How Name Server Checks Protect Your Network Against DNS Tunneling

Posted on January 28, 2020

Being a sort of open phonebook of the Internet, the Domain Name System (DNS) can be a corporate network's weakest link. The main problem lies in how it works. As a way to ensure that devices communicate correctly over the Internet, DNS servers map IP addresses to domains in response to user queries.

More specifically, when a user searches for a domain name on their browser, the browser sends a query to the stub resolver, an operating system component, before querying the local name server.

This local name server, which is usually recursive (but can also be iterative), then takes action to fulfill the said request. It will either answer back with a cached response or forward the query to root servers, a top-level domain (TLD) name server, and then to an authoritative name server. The authoritative server holds all of a domain's DNS records. One of those records is the A record, which contains the IP address of the queried domain and tells the authoritative server to respond. More information regarding this process can be found in our DNS primer.

Unfortunately, firewalls and web proxies don't necessarily monitor these queries, making it easy for attackers to send forged or malformed requests instead. With that in mind, a common form of abuse against the DNS protocol is DNS tunneling. This article takes a look at what the attack entails and how Reverse NS API can lead to its detection.

How DNS Tunneling Works

The DNS system uses the User Datagram Protocol (UDP) port 53 to broadcast and receive requests. UDP usually has a limit of 512 bytes per packet. As long as the packets don't exceed that size, no alarms are raised.

This UDP limitation makes it convenient for cybercriminals to move data through a covert channel and into their command-and-control (C&C) server. Attackers simply need to break down files into smaller pieces to go undetected as they pass through the so-called "tunnel."

However, in the event of an advanced persistent threat (APT), the malicious payload is typically delivered before a DNS tunnel is set up. Here's how DNS tunneling typically goes:

- An attacker does reconnaissance on a host to find out if it allows queries to any DNS server. He then delivers the malicious payload through a spam or phishing email.
- The payload installs the DNS tunneling tool on the victim's computer. At the same time, the attacker installs the same DNS tunneling tool to an online server he/she can control.
- This installation process looks like a common DNS query or response in a packet analyzer. A query for a TXT record is sent to the C&C server, which contains the victim's computer's name or serial number. The said data is encoded using an undetectable algorithm. A sample query looks like this: installed. [computer info and serial number].attackersdomain[.]com. It represents the installation beacon, which informs the attacker that the payload is now operational.
- The same process is replicated during data exfiltration. Each bit of sensitive information is encoded as well. It also resembles the previous installation query in syntax, as seen in this example: newdata. [payment card info encoded in base32].attackersdomain[.]com.
- The response is a TXT record that contains a command that instructs the payload to collect additional information from specific folders. It may also allow the attacker to open and run programs on the victim's computer remotely.

How Reverse NS API Boosts Your Network Security

Indeed, the structural weakness of the DNS protocol makes it easy prey for enterprising adversaries. However, there are ways to bolster network protection with the use of threat intelligence. Reverse NS API, for instance, can help security engineers:

- **Monitor DNS activities to distinguish legitimate queries from junk traffic based on near-real-time threat data:**

DNS tunneling may involve thousands of queries to a single parent domain, or various suspicious-looking query strings and response packets. You can use Reverse NS API to conduct **name server checks** on unusual domains from your server logs so that you can restrict inbound and outbound traffic coming from and going to them.

- **Perform user and entity behavior analytics:** You can identify anomalous user or system behaviors by evaluating log data with Reverse NS API. The tool allows engineers to **check name servers** for a given domain and find out if it is linked to a known C&C server or a malware campaign.

- **Do DNS forensic analysis:** Reverse NS API can aid in passive reconnaissance and DNS audits. The tool provides cyber investigators with a quick overview of all connected domains and subdomains on the same name server, including when they first appeared online or were last changed. The forensic analysis could otherwise take weeks or months if individual WHOIS records are vetted manually.

The DNS is responsible for how the Internet works. Because of this, cybersecurity professionals have to find a workaround for its intrinsic flaws to effectively shield their networks against cyber risks. Reverse NS API is a handy tool for analysts to anticipate or validate potential threats detected by their security systems.