DNS Open Recursion

Jerry Martin, Team Cymru
There continues to be a significant Internet threat due to the domain name system (DNS) open recursion capability that is enabled by default in many DNS implementations. The DNS is the component of network infrastructure that maps human-readable labels (names) to Internet addresses and other resources. The two potential threats are:

- Cache poisoning which occurs when malicious or misleading data finds its way into a DNS caching server. The bad data is then made available to programs running on workstations (e.g., web browsers and email applications) that request the cached data. These applications are then redirected to send data to compromised hosts or blackholes.

- Distributed denial of service (DoS) attacks could take out most any Internet site including generic top level domain (gTLD) name servers (e.g., .com, .net, .org). An attacker could launch a source address spoofed stream of queries, which could result in an amplified reflection attack directed back at the spoofed victim address.

Either of these attacks could allow someone to dramatically affect the ability to reach a particular site or query for a domain name. Without DNS servers, Internet activity on the affected domain comes to a virtual halt. It is absolutely critical to protect domain servers, be they gTLDs or secondary domains (e.g., a critical infrastructure’s domain server) or any public-facing DNS server as quickly as possible before this vulnerability becomes more widespread.

All organizations should determine if their domain servers allow external recursion and evaluate the requirement to allow recursive queries on their internal name servers. A single solution (e.g., script/fix/patch) is not possible due to the immense number of internal/external DNS configurations. While on the surface, this appears a daunting task, it is a rather simple procedure that should take system administrators an hour or two per domain server to accomplish.

The following courses of action are recommended to protect name servers:

- The system administrator will restrict recursive lookups only to the authoritative domains served by the DNS server and will not allow world recursive lookups for the server. This method will help ensure the DNS server’s cache does not become poisoned for another domain and that this server does not participate in reflection attacks.

- Log checking scripts that monitor the overall activity of the name server and look for DNS lookups from outside the domain will alert the administrator to an outsider possibly looking for a DNS server vulnerable to cache poisoning.

- For those organizations operating internal and external name servers, authoritative and recursive service should be separated. Basically, two separate DNS server types are maintained, with two separate sets of data. Internal DNS servers should only accept requests from internal hosts. The external DNS server only accepts queries for authoritative names that the DNS server maintains.

Root DNS servers do not allow this recursion feature, top-level domain (TLD) servers generally do not allow recursion. Configurations for servers below the TLDs are determined by each administrator or user, but in our experience open recursion is widely available on the Internet. This
A succinct description of the problem is further explained as documented by the CERT/CC in VU109475.

Extensive private sector research shows many name servers are configured to accept and perform recursive queries for anyone. This means that anyone can launch a query to a name server for any resource record (RR). Below is an example using a .com name server.

dig +recurs @x.y.com www.ebay.com

;<<>>dig 9.1.0 <<>> +recurs @x.y.com www.ebay.com

;;global options: printcmd

;;got answer:

;;>>>header<<<-opcode: query, status: noerror, id:63235
;;flags: QR RD RA; query: 1, answer: 3, authority: 6, additional:6

;; question section:
;www.ebay.com IN A

;; answer section:
pages.ebay.com 1195 IN A 216.33.156.119
pages.ebay.com 1195 IN A 216.32.120.133

;; authority section:
ebay.com. 3595 IN NS NS2.exodus.net.
ebay.com. 3595 IN NS sca02.sec.dns.exodus.net.
ebay.com. 3595 IN NS sca03.sec.dns.exodus.net.
ebay.com. 3595 IN NS NS.exodus.net.

;; additional section:
NS2.exodus.net. 172791 IN A 07.82.198.150
pacifier.com. 172791 IN A 199.2.117.66
sca02.sec.dns.exodus.net. 899 IN A 209.1.235.120
sca03.sec.dns.exodus.net. 899 IN A 16.32.126.150
crocodile.ebay.com. 172791 IN A 216.32.120.21
NS.exodus.net. 172791 IN A 206.79.230.10

;; query time: 213 msec
;; server: x.x.x.x#53(x.y.com)
;; when: sat oct 29 13:57:17 2005
;; msg size rcvd: 318
This host has just looked up www.ebay.com on the behalf of a third party. It has collected significant additional information as well. Had the query pointed to a name server controlled by a hostile party, the cache could have been poisoned with RRs of the attacker’s choosing. A cache entry for the MX (mail exchanger, effectively the mail relay host for a host or domain) paypal.com could have been created. This would yield significant important data for any one of a number of miscreants.

While cache poisoning is certainly one risk, it can be mitigated by running a recent version of the DNS software on the name server. However, allowing recursion, even when cache poisoning is less likely, leaves the name server open to other abuse. If a name server will respond to any queries from all sources, an attacker can flood the name server with repeated queries. These queries could be quite intense and crafted to avoid using cached data.

A list of potential targets may be obtained by issuing a single command. The second step is to determine if the particular domain permits recursive queries which, again, is done with a single command. An “RA” entry in the “Flags:” section generally indicates when recursion is allowed. Additionally, the queries launched against this name server do not need to be legitimate. A steady stream of unique, bogus queries will work equally as well. The attacking platforms do not need to receive a response to these queries, which use UDP, thus source address spoofing works well.