to provide updates [6].

3. Break into the platform

A third DNS spoofing attack involves breaking into the target network's DNS server. An example of this attack is the buffer overflow vulnerabilities of earlier BIND versions, which allowed root access to attackers. Once an attacker has control of the underlying DNS platform, he has control of the network environment.

To aid troubleshooting and monitoring, it is useful to know that DNS communication uses both the TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) protocols. DNS queries are handled on UDP port 53, while DNS zone transfers are handled on TCP port 53. Most firewalls will be configured to block TCP port 53 to prevent zone transfers.

A Zone (name domain) transfer is the transfer of the DNS database to a secondary server. It allows name servers that are authoritative for the same domain to stay in sync with each other [4]. DNS servers should be configured to allow zone transfers between primary and secondary DNS servers only, because the information in a zone transfer would be very useful to an attacker, such as the IP addresses of important hosts. Zone transfer attempts are often the first indication that a network is being probed. To attempt a zone transfer of a domain, issue the following command 'ls –d <domain name>'.

One way of preventing the damage that unauthorised zone transfers can cause, is to use a split-DNS system. This involves setting up an internal DNS server (containing RRs [Resource Records] for all necessary hosts, e.g. an internal mail exchange (MX), and an internal name server (NS). Then an external DNS server is set-up containing just the information needed by external hosts, such as an SMTP gateway, and an external NS. Although most mail servers can now handle SMTP mail as well (both MS Outlook and IBM's Lotus Notes have SMTP gateways), it is more secure to have a separate machine for incoming SMTP mail. Then if this external mail exchange is compromised, the attacker will not automatically have access to the internal mail system.

The future of DNS

DNS is vulnerable to spoofing because of the absence of authentication. DNSSEC is a new security mechanism that plugs a hole in the Internet's DNS by allowing Web sites to verify their domain names and corresponding IP addresses using digital signatures and public-key encryption [1]. This means that when a DNS client receives an answer to its query, it can verify that the reply came from an authorised source. Although DNSSEC is already included in BIND 9, and although it will be bundled with many operating systems, there are some barriers to its use.

DNSSEC will require more powerful hardware, more bandwidth, and will mean more work for systems administrators, and it will also require changes to every DNS server (the Internet's root and top-level Domain Servers, and end-users' local DNS servers). Until these changes have been made, one cannot be sure whether a Web site does not offer DNSSEC, or whether traffic is being diverted from the legitimate Web site.

REFERENCES

[1] Marsan, Carolyn. "DNS security upgrade promises a safer 'Net" 16 October 2000. URL:http://www.nwfusion.com/news/2000/1016dnsec.html, Accessed 6 November 2000.

[2] SANS Institute URL:http://www.sans.org/topten.htm, Accessed 6 November 2000.

[3] Network solutions: named.root is a list of the official root name servers. URL:ftp://ftp.rs.internic.net/domain, Accessed 6 November 2000.

[4] Bauer, Michael. "Securing DNS and BIND", Linux Journal, October 2000.

[5] CERT URL:http://CA-97.22.bind, May 26 1998. Accessed 6 November 2000.

[6] SecuriTeam, "WindowsNT DNS". URL:http://www.securiteam.com, Accessed 6 November 2000.