Notes:

### A Poor Man's DNS Anomaly Detection Script

<https://isc.sans.edu/diary.html?storyid=13918>

Reading the MS DNS log

<http://www.computerperformance.co.uk/w2k3/services/DNS_debug_logging.htm>

Understanding the windows event log entries for DNS

[http://technet.microsoft.com/en-us/library/cc785424%28v=ws.10%29.aspx](http://technet.microsoft.com/en-us/library/cc785424(v=ws.10).aspx)

Viewing the options for DNS debug config

[http://technet.microsoft.com/en-us/library/cc776361%28v=ws.10%29.aspx](http://technet.microsoft.com/en-us/library/cc776361(v=ws.10).aspx)

# For Google, DNS log analysis essential in Aurora attack investigation

<http://searchsecurity.techtarget.com/news/1514965/For-Google-DNS-log-analysis-essential-in-Aurora-attack-investigation>

Concentrated DNS queries in a specific place represented invasive operations, she said. Most traffic mainly reaches out to common sites. A warning sign is when traffic is detected going to a new website that was recently registered and no one had visited before.

"DNS query logs may be the only method you have to find new generations of malware," she said. "The adversary will need to reach other systems to install that malware. We often look for the big [anomalies], but we have to monitor for the subtle too."

What are the top DNS lookups?

What are the new DNS lookups?

What are the bottom lookups?

What are the bottom country TLDs?

Can even see suggestions of sessions in stats even though you have not other data. i.e. series of cruise sites in a row.

Analyzing Statistical Anomalies in DNS Queries

<https://discussions.nessus.org/thread/5298>

“The logs above indicate that for the hosts in question, they normally have 6 to 12 DNS lookups during this hour of the day, however today, these hosts have made 100s of queries.”

Statistical analysis can find where there is a spike in queries by a certain computer or computers to a certain site, a group of sites or a new site. This could be normal (certain computers scheduled to retrieve something at a certain time), odd but kosher (forgotten update script) or malicious (malware propagating).

Check IP of machines requesting DNS from server. Should only be from your network, right?

DNS replies that have an RFC1918 address external to your network

<http://isc.sans.edu/diary/Got+Packets+Odd+duplicate+DNS+replies+from+10+x+IP+Addresses/13222>

Also

“As a side note, initially the DNS protocol specifically allowed for replies to arrive from an IP address different then the one the query was sent to:

"Some name servers send their responses from different addresses than the one used to receive the query. That is, a resolver cannot rely that a response will come from the same address which it sent the corresponding query to. This name server bug is typically encountered in UNIX systems." (RFC1035)

However, later in RFC2181, this requirement was removed:

"Most, if not all, DNS clients, expect the address from which a reply is received to be the same address as that to which the query  eliciting the reply was sent.  This is true for servers acting as clients for the purposes of recursive query resolution, as well as simple resolver clients.  The address, along with the identifier (ID) in the reply is used for disambiguating replies, and filtering  spurious responses.  This may, or may not, have been intended when the DNS was designed, but is now a fact of life." (RFC2181)

But we are NOT looking for responses that are coming from the wrong source, but duplicate responses. Once from the correct and once from the incorrect address.”

<http://www.securityweek.com/do-you-know-what-your-dns-resolver-doing-right-now>

Here are some rules of thumb to help with simple triage:

1) Look for long, randomized hostname queries sent to the same or small subset of domains. This one is a no-brainer, and you can start by just looking for extremely long hostnames being resolved.

2) Look for TXT requests and of course TXT responses that contain large amounts of gibberish.

3) Watch for “beaconing” behavior—the same hostnames (that aren’t in the Alexa list) being pinged regularly.

4) Look for domains in oddball TLDs and subdomain resellers—especially ones known to harbor many malicious registrations like .su, .ru, and .tk.—particularly if you don’t do business in those countries normally.

5) Look for patterns of recursive DNS usage during odd hours. Most machines don’t need to look up new domain names when people aren’t surfing or doing other “randomizing” types of activities that humans do, so lots of non-cached hostnames shouldn’t be hit late at night.

6) Examine the history and hosting of the nameservers being used by domains queried and the reputation those nameservers have. If they are located in unusual countries, are being supported by fast flux, are brand new domains, or are in hosting facilities with abuse problems, the traffic needs more scrutiny.

<http://www.securityweek.com/day-internet-will-break-millions>

<http://www.securityweek.com/dnschanger-wake-call-enterprise-and-government-dns-resolver-management>

<http://isc.sans.edu/diary.html?storyid=4831>

### **Watching those DNS logs**

From the attack details that have so far been disclosed, scenario (2) involves a high number of queries for bogus names in the domain that you own. Your authoritative DNS server will respond with "NXDOMAIN" (no such domain) to all these requests, but every such query provides the attacker with a chance to inject a bogus reply to poison the querying resolver of your client's ISP.

Emergingthreats.net has a Snort rule (sid:2008470) to catch an excessive number of NXDOMAIN replies received by your resolver. This covers scenario (1). Scenario (2) would require a way to watch for a high number NXDOMAIN answers sent by your own authoritative DNS server. A quick check we made turned out that neither BIND9 nor MS-DNS seem to offer any easy way to log NXDOMAIN answers of your DNS server without going into debug-dump-it-all mode. The statistics collected by BIND keep track of sent NXDOMAIN records in the "SNXD" counter, but that's about it.

<http://pentestmonkey.net/blog/mssql-dns>

<http://blog.cloudflare.com/deep-inside-a-dns-amplification-ddos-attack>

<http://blog.rootshell.be/2011/11/03/show-me-your-dns-logs-ill-learn-about-you/>

* Ubuntu looks to be the preferred Linux distribution due to the huge amount of requests to ntp.ubuntu.com.
* Gmail is a common e-mail platform but lot of people manage their emails via IMAP (imap.gmail.com).
* ocsp.verisign.com / ocsp.thawte.com are quite well used (“Online Certificate Status Protocol“).
* Bittorrent remains a classic tool to search for content.
* WordPress remains a top platform for security bloggers.
* WPAD (“Web Proxy Autodiscovery Protocol“) is a nice way to detect from where are coming your visitors. Most browsers try to resolve “wpad.company.tld” to configure their proxy settings.
* Special mention to Peter from [corelan.be](http://www.corelan.be/), who was resolved quite often!

Dumbella has commercialized DNS reputation

<https://blog.damballa.com/archives/1834>

NOTE: make a note of where you got your blacklists and why a domain is on a particular blacklist, so if you get a hit on a domain, you know how you should react. Is it just a warning, require a little or a lot investigation, or an all out immediate response?

FEATURE: Every time the script runs, check the time on each DNS server to make sure the times are synced with each other and with whatever time you are using on the Enterprise.

<http://ananalyticalapproach.blogspot.com/2012/02/following-trail-of-dns.html>

### Following The Trail Of DNS

DNS functions in a "nested" fashion within an enterprise.  Many enterprises use Active Directory Domain Controllers, or some other nested means by which to resolve domain names.  This creates multiple layers of "NATing"/"client abstraction" issues that both necessitates multiple layers of DNS logging (often redundant), and creates the need to "following the trail" of DNS transactions through the network.

Most DNS logging doesn't "match up" or sessionize requests with their corresponding responses (or lack thereof).  This leads to 2 problems.  First, nearly twice as much data is generated as is really necessary.  Instead of just one line in the logging with the DNS request and its corresponding response(s) (or lack thereof), there are two or more lines in the logging.  Most of the information in the additional lines is redundant, with only a small amount of that information being new and/or adding value.

Suggests: To me, a sessionized, intelligently aggregated DNS logging ability could produce a compact, efficient data source for retention, analysis, and monitoring within an enterprise.

I think that centralized logging of the different levels of DNS servers is the better way to go and that gets integrated into/with all the other logs

<http://ananalyticalapproach.blogspot.com/2011/10/passive-dns.html>

### Passive DNS

Collecting passive DNS data on your network can produce a data source of extremely high analytical value.  Because the data collection is passive, it has little (or no) impact to operations.  What is so interesting and valuable about passive DNS data you ask?  Well, for starters, it records what domain names were assigned to what IP addresses *at the time those domain names were requested.*  This makes all sorts of interesting analysis possible.  For example, which domain names were requested that point to IANA reserved (e.g., 192.168.0.0/16) or downright silly (e.g., 111.111.111.111) IP addresses?  This can be an indicator of compromise, as malware authors will often park their callback or second stage domains at these types of IP addresses.  Additionally, all the standard analysis that one would do on DNS logs can be done on passive DNS data as well.  For example, which domain names have been changing IP addresses frequently or have extremely short TTLs (i.e., fast flux)?  Or, as another example, which domain names have been requested periodically, or in a pattern more typical of a machine than a human-being?

DNSCAT – netcat over DNS

<http://www.skullsecurity.org/wiki/index.php/Dnscat>

Future Feature: Re -Turn on full logging and analyze returns. Find way to match returns to queries.

Three FEATURES (from Passive DNS blog): **1/** For example, which domain names have been changing IP addresses frequently or **2/** have extremely short TTLs (i.e., fast flux)?  **3/** Or, as another example, which domain names have been requested periodically, or in a pattern more typical of a machine than a human-being?