# The Domain Name Service as an IDS

How DNS can be used for detecting and monitoring badware in a network

by

Antoine Schonewille, talitwan@os3.nl
Dirk-Jan van Helmond, dirkjan@os3.nl

as a Research Project for the Master System- and
Network Engineering at the University of Amsterdam

February 5, 2006


# Abstract

SURFnet is looking for technologies to expand the ways they can detect network traffic anomalies like botnets. Since bots started using domain names for connection with their controller, tracking and removing them has become a hard task. This research is a first glance at the usability of DNS traffic and logs for detection of this malicious network activity. Detection of bots is possible by DNS information gathered from the network by placing counters and triggers on specific events in the data analysis. In combination with NetFlow information and IP addresses of known infected systems, detection of bots of network anomalies can be made visible. Also the behavior of a bot can be documented and additional information can be gathering about the bot. Using DNS data as a supplement to the existing detection systems can give more insight in the suspicious network traffic. With some future research, this information can be used to compile a case against particular types of bot or spyware and help dismantling a remote controlled infrastructure as a whole.


## Note

We started this research project with the question if the Passive DNS Software of Florian Weimer was useful for bot detection. We immediately found out that the sensor of the Passive DNS Software strips the source address from the collected data for privacy reasons, making this software not useful at all for our purpose. We deviated from the Research Plan (Plan van Aanpak) and took a more general approach to the question; "Is gathered DNS traffic usable for badware detection".

## Research Summary

In our opinion, DNS analysis still has some drawbacks when it comes to primal detection of worms, bots or other badware. The number of false positives is high and the data analysis is very cpu-intensive. The resolver cache in the client clouds the number of connections a host initiates. A query is only done once in a period of time, so the number of connections can not be deducted from the logs. This leads to the fact that DNS statistics just are not conclusive enough about the behavior of a system for the possibility to mark a particular as infected with 100% certainty. This does not mean that DNS can not be used for detection of this kind of traffic.

Some interesting results have been found from the data that was gathered. The most important result of this research is that detection and monitoring badware trough DNS can be achieved, but the results are just not exact science. The combined results of several types of analysis are often good indicators that can indicate certain systems as suspicious. Then further investigation can make clear if this indicator is a false positive or a real threat. The most usable indicators that were analyzed are:

- Matching DNS lookups against known bad domain names
- Top 10 lists with domain names queried and systems who performed the queries
- Monitoring the use of other resolvers than the ones provided
- Watching after queries with non-regular query type (MX, AXFR, IXFR)

The best results are gathered when the monitoring of DNS is joined with the information of several other warning systems. Monitoring an infected system and cross referencing the results with Passive DNS Database and nfdump results gives very good indicators of misbehaving systems on the network. Future research should explore the possibility of a way to connecting these systems together and automatically generate reports about the results from each system.

In the event that a system misbehaves and it shows from the logs, a case could be made in combination with the NetFlow information of the activities of the particular system and the administrator of the infected system could be informed. Getting a detailed report of system (mis-)behavior is easier to investigate than an alert with an IP address and a timestamp. It also gives some assurance to an external administrator that the system is really misbehaving, making the priority of disinfecting the system higher.

# Contents

# 1 Introduction

As part of the Master study in the field of System- and Network Engineering (OS3)[1], at the University of Amsterdam, the students will perform a research project on behalf of SURFnet[2]. The research project will have an emphasis on the subject of the possibilities of using the domain name systems as an intrusion detection system.

## 1.1 Introduction to SURFnet

SURFnet is a network provider for universities, colleges and other educational and research institutions. Their core business is offering high-bandwidth network-layer connectivity. Customers themselves have to take care of the services needed for accessing the Internet, e.g. DHCP and DNS.

For the purpose of network availability, SURFnet wants to have a grip on service-disrupting traffic to protect their own network and the Internet community at large. To accomplish this, SURFnet monitors their network traffic for malicious activities. This detection is mainly done by analyzing samples of Netflow information. Unfortunately, results of this analysis only reveal session information about the disruption. There is a need for more detailed data to identify the cause of this traffic. To gather more detailed information, SURFnet is currently an IDS sensor[5] network to have more insight in this type of traffic. Other less conventional methods of detection are now under consideration. The monitoring the domain name queries is one of them.

## 1.2 The issues with DNS

Earlier botnets contacted their controller by a static IP that was hardcoded in the source of the bot. When the IP address was retrieved from the source through reverse engineering or monitoring a bots behavior, that specific address could be black-holed and the network was safe again. Bot programmers had to find ways to circumvent these issues, and they did. Nowadays, bots have domain names coded in them which they can resolve to connect to a controller. This method makes the connection transparent, giving room for the controller to move between hosts by changing A records. A problem that is often faced when trying to find a domain name to an IP address, is the fact that not for every A record, a matching PTR record in the reverse lookup zone exists. So we can't find out which domain names a client queries when he contacts a controller. Fortunately, this issue is partially solved by the Passive DNS Replication Project. The Passive DNS project builds a database that caches answers of queries for A records. This way matching records can be queried. Sadly, this still leaves the problem of detecting the bots.

## 1.3 Current Warning Systems

Currently, SURFnet has two warning systems in operation on their network.

**Network Emergency Responder and Detector** N.E.R.D. [3] is an IDS developed by TNO for SURFnet. N.E.R.D. collects NetFlow data, to which statistical analysis is applied for anomaly detection, e.g. DoS attacks and massive portscans. Backtracking the source ip addresses of these attacks

usually leads to the bot infected systems. But detection is a reactive based approach, and means that you are too late, since your resources have already been misused, and the damage could already have occurred.

**Bad Hosts Lists combined with IP Flow information** With regular interval SURFnet receives bad hosts lists that contain ip addresses and tcp ports whereof is known that they service irc bot controllers. These IP address and tcp ports are then matched to sampled IP flow data collected from routers in the SURFnet infrastructure. This is done by a tool called *nfdump*[4] by Peter Haag of Switch. When they match to a flow, the system that initiated the connection is highly likely to be infected. Now these systems can be cleaned before they are used in an attack. The advantage is that action can be taken on infection before resources are misused, thus diverting a possible attack. This is a proactive based approach, but it still requires you to have knowledge of bad ip addresses before you can act. And you have to depend on external sources for this information.

# 2 Research Goals

Detection of bots is hard when they are not yet active. If you know the IP addresses and TCP port on which a controller resides, you can extrapolate this knowledge from Netflow statistics, but this requires you to have knowledge of bot controllers. When these botnets are instructed to contact another controller, you are unable to detect newly infected machines.

## 2.1 Bot detection

To keep a grasp on badware clients and servers exchanging them have to be detected. The first goal is to find ways to detect clients in the network by means of DNS statistics, that are infected with badware. A way this could be performed is by monitoring the domain naming service and using these result to detect suspicious behavior. An automated system or network management could inform the systems administrator of the target network to take action against the infection.

## 2.2 Behavior Monitoring

Our second goal is to monitor the behavior of systems that are positively identified some kind of remote controlled malware by gathering the DNS lookups they perform. When a couple of infected systems start querying another domain name at a regular interval, further investigation can be stated against the domain name to see if there are services active on the destination system, that are not legit. If this is the case, the domain name involved could be shared between CERTs to inform others of this malicious activity. In their turn, other CERTs can cross reference the bad domain names with their DNS statistics.

# 3 Related Work

David Dagon has uttered the idea of DNS based detection of botnets in the presentation 'Botnet Detection and Response'[6] and John Kristoff[7] of the Northwest University talked about how to use DNS as a botnet mitigation tool, and what some results could be. Besides this, has been no scientific research done in the detection of bots and their behavior based on DNS data. Some projects use DNS in the process of detection, but none use DNS data as the base of intrusion detection. Other related research has been done by the following projects.

## 3.1 Passive DNS Replication

The Passive DNS Replication Project[8] is a project started by Florian Weimer of the University of Stuttgart, Germany that uses sensors placed at strategic places on the Internet that captures answers of DNS queries from clients. This information is used to build a database where IP addresses with the respective domain names from the query are stored. The goal is to make it possible to make reverse lookups with IP addresses for which no PTR records exist. This could be useful for detecting which domain names are used to contact a system on the Internet. An example could be when a user has control over a forward lookup zone and uses records from this zone for contact with systems he uses. Often, these people have no authority over the reverse lookup zone. This way, they can't make a PTR record for their system. Situations where this could happen is with virtual hosting as a reseller or user, and when zones are used to point A or MX records to IP addresses of customers with ADSL or Cable Internet connections.

## 3.2 Bad domain names (DNSWatch)

Norman Elton and Matt Keel of the College of William & Mary mention the possibility of detecting bots and worms by comparing lookups to a known list with bad domain names in their presentation 'A Discussion of Bot Networks'[10]. The way they propose to gather the list with bad domain names is trough DNSWatch[9], a PERL script by John Kristoff that parses DNS logs to detect the change of DNS resource records. This information could be used in combination with the information of hosts using a specific resource record.

# 4 Hypothesis

Our hypothesis is defined as follows: Systems, infected with badware (malware, bots or viruses) that make contact with external hosts by using the domain name service, give away information about themselves through the queries they perform. When these lookups are gathered, they could be used to pinpoint infected systems. With specific analysis, this could also reveal some information about the infection and the source.

# 5 Methodology

Our research project consisted of two phases. The first phase is finding out which method can be used best for gathering the needed data from the DNS system. The second phase consisted of analyzing this data for use of detection and monitoring.

## 5.1 Data collection methods

There are two main options of gathering the DNS data. The usage of query logging on DNS servers can be used to parse the performed DNS queries into a database. A second option is eavesdropping traffic to a name server in a live network, and forwarding requests to a collector.

**Parsing the gathered logs** of the used name servers into a database at specific intervals can be a way of retrieving the needed data. The parsing of logfiles is by far the least intrusive type of monitoring from a network point of view. No adaption of the network layer would be needed. A disadvantage is that when systems use other resolvers than the ones that are configured for monitoring, i.e. publicly accessible resolvers on the Internet, you would miss out on this data, which possibly could contain important information.


Figure 1: Parsing a DNS querylog into a database

**Eavesdropping a live network** is a more intrusive and also more intensive way of gathering DNS queries performed. To gather all possible queries, a sensor has to be placed in the traffic stream to the local DNS resolvers and on the Internet uplink. Because these connections usually bear large traffic streams, a switch that supports span ports and can filter packet flows is a prerequisite.

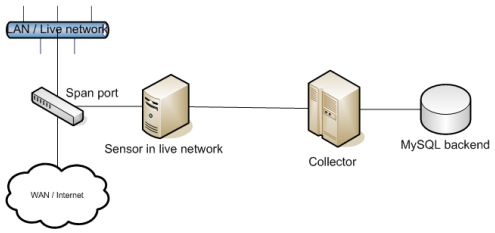


Figure 2: Eavesdropping the network for DNS queries and log into a database

## 5.2 Data analysis methods

The biggest problem is that the behavior of a bot is not documented somewhere. This means that it is not clear what queries are performed on which interval and on what basis. To get useful results, we have to find specific patterns in the logged data.

Analysis of the following can indicate uncommon activity. A host could be infected with a bot or be used for hacking purposes. Our project focused on the following types of analysis:

**Queries to resolve known bad domain names** Queries of machines can also be matched with a list of bad domains. This analysis is probably the most powerful, because of its immediate result.

**Top 10 statistics** Different Top 10 statistics can be a start point to detecting behavior that is out of line. Requests per hour per client, number of queries to a specific domain name and number of different name servers used. The systems at the top of these lists can be used for Top 10 statistics like number of requests per hour per query, number of name servers per client, etc. Abrupt changes in one of these top 10 ranks could be reported for further investigation.

**Monitor the queries of known infected machines** With the Netflow data from nfdump it is possible to detect connections from clients to machines that are on a blacklist[1]. The machines that initiate this connections are marked as suspicious. By watching their queries, the behavior of the bot can be documented, which can lead to detection of other bots in the network. The Passive DNS project could also be involved in these tactics.

**Query anomaly (new queried domains)** New domains, queried en mass by A or PTR requests are at least suspicious. This can mean that a new website became increasingly popular in one day, or something else is going on. By looking at that information, it is possible to detect a new (widespread) virus or botnet infection, but also results from the infection like DDoS attacks.

**Used resolver deviation** Large deviation in which resolver was consulted could be an indication of uncommon activity. Normal operating machines usually only send requests to resolvers, defined in their resolver list, configured either by hand or DHCP. Uncommon behavior could indicate malicious activities. If most contact is with root servers and with name servers that are authoritative for the requested domain,it can be the case that it is more likely that a consumer has a local DNS server that performs iterative lookups.

**Queries with uncommon qtype (MX/AXFR)** There are several queries you don't expect to find on an access layer network, i.e. many MX requests from end users can indicate a system that is abused as an open mail relay. This relay can than help to further propagate a virus or bot to other machines. Certain hack attempts can also be detected when many AXFR or IXFR requests are made to high profile domain name systems

[1] Current blacklists are retrieved from a closed security mailing list

i.e. hotmail.com or netscape.com. At the moment of writing, there are still many vulnerable BIND versions active. A scan may reveal the version of BIND followed by a hack attempt. Machines performing AXFR or IXFR requests are reason for further investigation.

**Baselining the DNS usage** Comparing more monitored networks, e.g. a clean and a dirty network to each other, can tell more about query behavior. The differences could give a lead to infected machines or bad domains. One drawback could be that the two compared networks have to be more alike. Different types of networks, e.g. an office network or a university campus network could be too different to be compared.

**Startup monitoring** Looking at the start up of a possibly infected machine could tell more about bots or botnets. If the DNS queries performed right after booting changes over time, this can reveal a fresh installed bot contacting its controller. This behavior can also appear when a machine was disconnected for some time and than reconnects to the Internet.

# 6 Data analysis results

All the anomaly detection was done by hand. Automation of most detection would be quite hard, because there is no real strategy to define that something is wrong, but for some behavior can be set traps.

## 6.1 Matching against known domain names

The match against a known bad host or domain list is easy to be done. (See appendix: B.1) Getting a list with up-to-date information of domain names is less trivial. Somewhat of a disadvantage of this approach is that it is rather hard to keep a list of bad domain names up-to-date. Possibilities are Fraudwatch[13] that is a company that is selling these lists, and there are some closed mailing lists like Gadi Evron's[14] lists and nsp-security[15] that maintain these lists. The domain names retrieved from bots in sandboxes or in mailing lists[2] were mostly of older botnets. So this had no usage to detect recent badware threats. Earlier threats were taken care of already with help from netflow information. The infected machines were put in quarantine and the owner was forced to clean things up before reconnection was possible. Though this way of handling issues works well, the probabilities of finding useful results became limited.

## 6.2 Top 10 requested queries lists

The first executed query to display a top requests list, showed some surprising results. (See appendix: B.2) Other than the expected requests like queries for google.com or msn.com, the top rank was often represented by unknown, unresolvable domains. After investigation, it appeared that these domains used to host a botnet controller, which was taken down a couple of months ago. The domain was not cached as negative. A conclusion from this can be that these requests were done by bots which used their own DNS resolver, since the resolver available in Windows also caches negative queries. In a way this can be used

[2] We didn't subscribe to the mailing lists of Gadi Evron or nsp-security

to detect systems that are still infected with older or rogue bots, detached from their controller.

## 6.3 Top 10 requesting clients lists

In our data, there was a big difference between the top 10 one on the list and the rest. The 20%/80% rule is a little bend here to the 1%/99% rule. 1% of the clients performed 99% of the requests. Further investigation showed that most of the top clients matched to the records in the top query lists. Also many clients in the top list were requesting domains that were not available. Just like the top queries list, this could indicate bot-precence, since Windows does negative caching for 4 or 24 hours[3].

## 6.4 Monitoring and Cross-referencing

Monitoring known infected systems and cross referencing them with nfdump data or the Passive DNS Database gave many information. When a client connects to a controller this can be detected with nfdump. If the DNS name was not cached, this would show a hit with the exact same timestamp with our DNS logs (See appendix: B.3). We would then use the domain name that it used to contact the controller to cross match this with other clients in our log. This showed results of other systems infected. Unfortunately they had all already been noticed by a nfdump run. When we matched the domain name in the Passive DNS Database, it revealed four other domains on the same IP address. We could match two other domains to other clients in our log, making contact with this suspicious website. These have also been noticed by a nfdump run. This showed that it is possible to detect other bots through DNS monitoring, once a lead is found. Also is requires a lot of work for something that is done faster with nfdump. It couldn't be concluded if this bot later started querying any of the other domains because our data was limited to about 14 days.

## 6.5 Used resolver deviation

The nameserver deviation research produced some interesting information. A rather large number of machines used other nameservers besides the suggested ones by the provider. When the regular name servers were filtered out, two results were found. A remarkable result was that about twenty clients used over a 1000 name servers. Closer inspection led to the discovery that most of the queries were done to root servers or authoritative name servers. We can conclude from this that it is highly likely that the customer has a resolving name server, performing iterative lookups. The second thing we found out was that a very large number of systems (around 15%) uses 1 to 4 name servers other than the ones provided by the ISP. We found out that many of these were to highly redundant name servers of high profile internet providers like AOL of Qwest. Most of the requests done to these servers were for commercial purpose like webads, adware and spyware. The developers of these software were probably worried of being black holed at the DNS server of the ISP. Monitoring the used number of name servers on a hourly basis can lead to the discovery of infected systems (See appendix: B.4).

[3]Depends on configuration

### 6.6 Query anomaly (new queried domains)

An interesting event we found was a specific host that was in the top ranks querying a domain name that was not resolvable anymore. At one moment this specified client disappeared from the top rank. When the cause was found, it appeared that this client stopped querying this domain, just after he had contact with an IP address of which we knew a controller resided. So we can conclude that systems receiving a command from a controller can in some times be detectable by DNS log information.

### 6.7 Queries with non-regular qtype

There were many lookups for other qtypes than the usual A, quad A and PTR. The qtypes that should require focus are large amount of MX queries and the AXFR/IXFR queries, which can be used for hacking or spamming. The most MX queries mainly originated at clients that have local SMTP servers. These servers perform MX lookups before forwarding a message. There was however an anomaly we found here. There were a couple of clients that queried hotmail.com for a MX resource record, just before resolving some domains from other than by the ISP provides name servers. We couldn't trace the domains back to bot controllers and haven't found out where the domains were used for. From the AXFR/IXFR queries performed, most of them were from just a couple of hosts. Since the AXFR queries were targeted at high profile name servers from cisco.com, hotmail.com and netscape.com (See appendix: B.5), and it is highly unlikely that a secondary for any of these domains resides in the VLAN that was monitored, it is likely that the queries were initiated by a hacker or a bot.

### 6.8 Baselining the DNS usage

Establishing a DNS baseline is hard. It becomes even harder if you can't manage the clients. The monitored network had infected systems from the moment we started monitoring. Therefor baselining the DNS usage was not a possibility at the moment. It should be possible to establish a practical baseline by filtering out many of the infected hosts, i.e. by removing the top 10 DNS clients and the top 10 performed queries. Further research in baselining was not performed.

### 6.9 Startup monitoring

Monitoring DNS queries after a client starts up can be an interesting indicator if it is in a managed environment where all the systems are installed with the same software. Deviations from standard behavior could lead to suspicion. Unfortunately, because we could not baseline our network or individual clients, startup monitoring was not a real possibility to investigate.

## 7 Research results

Detailed DNS analysis can give information about unwanted network activities. When these types of analysis are automated in an information system that could automatically generate a rapport of incidents. This could greatly improve the knowledge of badware on the network. In the research goals was pointed out

that both de detection and the gathering of additional information should be done through DNS statistics collection.

## 7.1 Most effective ways of detection

Both goals can be achieved through the use of DNS statistics. From results it showed that detection is never 100% accurate. Monitoring a suspicious system and cross referencing with Passive DNS Database and nfdump information could yield near 100% positive detection results and much better data gathered. The detection of infected systems in the network could be done by some of the following means:

- Matching DNS lookups against known bad domain names
- Top 10 lists with domain names queried and systems who performed the queries
- Monitoring the use of other resolvers than the ones provided
- Watching after queries with non-regular query type

Also we believe that in an environment where end systems are managed, baselining DNS usage and startup monitoring could lead to results.

## 7.2 Usability of the results

As shown in the 'Results Gathered' appendix, our results were mostly pointers to suspicious activity. Though we started with the goal of bot-detection, it shows that DNS statistics are very susceptible for false-positives and are not conclusive enough for marking bad hosts with a 100% certainty. The local cache on a client also caches queries for a period of time. The DNS logs are deprived from a lot of information this way. Besides that, the DNS detection was always behind on the detection based on NetFlow statistics. In the example in appendix B.3 showed that is was possible to detect new domain names from a previous infected systems, but the results are based on a lot of ifs and whens. You should have a match on the timestamp and there has to be information in de Passive DNS replication Database. Also the performed queries are very CPU intensive. It would be impossible to match all the found IP addresses and timestamps between the two systems. Especially the timestamp hit is hard to match. Sometimes there is a time delta between the results of nfdump and the DNS logs. Also some clients that are infected spawn over a 100 queries per second. Matching it to a specific hit in nfdump results becomes difficult then. Then there are the smart bots, that just avoid detection be performing a query earlier, and caching the result. Relying purely on DNS for intrusion detection is not advisable.

The use of DNS as an additional source of information about systems is much more usable. When a nfdump reports a system, it gives you an IP address and a timestamp. This information doesn't really show what is wrong with the system, just that it makes contact with known bad systems on the Internet. When the gathered DNS data can be used to find additional information about the systems behavior, an analysis of the thread of the system can be made more precise. For example, systems that make a lot of AXFR queries are less of

a thread to systems that contact the domain names of current or former bot controllers. Systems that use over 20 different name servers are more likely hosts to an infection than a system that does a hundred lookup in a minute. This detailed information can be of great value to the system administrator.

### 7.3 Accuracy of the results

The findings presented in this document are based upon results which depend on many different factors. First of all when looking at a network, one has to question to what extent this network is representative. Though two networks have been monitored, these may be too equal to each other. Also, the IP addresses of most of the machines on the two networks are configured by DHCP. It is therefore likely that a machine could have been assigned a different address after (re)booting what results in information shifts.

Another factor is the sensor, responsible for filtering the required DNS data out of the complete stream. Calculation showed that this machine was theoretical capable to perform the task. This was confirmed by watching the system performance of the Ethernet card and the CPU load. However, these numbers showed average like statistics. Taking the system utilization of an average of 65%, it was very probable that high traffic peaks resulted in missed DNS data.

## 8 Conclusion

Using the DNS system as an IDS is an attractive possibility. The data is relatively easy to gather by eavesdropping on an uplink, but analysis of the data is hard. Many of our results match the findings of John Kristoff's research[7]. The results showed that there were several matches between data gathered from nfdump results and the data gathered from the DNS logs. But nfdump was often earlier with detection. Possibly because DNS analysis had to be done by hand, and it was impossible to cross match every hit. Automation of this process could reveal results faster.

The real conclusion that can be drawn here is that detection purely based on DNS statistics alone is not a real viable possibility. Badware does also use the DNS system and DNS monitoring can surely aid in the detection of unwanted network traffic. Because the detection is not accurate to the extend of 100% positively pinpointing all known or new infections. Rather the DNS data should be used for gathering additional information about infected systems that were logged from a nfdump. This could in turn be cross referenced with other DNS monitoring systems like the Passive DNS Database to yield usable information about systems and behavior of badware, to which further investigation can be stated.

## 9 Future Research

Future Research on the subject of DNS as an IDS could provide a way to do some automatic risk analysis with information like the used number of name servers, large quantities of queries or uncommon query types automatically. When the risks generated by the behavior of a client succeeds a certain threshold, a report could be generated and send to the system administrator, which could take

actions to clean the system. Also typical behavior of a bot could be monitored to expand the knowledge of the way a bot acts. This information can be relevant to taking down a botnet as a whole.

An overview of interesting future research:

**Interfacing with external systems** like the nfdump NetFlow information or the Passive DNS Database could automate the reporting of results. An interface that is usable by some kind of central detecting, monitoring and reporting tool that uses all this information together to build a profile of suspicious systems would be of great value.

**Interface with a DHCP server** can help detect the movement of IP addresses through a network. This could clarify the results.

**Heuristics** can be used to detect 0-day infections of badware. Since performing heuristics is a very intensive and complex method of analysis, deleting unusable information from the database is also a subject that needs further exploring. Determining the trivia of the DNS data, is probably a research of its own.

**Baselining** DNS usage in a managed environment is also a subject that would surely result in a viable way of badware detection.

**Automated back-holing** by injecting a diverting resource records of bad domain names in the resolver cache can help quarantining systems from controllers, without interfering with the normal action of a user.

# A Specification of used Hardware and Software

The used DNS data was gathered by a sensor and a collector. The sensor in Eindhoven ran on an Asus S-presso S1-112 barebone that was equipped with a Pentium 4 processor on 3.4 GHz and onboard LAN. Only 256 Megs of RAM was installed what appeared to be more than enough. To be able to fetch the DNS queries of the line, an additional 1 GBit Ethernet card was installed. After the required data was extracted from the stream, it was send to the collector in an UDP packet. The collector, running on an older machine in Amsterdam was responsible for recording the retrieved DNS query traffic to a MySQL database. This way of gathering data allows multiple sensors to report to a single collector.

The sensor software was based on the Passive DNS sensor of Florian Weimer. The original software only collected the nameserver answer. The research was requiring the IP address the client with its request, what rendered this software useless. Modifications were made to grab and send the nameserver, requestor, timestamp and qtype only, instead of the entire DNS answer UDP packet. The raw network traffic was captured with the help of the PCAP libraries. The compiled binary was small and used little CPU time, though running the entire network stream through memory, required a lot more. The average CPU utilization was 65% with a network stream of about 600MBit/s. These figures tended to fluctuate a bit depending on time of day.

The collector software was also build in C and depended on the mysqlclient libraries. The received packets were stripped and put into the database. CPU usage of the collector at peaks was not higher than 1%. Though the machine was able to keep up to the amount of traffic, its performance was appealing when running complex queries.

The advantages of using compiled code above scripts, is that it runs faster and it can be platform optimized for more performance. The use of static field sizes in the UDP stream from the sensor to the collector, made fast memory copy functionality possible.

For future research, it is recommended to modify the collector to record the data to a PostgreSQL database instead of MySQL, because of the query performance issues. Also the use of hardware with PCI-express as sensor is highly recommended.

# B Gathered results from a live network

This chapter contains some examples of results that we gathered of a live network we could use for testing. Gathering DNS queries started on Thursday Januari the 19th and lasted for a period of two weeks. During this period, all data was collected from a live network. Unfortunately, the core switch was not able to filter on certain traffic, so the complete stream of network traffic was copied to the span port. The sensor was at that time equipped with a 100MBit Ethernet card, resulting in a huge loss of traffic. After replacing the card with a 1GBit one, more accurate data was gathered. Almost 6 million of records were recorded during the first run. A second run produced over 20 million records which were more reliable and complete. In total over 3Gb of MySQL database data was collected. The monitoring was ended on Thursday the 2nd February.

## B.1 Matching against known bad domain names

The easiest way to detect clients from the DNS statistics is straightforward by searching for bad domain names in the data. An example is the bot *W32.Linkbot.M* [12]. This bot contacts a controller at the 'home.played.co.uk' domain on port 6667. If we perform a database query on this domain name, we gather useful results, of systems still infected with this particular bot.

```
mysql> select count(id) as cnt,requestor,nameserver,time,qtype,query from dnsquery where query='home.played.co.uk'  group
by requestor order by cnt desc limit 25;
+-----+--------------+-------+------------------+
| cnt | requestor    | qtype | query            |
+-----+--------------+-------+------------------+
| 183 | 10.10.34.59  | A     | home.played.co.uk |
| 178 | 10.10.35.118 | A     | home.played.co.uk |
| 171 | 10.10.35.44  | A     | home.played.co.uk |
| 136 | 10.10.36.53  | A     | home.played.co.uk |
| 123 | 10.10.32.145 | A     | home.played.co.uk |
|  74 | 10.10.33.151 | A     | home.played.co.uk |
|  63 | 10.10.52.144 | A     | home.played.co.uk |
|  55 | 10.10.45.158 | A     | home.played.co.uk |
|  37 | 10.10.40.176 | A     | home.played.co.uk |
|  33 | 10.10.58.159 | A     | home.played.co.uk |
|  32 | 10.10.46.220 | A     | home.played.co.uk |
|  22 | 10.10.35.120 | A     | home.played.co.uk |
|  15 | 10.10.41.86  | A     | home.played.co.uk |
|  14 | 10.10.35.192 | A     | home.played.co.uk |
|  14 | 10.10.54.83  | A     | home.played.co.uk |
|  12 | 10.10.32.37  | A     | home.played.co.uk |
|  12 | 10.10.58.205 | A     | home.played.co.uk |
|  12 | 10.10.44.81  | A     | home.played.co.uk |
|  11 | 10.10.35.121 | A     | home.played.co.uk |
|  11 | 10.10.35.64  | A     | home.played.co.uk |
|  10 | 10.10.40.187 | A     | home.played.co.uk |
|  10 | 10.10.46.4   | A     | home.played.co.uk |
|  10 | 10.10.40.104 | A     | home.played.co.uk |
|   8 | 10.10.59.41  | A     | home.played.co.uk |
|   8 | 10.10.45.250 | A     | home.played.co.uk |
+-----+--------------+-------+------------------+
```

The domain 'home.played.co.uk' does still resolves to an IP address at the moment of writing this report. The list shows IP addresses of systems that have tried to contact the system that used to host the Command & Control IRC channel of the *W32.Linkbot.M*. Fortunately, the controller at this domain has already been dismantled. There is no immediate thread, but the machines remain infected and thus vulnerable.

## B.2 Top 10 Requests

Top 10 ranks can give important information about current network activities. It can show clients that are very active, or domain names that are very popular. Top 10 queries performed per hour can be used for monitoring increased or decreased DNS usage. Abuse can be quickly detected. The examples below show the top ranking queries. It can be noted immediately that the first 3 domain names are at least suspicious. Further investigation can be done with this information. For example a count which systems tried to contact the domain zflh1adf046.kng.mesh.ad.jp' and how many times. An alert could inform an administrator of the behavior of these systems.

```
mysql> select count(id) as cnt,query from dnsquery where time like '2006-01-31 10:%' group by query order by
cnt desc limit 10;
+------+------------------------------+
| cnt  | query                        |
+------+------------------------------+
| 5291 | mail.omgdidyougotpwned.info |
| 3474 | 46.10.239.60.in-addr.arpa    |
| 2763 | flh1adf046.kng.mesh.ad.jp    |
|  900 | mail.onsneteindhoven.nl      |
|  890 | 00.spazbox.net               |
|  794 | rad.msn.com                  |
|  777 | loginnet.passport.com        |
|  716 | dns.auto-startpage.com       |
|  674 | ad.nl.doubleclick.net        |
|  540 | mail.tweakdsl.nl             |
+------+------------------------------+
mysql> select count(id) as cnt,requestor,qtype,query from dnsquery where query like 'flh1adf046.kng.mesh.ad.jp'
and  time like '2006-01-31 10:%' group by requestor order by cnt desc limit 10;
+-----+-------------+-------+--------------------------+
| cnt | requestor    | qtype | query                    |
+-----+-------------+-------+--------------------------+
| 336 | 10.10.55.103 | A     | flh1adf046.kng.mesh.ad.jp |
| 238 | 10.10.35.85  | A     | flh1adf046.kng.mesh.ad.jp |
| 180 | 10.10.41.29  | A     | flh1adf046.kng.mesh.ad.jp |
| 168 | 10.10.48.201 | A     | flh1adf046.kng.mesh.ad.jp |
| 168 | 10.10.36.165 | A     | flh1adf046.kng.mesh.ad.jp |
| 168 | 10.10.37.201 | A     | flh1adf046.kng.mesh.ad.jp |
| 168 | 10.10.56.100 | A     | flh1adf046.kng.mesh.ad.jp |
| 168 | 10.10.47.123 | A     | flh1adf046.kng.mesh.ad.jp |
| 168 | 10.10.54.120 | A     | flh1adf046.kng.mesh.ad.jp |
| 140 | 10.10.33.109 | A     | flh1adf046.kng.mesh.ad.jp |
+-----+-------------+-------+--------------------------+
```

Top 10 requestors per hour can be used for monitor individual systems. The examples below show the top ranking requestors. The number of queries is high. 7600 queries per hour means 2 queries per second. This has to be an automated system. When we investigate further, we see that the system is trying to contact 'mail.omgdidyougotpwned.info' and '00.spazbox.net'.

```
mysql> select count(id) as cnt,requestor from dnsquery where time like '2006-01-31 10:%'
group by requestor order by cnt desc limit 10;
+------+-------------+
| cnt  | requestor    |
+------+-------------+
| 7577 | 10.10.58.13  |
| 7538 | 10.10.38.163 |
| 4872 | 10.10.38.10  |
| 3984 | 10.10.48.70  |
| 2956 | 10.10.42.163 |
| 2228 | 10.10.40.254 |
| 2124 | 10.10.54.120 |
| 2034 | 10.10.54.118 |
| 2005 | 10.10.49.67  |
| 1693 | 10.10.35.213 |
+------+-------------+
```

```
mysql> select count(id) as cnt,requestor,qtype,query from dnsquery where requestor like '10.10.58.13' and  time
like '2006-01-31 10:%' group by query order by cnt desc limit 10;
+------+-------------+-------+-------------------------------------------------------------------------+
| cnt  | requestor   | qtype | query                                                                   |
+------+-------------+-------+-------------------------------------------------------------------------+
| 5291 | 10.10.58.13 | A     | mail.omgdidyougotpwned.info                                             |
|  890 | 10.10.58.13 | A     | 00.spazbox.net                                                          |
|  716 | 10.10.58.13 | A     | dns.auto-startpage.com                                                  |
|  168 | 10.10.58.13 | A     | proxy.corsforcors.com                                                   |
|  132 | 10.10.58.13 | AAAA  | megumi.megumi.simlink.com                                               |
|   24 | 10.10.58.13 | AAAA  | mail.brabant.chello.nl                                                  |
|   13 | 10.10.58.13 | A     | www.google.nl                                                           |
|   10 | 10.10.58.13 | A     | www.burstnet.com                                                        |
|    9 | 10.10.58.13 | A     | www.collegegrad.com                                                     |
|    9 | 10.10.58.13 | A     | anrtx.tacoda.net                                                        |
|    8 | 10.10.58.13 | A     | iprocollect.realmedia.com                                               |
|    8 | 10.10.58.13 | SRV   | _ldap._tcp.dc._msdcs.campus.tue.nl                                      |
+------+-------------+-------+-------------------------------------------------------------------------+
```

## B.3 Infected Systems Monitor and Cross Reference

Matching data from nfdump and the DNS data gives good results. This is an example of how infected systems can be backtracked through DNS information. The flowstart information is raw data extracted with nfdump at a regular interval. One of the IP addresses in the nfdump results gives a direct hit with the DNS data at a specific timestamp.

An alert from a nfdump that raises suspicion.

```
mysql> select flowstart, srcip, packets, bytes from flows where srcip
like "10.10.%" and flowstart like "2006-01-27 %";
+---------------------+-------------+---------+-------+
| flowstart           | srcip       | packets | bytes |
+---------------------+-------------+---------+-------+
| 2006-01-27 19:06:25 | 10.10.44.142 |       1 |   140 |
+---------------------+-------------+---------+-------+
```

Searching the DNS data for a match on the timestamp results in the following information:

```
mysql> select * from dnsquery where requestor = '10.10.44.142' and time = '2006-01-27 19:06:25';
+---------+-------------+-------------+---------------------+-------+----------------------+
| id      | requestor   | nameserver  | time                | qtype | query                |
+---------+-------------+-------------+---------------------+-------+----------------------+
| 5812938 | 10.10.44.142 | 192.87.36.36 | 2006-01-27 19:06:25 | A     | suksa.mujaskax33.com |
+---------+-------------+-------------+---------------------+-------+----------------------+
```

The domain 'suksa.mujaskax33.com' should be investigated further, since it is relatively positive a controller. A search on this domain can be directly dine by searching the DNS data. Many hits can indicate a mass infection. Further information can be gathered on the domain name. When we use the domain name to cross reference it through the Passive DNS Replication Database, it gives us the following domain names that are also pointing at the same host. We can use these newly found addresses to match other systems in the DNS data.

Results from a search in the Passive DNS database.

```
suksa.mujaskax33.com revealed the following information:
suksa.mujaskax33.com  A  212.174.113.131
suksa.mujaskax33.com  A  212.174.113.137
suksa.mujaskax33.com  A  212.174.113.138
suksa.mujaskax33.com  A  212.174.113.240

212.174.113.131 revealed the following information:
suksa.mujaskax33.com  A  212.174.113.131
huat.njs8alla.com     A  212.174.113.131
qprw.dscipseib.com    A  212.174.113.131
ncusa.gaxxe45cf.com   A  212.174.113.131
lossa.lertgo23q.com   A  212.174.113.131
```

When we run these domain names through the DNS data, we find more hosts contacting the specified domains.

A search in the DNS data for the newly found domain names produced the following matches.

```
mysql> select count(id) as cnt,requestor,nameserver,time,qtype,query from dnsquery where query = 'suksa.mujaskax33.com'
group by requestor order by cnt;
+-----+--------------+--------------+---------------------+-------+----------------------+
| cnt | requestor    | nameserver   | time                | qtype | query                |
+-----+--------------+--------------+---------------------+-------+----------------------+
| 111 | 10.10.58.159 | 192.87.36.36 | 2006-01-27 19:14:46 | A     | suksa.mujaskax33.com |
| 176 | 10.10.44.142 | 192.87.36.36 | 2006-01-26 20:00:15 | A     | suksa.mujaskax33.com |
+-----+--------------+--------------+---------------------+-------+----------------------+
2 rows in set (1 min 56.50 sec)

mysql> select count(id) as cnt,requestor,nameserver,time,qtype,query from dnsquery where query = 'huat.njs8alla.com'
group by requestor order by cnt;
Empty set (1 min 56.39 sec)

mysql> select count(id) as cnt,requestor,nameserver,time,qtype,query from dnsquery where query = 'qprw.dscipseib.com'
group by requestor order by cnt;
+-----+--------------+--------------+---------------------+-------+--------------------+
| cnt | requestor    | nameserver   | time                | qtype | query              |
+-----+--------------+--------------+---------------------+-------+--------------------+
|   2 | 10.10.52.147 | 212.57.1.18  | 2006-01-26 18:01:44 | A     | qprw.dscipseib.com |
|  55 | 10.10.42.228 | 192.87.36.36 | 2006-01-27 11:09:50 | A     | qprw.dscipseib.com |
| 205 | 10.10.38.206 | 212.57.1.18  | 2006-01-28 23:59:10 | A     | qprw.dscipseib.com |
+-----+--------------+--------------+---------------------+-------+--------------------+
3 rows in set (1 min 58.56 sec)

mysql> select count(id) as cnt,requestor,nameserver,time,qtype,query from dnsquery where query = 'ncusa.gaxxe45cf.com'
group by requestor order by cnt;
Empty set (1 min 59.26 sec)

mysql> select count(id) as cnt,requestor,nameserver,time,qtype,query from dnsquery where query = 'lossa.lertgo23q.com'
group by requestor order by cnt;
Empty set (1 min 43.32 sec)
```

This does show that by crossreferencing the DNS queries systems perform, infected systems can be identified. A search in the nfdump confirms that the found IP addresses are in fact infected. The gathered IP's match back in a nfdump result

| flowstart | srcip | packets | bytes |
|---|---|---|---|
| 2006-01-25 10:47:22 | 10.10.44.142 | 1 | 55 |
| 2006-01-26 18:03:33 | 10.10.52.147 | 1 | 40 |
| 2006-01-27 11:09:41 | 10.10.42.228 | 1 | 157 |
| 2006-01-27 21:47:16 | 10.10.58.159 | 1 | 55 |
| 2006-01-29 16:30:00 | 10.10.38.206 | 1 | 55 |

We can safely assume that systems that make contact on any of these domain names are under the influence of an external system. We have also found three domain names that don't match yet. When we keep monitoring these domain names in the logs, and configure a trigger when one of these domains is resolved, we can monitor new infections. Further investigation can also be stated.

## B.4 Resolver Deviation

Machines that normally use the preferred nameservers, but in some cases don't can point to an infection. The displayed table shows machines that are querying a known bad domain with help from other than the preferred nameservers.

```
mysql> select d.requestor,d.nameserver,d.query from dnsquery d, dnsquery q where d.query='80gw6ry3i3x3qbrkwhxhw.032439.com' and
d.nameserver!='192.87.36.36' and d.nameserver!='192.87.106.106' and (q.nameserver='192.87.36.36' or q.nameserver='192.87.106.106') and
q.requestor=d.requestor group by nameserver limit 30;
+-------------+---------------+---------------------------------+
| requestor   | nameserver    | query                           |
+-------------+---------------+---------------------------------+
| 10.10.38.206 | 193.140.83.251 | 80gw6ry3i3x3qbrkwhxhw.032439.com |
| 10.10.38.206 | 205.171.2.65   | 80gw6ry3i3x3qbrkwhxhw.032439.com |
| 10.10.52.147 | 212.175.13.114 | 80gw6ry3i3x3qbrkwhxhw.032439.com |
| 10.10.38.206 | 212.57.1.17    | 80gw6ry3i3x3qbrkwhxhw.032439.com |
| 10.10.38.206 | 212.57.1.18    | 80gw6ry3i3x3qbrkwhxhw.032439.com |
| 10.10.55.120 | 85.255.112.15  | 80gw6ry3i3x3qbrkwhxhw.032439.com |
| 10.10.41.36  | 85.255.112.185 | 80gw6ry3i3x3qbrkwhxhw.032439.com |
| 10.10.44.58  | 85.255.113.114 | 80gw6ry3i3x3qbrkwhxhw.032439.com |
| 10.10.33.55  | 85.255.115.52  | 80gw6ry3i3x3qbrkwhxhw.032439.com |
| 10.10.55.120 | 85.255.115.58  | 80gw6ry3i3x3qbrkwhxhw.032439.com |
| 10.10.41.36  | 85.255.116.141 | 80gw6ry3i3x3qbrkwhxhw.032439.com |
+-------------+---------------+---------------------------------+
11 rows in set (21 min 38.86 sec)
```

Clearly vissible is that 10.10.38.206 is probably infected with badware, but DHCP is making statistics more complicated, leading to false conclusions.

The following query looks at the first request done per hour. By this, one can see a machine being powered on. Also is visible that the DHCP server assigns this IP to another machine on 2006-01-28 23:51:52. Also visible is that since that particulair time, the nameservers 192.87.36.36 and 192.87.106.106 aren't queried anymore.

```
mysql> select time,requestor,nameserver,query from dnsquery where requestor='10.10.38.206' group by date_format(time,'%Y-%m-%d %H');
+---------------------+-------------+---------------+---------------------------------+
| time                | requestor   | nameserver    | query                           |
+---------------------+-------------+---------------+---------------------------------+
| 2006-01-26 11:30:18 | 10.10.38.206 | 192.87.36.36  | zonelabs.com                    |
| 2006-01-26 12:00:12 | 10.10.38.206 | 192.87.36.36  | zonelabs.com                    |
| 2006-01-26 13:00:09 | 10.10.38.206 | 192.87.36.36  | 47.32.35.84.in-addrarpa         |
| 2006-01-26 14:00:10 | 10.10.38.206 | 192.87.36.36  | toolbarqueries.google.nl        |
| 2006-01-26 15:00:07 | 10.10.38.206 | 192.87.36.36  | zonelabs.com                    |
| 2006-01-27 14:13:58 | 10.10.38.206 | 192.87.36.36  | messenger.hotmail.com           |
| 2006-01-27 15:00:21 | 10.10.38.206 | 192.87.36.36  | 160.36.35.84.in-addr.arpa       |
|*2006-01-28 23:51:52*| 10.10.38.206 | 205.171.2.65  | 80gw6ry3i3x3qbrkwhxhw.032439.com |
| 2006-01-29 00:00:00 | 10.10.38.206 | 212.57.1.18   | nunah.info                      |
| 2006-01-29 01:03:14 | 10.10.38.206 | 205.171.2.65  | www.ad-w-a-r-e.com              |
| 2006-01-29 10:55:36 | 10.10.38.206 | 205.171.2.65  | www.ad-w-a-r-e.com              |
| 2006-01-29 11:00:05 | 10.10.38.206 | 212.57.1.18   | wicked.unstable-insecure.info   |
| 2006-01-29 12:00:05 | 10.10.38.206 | 205.171.2.65  | blast.pp.ru                     |
| 2006-01-29 13:00:01 | 10.10.38.206 | 205.171.2.65  | blast.pp.ru                     |
| 2006-01-29 14:00:01 | 10.10.38.206 | 212.175.13.114 | blast.pp.ru                    |
| 2006-01-29 15:00:00 | 10.10.38.206 | 205.171.2.65  | vother.info                     |
| 2006-01-29 16:00:01 | 10.10.38.206 | 205.171.2.65  | blast.pp.ru                     |
| 2006-01-29 17:00:00 | 10.10.38.206 | 212.175.13.114 | blast.pp.ru                    |
| 2006-01-29 18:00:00 | 10.10.38.206 | 205.171.2.65  | lud.pass.as                     |
| 2006-01-29 19:00:01 | 10.10.38.206 | 205.171.2.65  | blast.pp.ru                     |
| 2006-01-29 20:00:02 | 10.10.38.206 | 212.175.13.114 | cgi.ebay.nl                    |
| ...                 |             |               |                                 |
+---------------------+-------------+---------------+---------------------------------+
```

This machine came to the eye because its use of also other than the preferred nameservers. Although no real nameserver deviation was showed, other weird things were noticable. Powerup behaviour of this machine is not normal. The first request performed is to resolve 80gw6ry3i3x3qbrkwhxhw.032439.com, that possibly hosts a bot-controller.

When looking for machines that are using only other nameserver than the preferred ones, one can notice that the list shows machines responsible for undesired activities.

```
mysql> select count(id) as cnt,requestor,nameserver,query from dnsquery where nameserver!='192.87.36.36' and nameserver!='192.87.106.106'
group by requestor,nameserver order by cnt desc limit 30;
+-------+-------------+---------------+-----------------------------------------------------------------+
| cnt   | requestor   | nameserver    | query                                                           |
+-------+-------------+---------------+-----------------------------------------------------------------+
|  60762 | 10.10.49.67  | 212.203.12.53  | surveyworld.net.sc.surbl.org                                   |
|  32244 | 10.10.38.206 | 205.171.2.65   | 80gw6ry3i3x3qbrkwhxhw.032439.com                               |
|  17095 | 10.10.38.206 | 212.57.1.18    | 00.devoid.us                                                   |
|  16341 | 10.10.38.10  | 128.63.2.53    | NS2.DNS.BR                                                     |
|  16252 | 10.10.38.10  | 202.12.27.33   | n.ns.spamhaus.org                                              |
|   9237 | 10.10.38.118 | 85.255.113.109 | tracker.prq.to                                                 |
|   2080 | 10.10.38.118 | 85.255.112.92  | davide.atspace.com                                             |
+-------+-------------+---------------+-----------------------------------------------------------------+
```

## B.5 Uncommon AXFR/IXFR requests

We see that just a couple of hosts make lots of uncommon queries. This can indicate hacking or a bot infection. Further investigation can be stated.

```
mysql> select count(id) as cnt,qtype,requestor,nameserver,time,query from dnsquery where qtype = 'AXFR' group
by query order by cnt desc limit 250;
+------+-------+-------------+----------------+---------------------+----------------------------------+
| cnt  | qtype | requestor   | nameserver     | time                | query                            |
+------+-------+-------------+----------------+---------------------+----------------------------------+
...
|  395 | AXFR  | 10.10.38.10 | 192.5.5.241    | 2006-01-27 03:47:39 | netscape.net                     |
...
|  353 | AXFR  | 10.10.38.10 | 192.203.230.10 | 2006-01-27 03:39:14 | hotmail.com                      |
...
|  296 | AXFR  | 10.10.38.10 | 128.9.0.107    | 2006-01-27 04:10:02 | cisco.com                        |
...
|  260 | AXFR  | 10.10.38.10 | 198.41.0.4     | 2006-01-27 03:43:10 | gmail.com                        |
...
|  158 | AXFR  | 10.10.38.10 | 128.8.10.90    | 2006-01-27 05:17:17 | msn.com                          |
...
|  124 | AXFR  | 10.10.38.10 | 198.41.0.4     | 2006-01-27 04:20:41 | nl.abnamro.com                   |
+------+-------+-------------+----------------+---------------------+----------------------------------+

mysql> select count(id) as cnt,qtype,requestor,nameserver,time,query from dnsquery where qtype = 'IXFR' group
by query order by cnt desc limit 250;
+-----+-------+-------------+----------------+---------------------+----------------------------------+
| cnt | qtype | requestor   | nameserver     | time                | query                            |
+-----+-------+-------------+----------------+---------------------+----------------------------------+
| 451 | IXFR  | 10.10.37.151 | 213.132.174.72 | 2006-01-27 03:05:19 | geldtmeijer.nl                   |
| 447 | IXFR  | 10.10.37.151 | 213.132.174.72 | 2006-01-27 03:06:27 | mkd-lelystad.nl                  |
| 447 | IXFR  | 10.10.37.151 | 213.132.174.72 | 2006-01-27 03:06:42 | lachambrebrocante.nl             |
| 445 | IXFR  | 10.10.37.151 | 213.132.174.72 | 2006-01-27 03:05:55 | ledervaren.nl                    |
| 445 | IXFR  | 10.10.37.151 | 213.132.174.72 | 2006-01-27 03:06:56 | williamvanantwerpen.nl           |
| 444 | IXFR  | 10.10.37.151 | 213.132.174.72 | 2006-01-27 03:04:59 | mantra.nl                        |
| 442 | IXFR  | 10.10.37.151 | 213.132.174.72 | 2006-01-27 03:06:12 | cel-online.nl                    |
| 441 | IXFR  | 10.10.37.151 | 213.132.174.72 | 2006-01-27 03:05:34 | evolvere.nl                      |
| 441 | IXFR  | 10.10.37.151 | 213.132.174.72 | 2006-01-27 03:06:26 | kattenvel.nl                     |
| 441 | IXFR  | 10.10.37.151 | 213.132.174.72 | 2006-01-27 03:06:34 | rolsma.nl                        |
+-----+-------+-------------+----------------+---------------------+----------------------------------+
```

Further investigation can be stated.

# References


[1] **University of Amsterdam, System- and Network Engineering**
http://www.os3.nl

[2] **SURFnet**
http://www.surfnet.nl

[3] **Network Emergency Responder and Detector**
http://www.nerdd.org

[4] **Peter Haag's nfdump**
http://nfdump.sourceforge.net/

[5] **SURFnet IDS Project**
http://ids.surfnet.nl

[6] **David Dagon - Botnet Detection and Response**
http://www.caida.org/projects/oarc/200507/slides/oarc0507-Dagon.pdf

[7] **John Kristoff - Botnet Detection and Mitigation**
http://www.it.northwestern.edu/bin/docs/bots_kristoff_jul05.ppt

[8] **Passive DNS Replication**
http://www.enyo.de/fw/software/dnslogger/first2005-paper.pdf

[9] **John Kristoff - DNSWatch**
http://aharp.ittns.northwestern.edu/software

[10] **'A Discussion of Bot Networks'**
http://www.educause.edu/ir/library/pdf/SPC0568.pdf

[11] **Sensor and Collector source code available**
http://www.os3.nl/~talitwan/RP1/

[12] **W32.Linkbot.M information**
http://www.symantec.com/avcenter/venc/data/w32.linkbot.m.html

[13] **Fraud Watch International**
http://www.fraudwatchinternational.com/main.shtml

[14] **SecuritTeam Blog of Gadi Evron**
http://blogs.securiteam.com/index.php/archives/author/gadi/

[15] **nsp-security closed mailing list**
https://puck.nether.net/mailman/listinfo/nsp-security