DOMAIN NAME SYSTEM ATTACKS

DNS

We can access webpages using URLs thanks to the Domain Name System (DNS). DNS enables computers to identify the IP address that corresponds to a given domain name, enabling us to connect to websites by name rather than having to manually remember a list of IP addresses.

The specifics of how DNS functions may fill numerous articles, but the gist is that DNS operates through a hierarchy-based collection of DNS servers.

Domain Hijacking

If you want a domain name, you must register it with a domain registrar and supply an IP address. After that, the domain registrar will inform the DNS that the domain is now linked to the specified IP address. By using social engineering, hacking, or simply snatching a domain that someone failed to renew, an attacker can acquire control of a domain name without the owner's knowledge.

The domain registrar and occasionally the domain owner are the targets of domain hijacking, not the DNS.

The level of public confidence in a domain is referred to as its domain reputation. Even if the rightful owner can reclaim control, a compromised domain may be used for spam and scams, harming its reputation and leading to the name being blacklisted.

URL Redirection

HTTP requests have a capability called URL Redirection that enables requests to be forwarded to a different page. For instance, you might be led to a login page if you try to see a page of a website that requires you to log in. Although there are many good purposes for this, it can also be employed to deceive victims into going somewhere they didn't intend to. This type of redirection is often only useful if the attacker has access to a website that the target wants to visit, but it can be used in conjunction with other strategies like phishing.

Man in the Middle

The first method an attacker can use to do this is to pose as a local network DNS server. They can act as a DNS server by exploiting ARP poisoning, and they can reply to DNS requests with any IP address they choose.

DNS Poisoning

By interfering with DNS and making domain names point to the incorrect IP address, an attacker can also direct users to websites they didn't plan to visit.

Client Cache Poisoning

The HOSTS file was a method of instructing your computer to associate domain names and IP addresses in the pre-DNS era. This is a file that contains a list of domain names and IP addresses and is kept on your computer. Most operating systems still recognize and respect the HOSTS file, which enables manual association of domain names and IP addresses on computers.

You can probably guess where this is going: If an attacker can add a malicious entry to the HOSTS file on a target computer, that computer will use that entry's IP address to resolve the associated domain name rather than the one provided by DNS. The term "DNS Client Cache Poisoning" describes this.

Server Cache Poisoning

Server cache poisoning targets DNS servers, whereas client cache poisoning targets a client. Not every domain is "known" to every DNS server. DNS servers maintain caches of domains and IP addresses. When they get a request, they first check their cache and, if necessary, request the information from a server further up in the hierarchy.

The DNS server will give a fictitious response to anybody who requests a compromised domain if an attacker can enter malicious data into this cache, at least until the cache is updated. Like the ARP poisoning technique, but on a much bigger scale, this is frequently performed by a mix of Denial-of-Service and impersonation.

Conclusion

Being able to access resources via URLs is incredibly practical for us as humans, but the infrastructure we employ to make this practicality possible could be a target for attackers. Unsuspecting victims could be redirected to a location of the attacker's choice without the victim noticing anything is wrong if the attacker can disrupt domain-name resolution.