# Global Information Assurance Certification Paper

# Protecting Home Devices from Malicious or Blacklisted Websites

*GIAC (GCIA) Gold Certification*

Author: Sumesh Shivdas, msumesh@hotmail.com
Advisor: Dr. Toby Gouker
Accepted: June 25th 2015

Abstract

The majority of the devices on a home network have unrestricted outbound connectivity to the Internet. (Barcena & Wueest, 2015) Other than the use of "opendns", which only provides some protection against phishing, fraud and limited blacklisting, a homeowner's options are limited. To provide protection from known malicious sites and produce DNS query logs for further detailed analysis, a simple virtual machine set up with DNS is proposed. When coupled with "opendns", unlimited blacklisting capability and automatic updates to block malicious sites from all devices is provided. The solution also provides the capability to analyze all the DNS logs using a log based Intrusion Detection System like OSSEC.

# 1. Introduction

The number of devices connecting to a typical home network has been steadily increasing over time. (Barcena & Wueest, 2015) Consumer networking devices, like routers and switches, make it very convenient for these multiple devices, wired or wireless, to connect to the internet. Home networks, unlike business networks, do not use firewalls to control egress traffic. Home routers act as a simple ingress firewall as they perform NAT (Network Address Translation), functionality shielding the home network from the internet. Home routers do not block any egress traffic by default.

The popularity of internet and the reliance on it for almost anything has given rise to increased internet related crimes (Symantec Internet Security Web Threat Report, 2015). One of the areas of concern is the threat from malware delivered from visiting malicious or compromised web sites. Malware can steal personal information from systems, or it can make our system part of a bigger botnet used for evil purposes. Malware are becoming complex and intelligent. They deploy techniques to delay identification of the command and control center.

The cleanest way of recovery from a malware infected computer is to format and re-install the OS; then the data should be restored. While re-installing the OS is the cleanest way to recover from a malware event, it is also time consuming and tedious. It would be far more efficient to prevent malware from being successful than to spend time cleaning up after detected intrusions. OpenDNS provides a good, no-cost option to block some of the sites and prevent phishing, but there is no easy way to identify which individual system from the home network is trying to connect to a malware site. It has information only at the level of the Internet facing IP address assigned by the ISP to the home network. Additionally, there are several other drawbacks in OpenDNS: it limits the number of sites one can whitelist or blacklist. It does not provide DNS queries for further analysis, and the retention period of the queries are limited to a week. In this paper an alternative solution is proposed, which can be easily implemented using open source and freely available software. The solution can protect home users from malware sites and also gives them an ability to blacklist or whitelist unlimited number of sites. It also

produces all the DNS queries performed by individual systems in the home network for further analysis.

## 2. Background -Typical Home Internet Setup

The figure below depicts a typical internet setup for home users. A modem connects the home network to the internet. A router allows multiple home devices to connect to the internet via the cable modem. The router does Network Address Translation (NAT) and acts as a hardware firewall hiding the home network from the internet. The router is configured to use the DNS servers, provided by ISP, for name resolution.

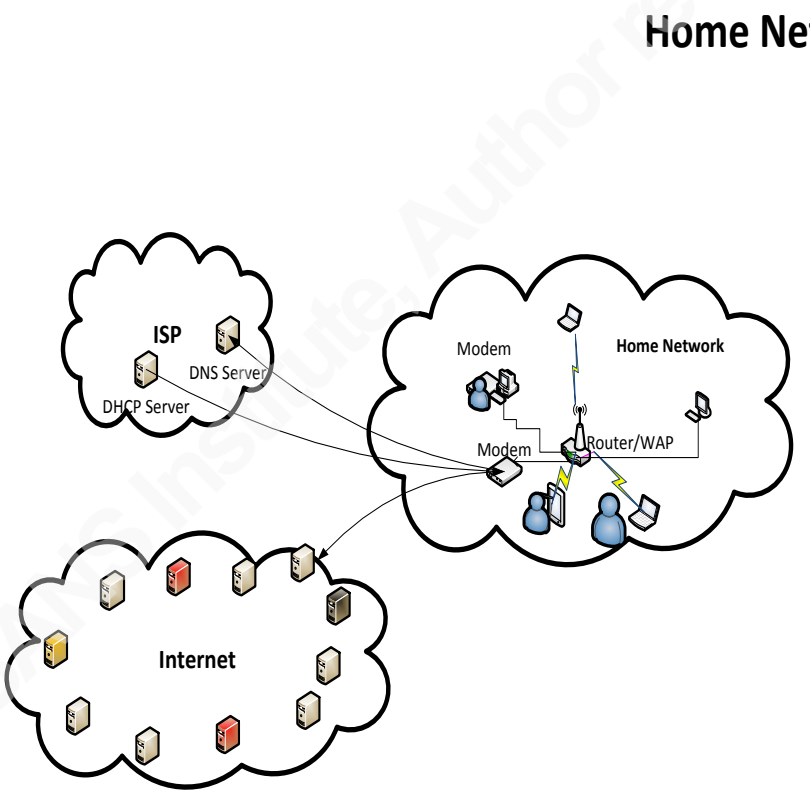



Figure 1: Typical Home Network

In this home network setup there is no capability to restrict the host(s) from accessing a blacklisted server delivering malicious content. In addition, there is no ability to know which host(s) from the host network is accessing the malicious site.

The DNS query flow in this setup is shown in the figure below. All the resolutions are performed by a default DNS server provided by the ISP.

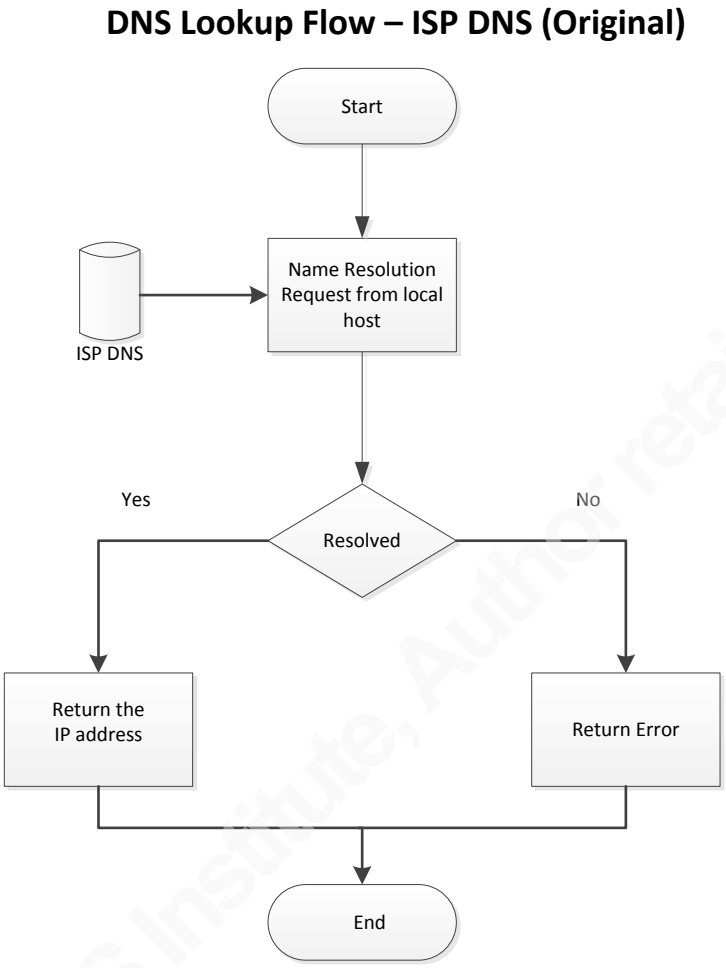



Figure 2: DNS Lookup with ISP DNS

## 3. Proposed Solution

### 3.1 New Home Internet Setup

The figure below shows the proposed setup for a home network. It includes a DNS server running on a virtual machine. The virtual machine is OSSEC Virtual Appliance 2.8.1 Oracle Virtual box 4.3.28 running under Windows 7 OS. Bind version 9 was installed on the Virtual Machine to run as the Internal DNS server.
This setup requires the home network devices to do the name resolution using the virtual DNS server setup in the Home Network. This DNS server is setup as a

forwarder. It will forward the DNS queries, which it cannot resolve to OpenDNS DNS servers. The ISP provided DNS servers are bypassed in this setup. The figure below depicts the DNS query flow in this setup.

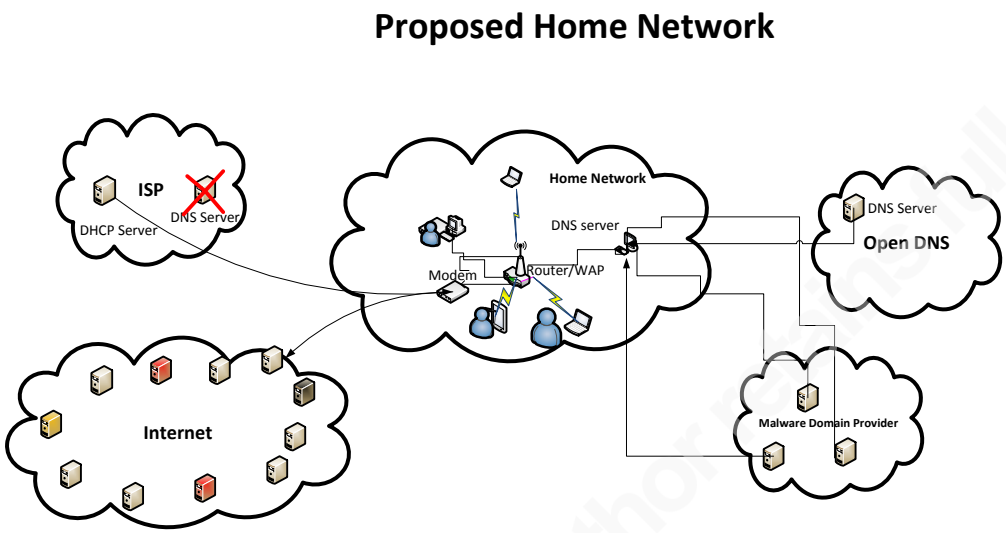
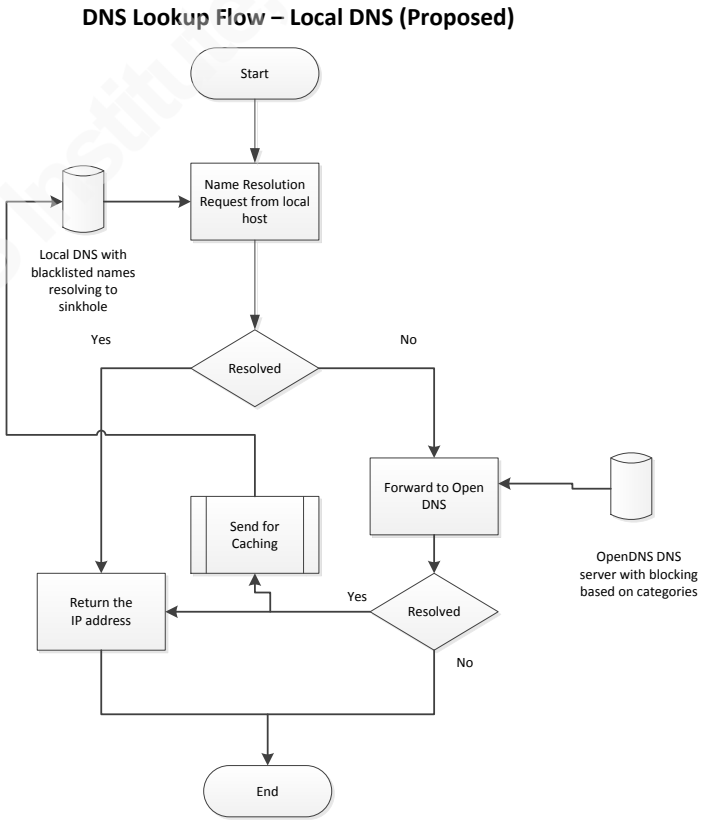



Figure 3: Proposed Home Network




Sumesh Shivdas, msumesh@hotmail.com

Figure 4: DNS flow with proposed setup

### 3.2 Open DNS Setup

A free "opendns" home account was setup to act as an external DNS. Open DNS provides web content based filtering (categories listed below), basic malware protection, two weeks of basic reporting and phishing protection. It also allows users to apply custom white listing and blacklisting limited to 25 entries each.

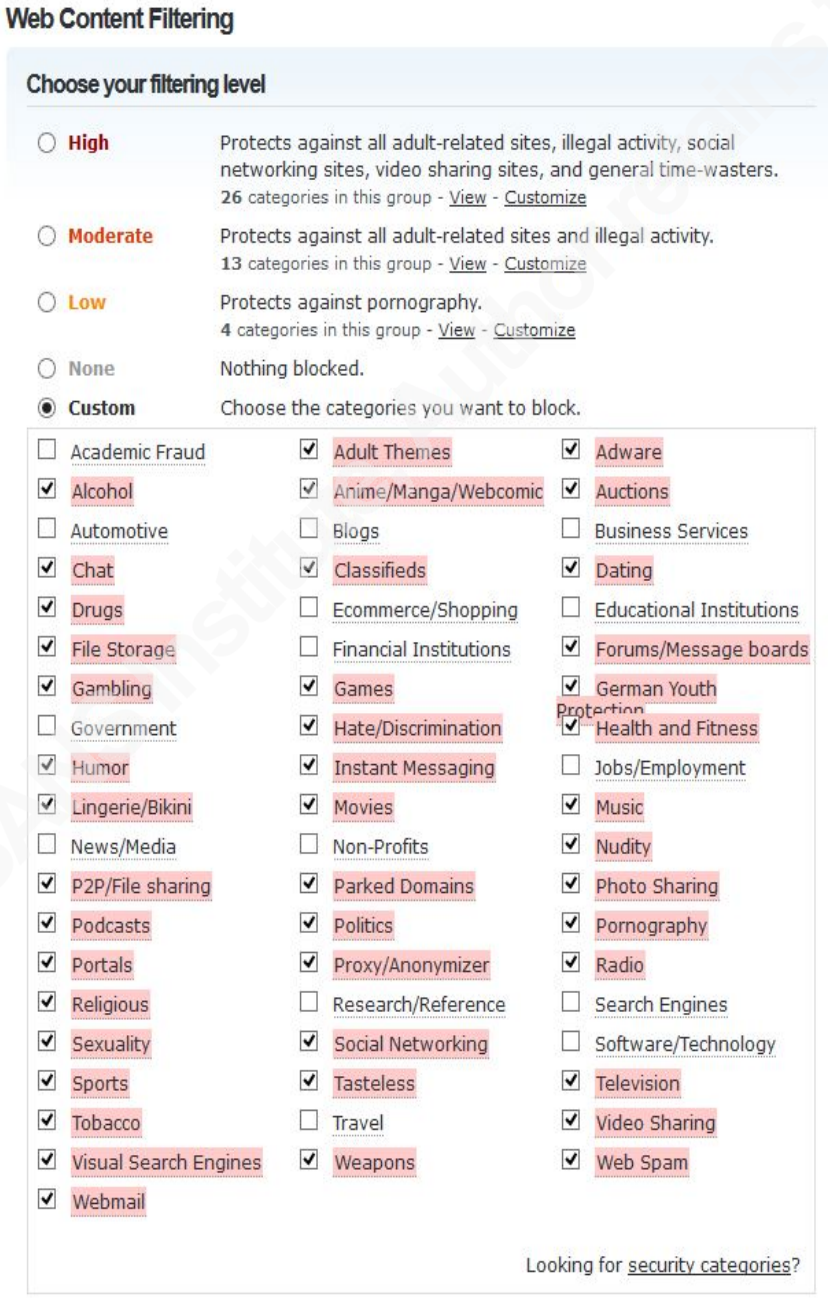


Figure 5: Filter categories available in OpenDNS

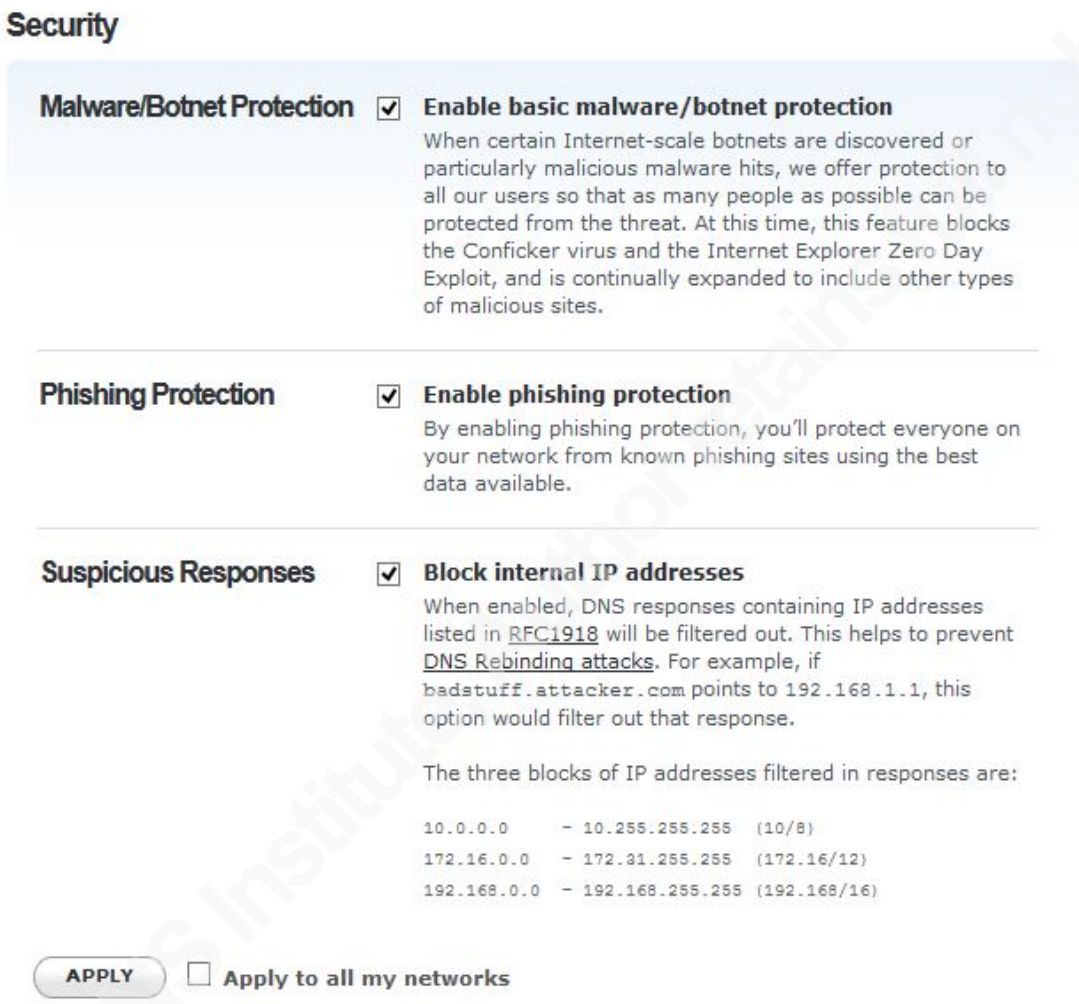



Figure 6: Basic Malware protection in OpenDNS

### 3.3 Home DNS sever Setup

Bind package was installed on the OSSEC Virtual Server to setup a local DNS server. The snippet below shows the installation command and part of the logs.

```
[user@ossec dns]$ more install.log
user@ossec ~]$ sudo yum install bind
Loaded plugins: fastestmirror, refresh-packagekit, security
Setting up Install Process
Determining fastest mirrors
 * base: mirror.symnds.com
 * extras: centos.den.host-engine.com
 * updates: mirrors.kernel.org
base                                                                      | 3.7 kB     00:00
extras                                                                    | 3.4 kB     00:00
extras/primary_db                                                         |  31 kB     00:00
updates                                                                   | 3.4 kB     00:00
updates/primary_db                                                        | 3.3 MB     00:01
Resolving Dependencies
--> Running transaction check
---> Package bind.x86_64 32:9.8.2-0.30.rc1.el6_6.2 will be updated
---> Package bind.x86_64 32:9.8.2-0.30.rc1.el6_6.3 will be an update
--> Processing Dependency: bind-libs = 32:9.8.2-0.30.rc1.el6_6.3 for package: 32:bind-9.8.2-0.30.rc1.el6_6.3.x86_64
--> Running transaction check
---> Package bind-libs.x86_64 32:9.8.2-0.30.rc1.el6_6.2 will be updated
--> Processing Dependency: bind-libs = 32:9.8.2-0.30.rc1.el6_6.2 for package: 32:bind-utils-9.8.2-0.30.rc1.el6_6.2.x86_64
---> Package bind-libs.x86_64 32:9.8.2-0.30.rc1.el6_6.3 will be an update
```

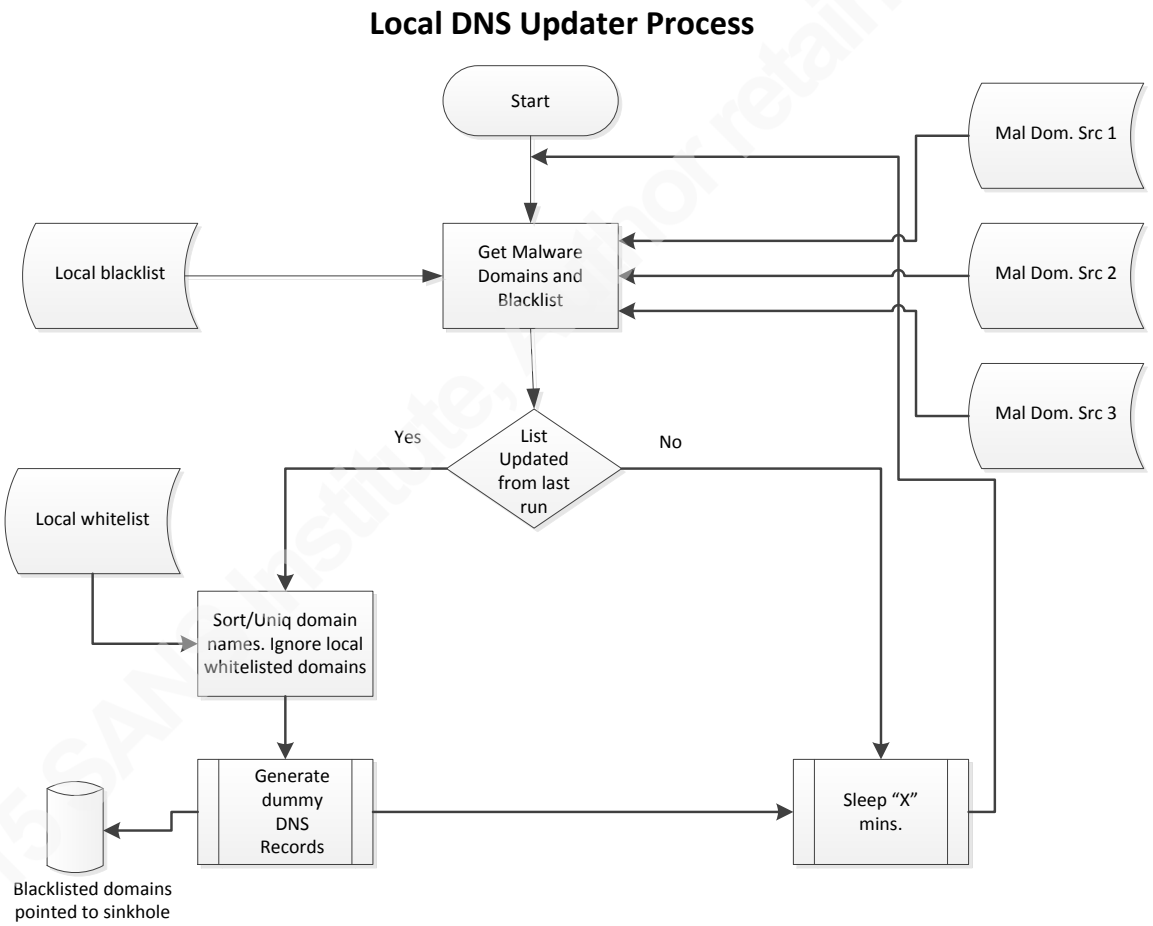



Figure 7: Refreshing malware domain list

The figure above shows how the local DNS server gets updated. A process constantly runs on the DNS server. This server polls the configured servers, providing a list of

malware domains and looking at the optional local user configured blacklist file. The process checks if the files are updated based on the last downloaded file checksums. If the files are updated, it creates a unique list of malware domains from the names, compares it with an options whitelist file provided by the user (never block this site), and ignores any domain in the whitelist. It then creates DNS entries pointing the blacklisted domain names to a non-existing local IP address (10.0.0.1 in this case). Any device in the home network attempting to resolve the names in this DNS database will be pointed to the non-existing IP address and will not be able to connect to the actual server. Multiple sources of known malicious servers can help keep a more complete and up-to-date list.

The Local DNS server is configured as a forwarder, as shown in the figure 4:DNS Lookup Flow – Proposed.

The following updates are done to the named configuration file /etc/named.conf. Access Control List "goodclients" is defined as the private Home IP address range. This allows limiting the clients that can query the DNS server. The DNS server has been assigned a fixed IP address (reserved in the DHCP server) of 192.168.1.21. The recursion has to be set to "yes" to allow the DNS server to forward the request to the OpenDNS DNS servers. The client queries are configured to be logged under /var/named/log/client_queries.log with 3 versions and 5Mb in size. Finally, a zone file is included, which contains sinkhole domains.

```
acl goodclients {
    192.168.1.0/24;
    127.0.0.1;
};


options {
        listen-on port 53 { 127.0.0.1; 192.168.1.21; };
        allow-query     { goodclients; };
        recursion yes;
        forwarders {
          208.67.222.123;
          208.67.220.123;
        };
        forward only;

        dnssec-enable no;
        dnssec-validation no;
        dnssec-lookaside auto;

};


logging {
    channel queries_file {
        file "/var/log/named/client_queries.log" versions 3 size 5m;
        severity dynamic;
        print-time yes;
    }
    category queries { queries_file; };
};

include "/etc/named.blacklist.zones";
```

An example of what the /etc/named.blacklist.zones file looks like is given below. It contains the name of the domain and the name of the file containing the resolution information. All the zones are pointing to the same file /var/named/data/blacklist.host.

```
zone "support-mailweb.info"  {type master; file "/var/named/data/blacklist.hosts";};
zone "tkss.be"  {type master; file "/var/named/data/blacklist.hosts";};
zone "toolsinc.info"  {type master; file "/var/named/data/blacklist.hosts";};
zone "updat101up.ddns.net"  {type master; file "/var/named/data/blacklist.hosts";};
zone "wahacapitalua.com"  {type master; file "/var/named/data/blacklist.hosts";};
zone "xenonlab.ws"  {type master; file "/var/named/data/blacklist.hosts";};
// Added from local Blacklist
zone "espn.go.com"  {type master; file "/var/named/data/blacklist.hosts";};
zone "espn.com"  {type master; file "/var/named/data/blacklist.hosts";};
zone "facebook.com"  {type master; file "/var/named/data/blacklist.hosts";};
```

Sumesh Shivdas, msumesh@hotmail.com

The blacklist.host file contains the following information. It is resolving all the names (*) to the IP address 10.0.0.1 which is non-existent in the home network hence non-routable. Any attempt to access it will fail.

```
; This zone will Redirect all requests back to the non existing IP address (Sinkhole)

; Time to Live set to 86400 seconds (1 day)
$TTL    86400

@       IN      SOA     mydomain.com. mydomain.com. (
                        20150611103033 ; Serial Number
                        21600   ; Refresh  8 hours
                        7200    ; Update Retry    2 hours
                        864000  ; expire  10 days
                        86400 ) ; min ttl  1 day
                  NS      ossec.mydomain.com.

                  A       10.0.0.1

*               IN      A       10.0.0.1
```

### 3.4 Home Router Setup

The home router has to be configured to point to the new DNS server(s). In the test case, one VM has been set up. One would need to set another for fallback or configure the OpenDNS dns server as secondary DNS server.

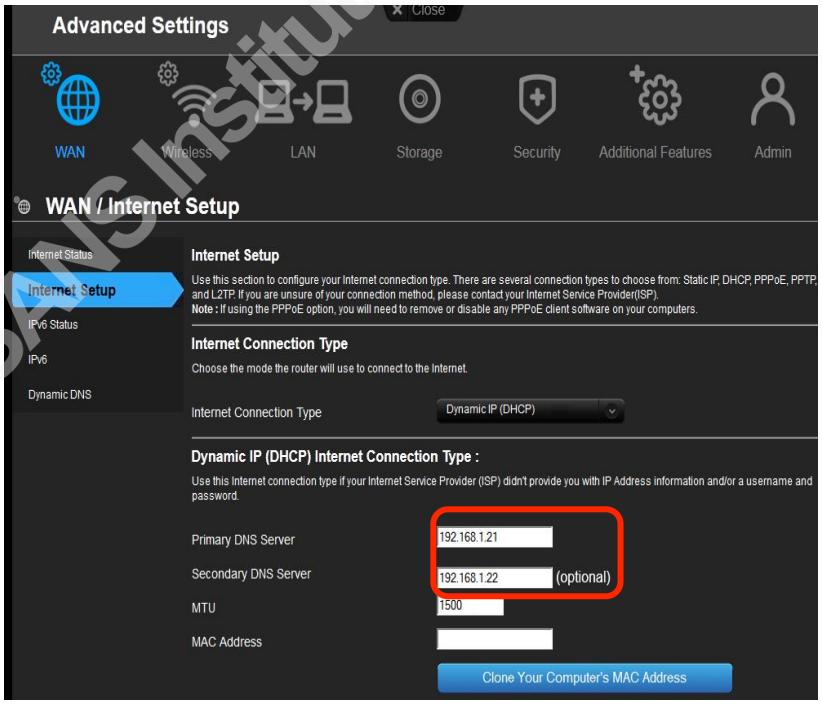



Figure 8: DNS setup at the router

The client gets the DNS IP address from the router. Above is a windows client showing the IP address of the active router.

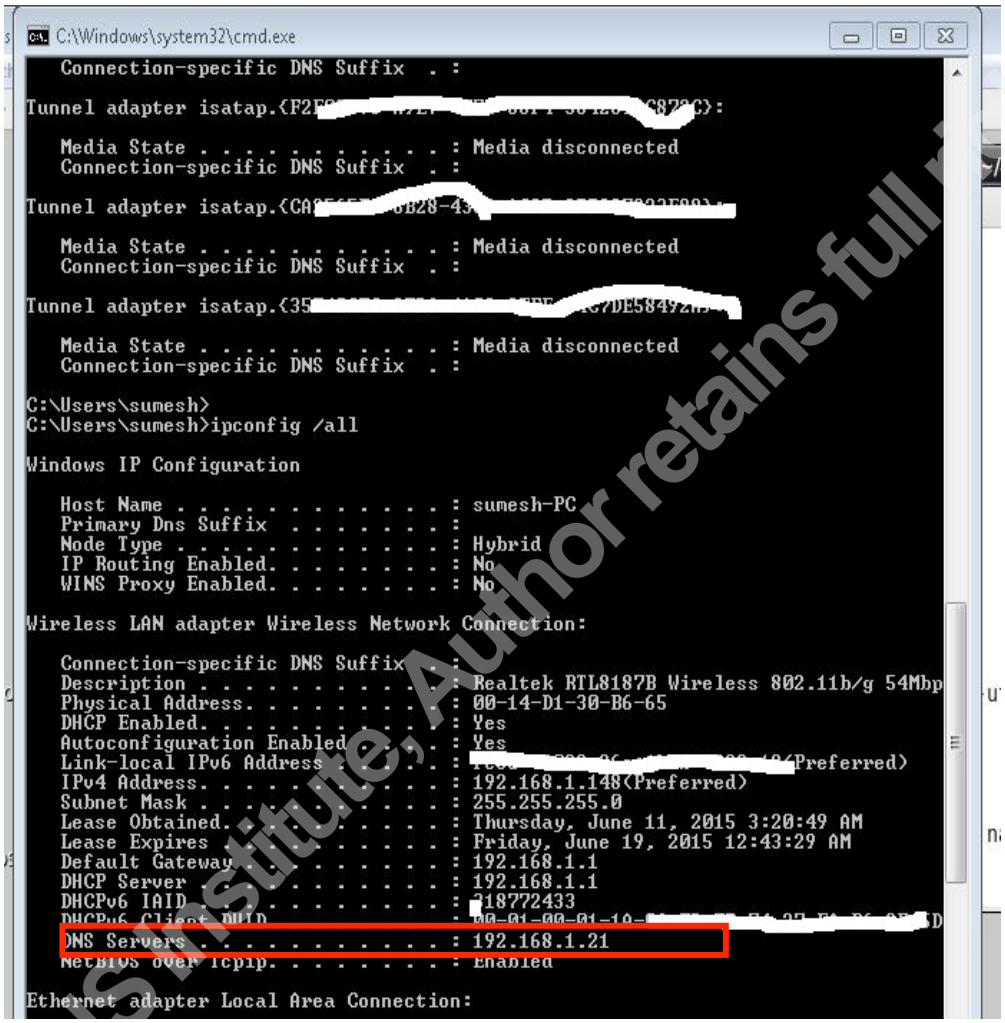
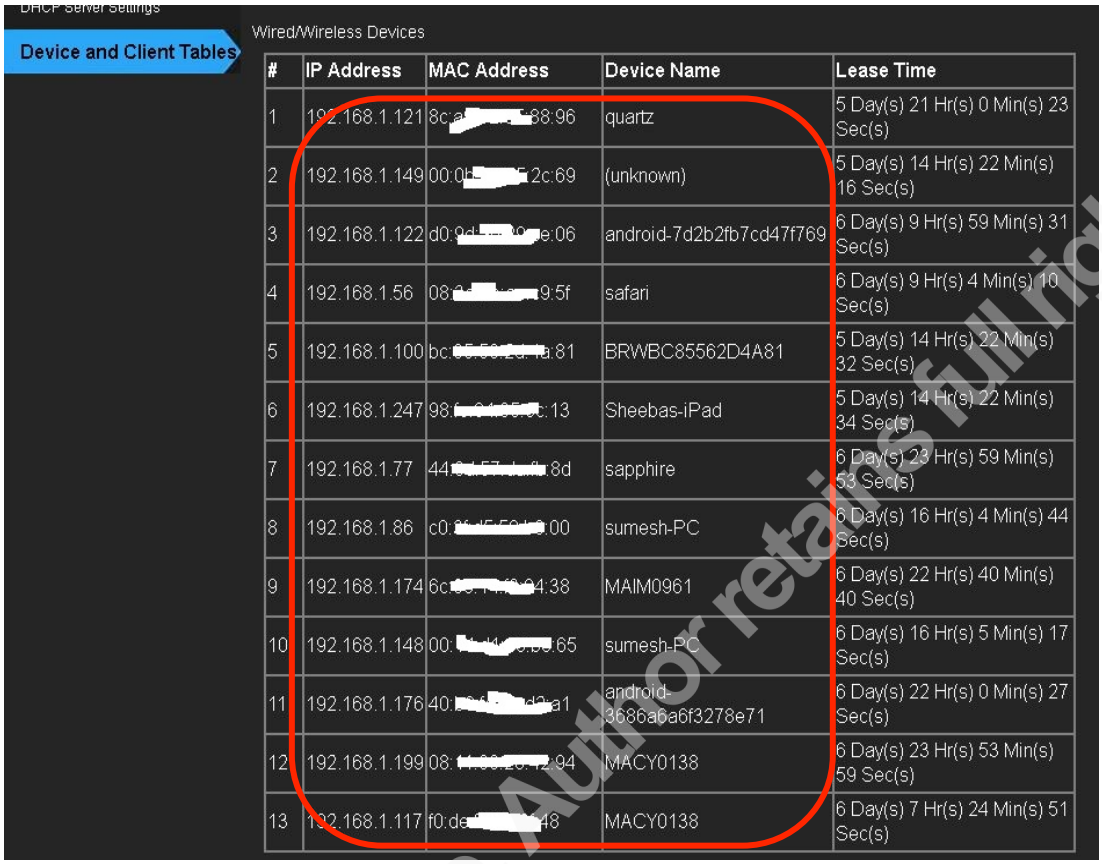



Figure 9: DNS servers at the client

The figure below shows a mapping of the host IP address and associated MAC addresses and host names. This can be used to get the name of the host from the DNS query logs.




| # | IP Address | MAC Address | Device Name | Lease Time |
|---|---|---|---|---|
| 1 | 192.168.1.121 | 8c:a[redacted]88:96 | quartz | 5 Day(s) 21 Hr(s) 0 Min(s) 23 Sec(s) |
| 2 | 192.168.1.149 | 00:0[redacted]2c:69 | (unknown) | 5 Day(s) 14 Hr(s) 22 Min(s) 16 Sec(s) |
| 3 | 192.168.1.122 | d0:[redacted]e:06 | android-7d2b2fb7cd47f769 | 6 Day(s) 9 Hr(s) 59 Min(s) 31 Sec(s) |
| 4 | 192.168.1.56 | 08:[redacted]9:5f | safari | 6 Day(s) 9 Hr(s) 4 Min(s) 10 Sec(s) |
| 5 | 192.168.1.100 | bc:[redacted]a:81 | BRWBC85562D4A81 | 5 Day(s) 14 Hr(s) 22 Min(s) 32 Sec(s) |
| 6 | 192.168.1.247 | 98:[redacted]:13 | Sheebas-iPad | 5 Day(s) 14 Hr(s) 22 Min(s) 34 Sec(s) |
| 7 | 192.168.1.77 | 44:[redacted]8d | sapphire | 6 Day(s) 23 Hr(s) 59 Min(s) 53 Sec(s) |
| 8 | 192.168.1.86 | c0:[redacted]:00 | sumesh-PC | 6 Day(s) 16 Hr(s) 4 Min(s) 44 Sec(s) |
| 9 | 192.168.1.174 | 6c:[redacted]4:38 | MAIM0961 | 6 Day(s) 22 Hr(s) 40 Min(s) 40 Sec(s) |
| 10 | 192.168.1.148 | 00:[redacted]:65 | sumesh-PC | 6 Day(s) 16 Hr(s) 5 Min(s) 17 Sec(s) |
| 11 | 192.168.1.176 | 40:[redacted]a1 | android-3686a6a6f3278e71 | 6 Day(s) 22 Hr(s) 0 Min(s) 27 Sec(s) |
| 12 | 192.168.1.199 | 08:[redacted]:94 | MACY0138 | 6 Day(s) 23 Hr(s) 53 Min(s) 59 Sec(s) |
| 13 | 192.168.1.117 | f0:de[redacted]48 | MACY0138 | 6 Day(s) 7 Hr(s) 24 Min(s) 51 Sec(s) |

Figure 10: Client list at the router

### 3.5 Advanced setup

The setup in this section is optional. This setup is only required if one needs to prevent users in the home network from intentionally bypassing the local DNS setup thus bypassing the controls.

The figure below shows firewall rules to control the DNS queries. DNS queries to external DNS servers will only be allowed from the local DNS server. DNS queries from any other host in the home network will be denied. The outbound firewall rule will block any user or system attempting to connect to internet bypassing the DNS server assigned by the router (e.g. attempting to use google or ISP provided DNS servers by modifying the DNS servers in the TCP/IP advanced properties). Firewall settings must be enabled with care; some devices may have hardcoded DNS servers (example Voice over IP devices) and may need to be allowed to talk to the dedicated DNS server(s).

Sumesh Shivdas, msumesh@hotmail.com

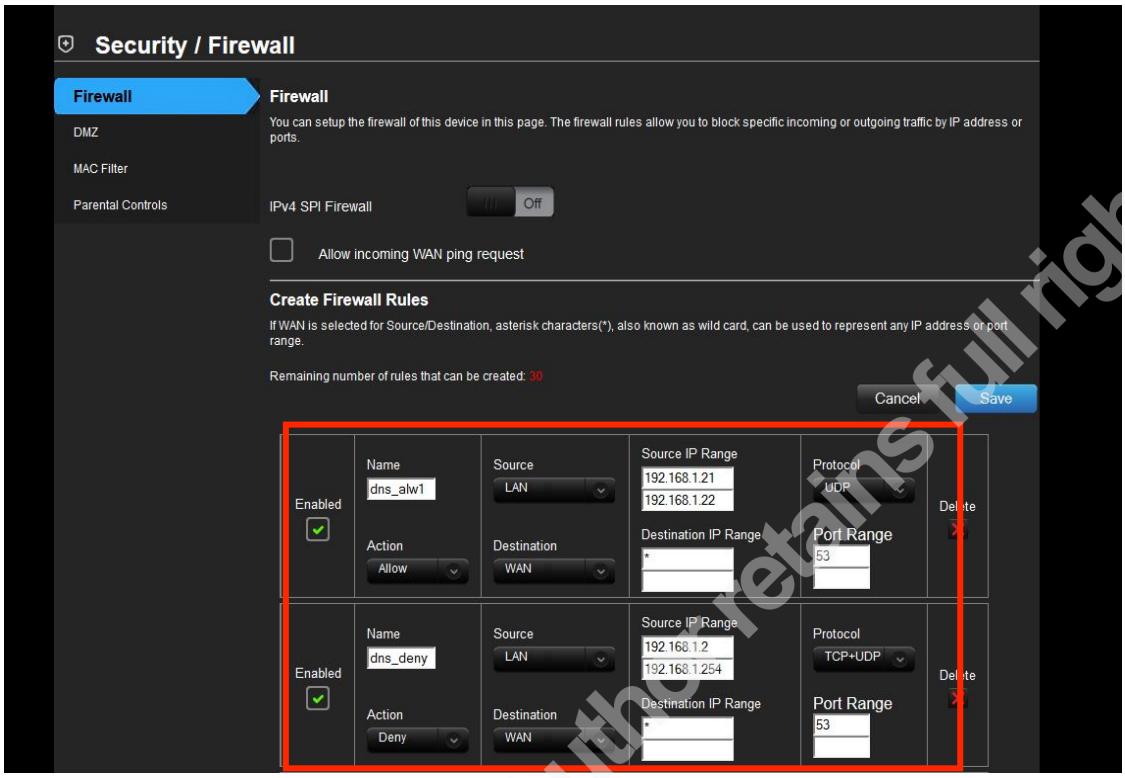



Figure 11: Firewall setup at the router

Using a firewall rule at the router prevents any client connected to the home network from bypassing the local DNS, and the configuration is central at one place. Advanced TCP/IP option changes can also be restricted locally on the system for a non-administrative user. The figure below shows how to do it on a Windows 7 system running the command "gpedit.msc" (group policy editor) as administrator. Select the **User Configuration -> Administrative Templates -> Network Connections -> Prohibit TCP/IP advanced configuration** option and set it to Enabled (default is "Not Configured")

Sumesh Shivdas, msumesh@hotmail.com

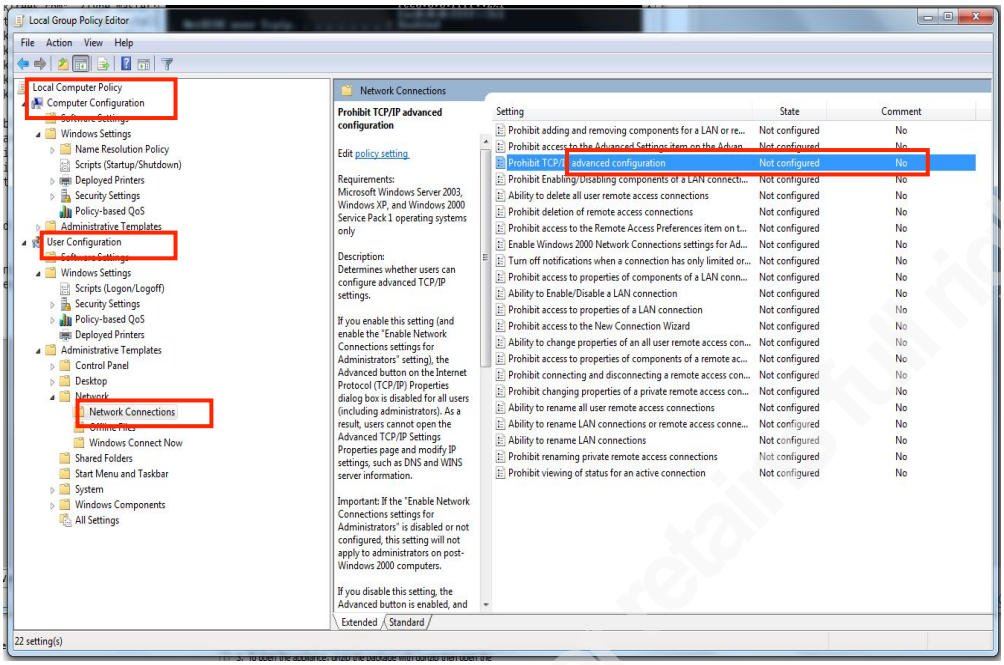



Figure 12: Disabling changes to Advanced TCP/IP properties

## 4. Solution Demonstration

### 4.1 DNS Query logs

Below are the query logs from the DNS server. Unlike OpenDNS, the local DNS server can report DNS queries performed by individual devices in the home network, which makes identifying the infected computer easier.

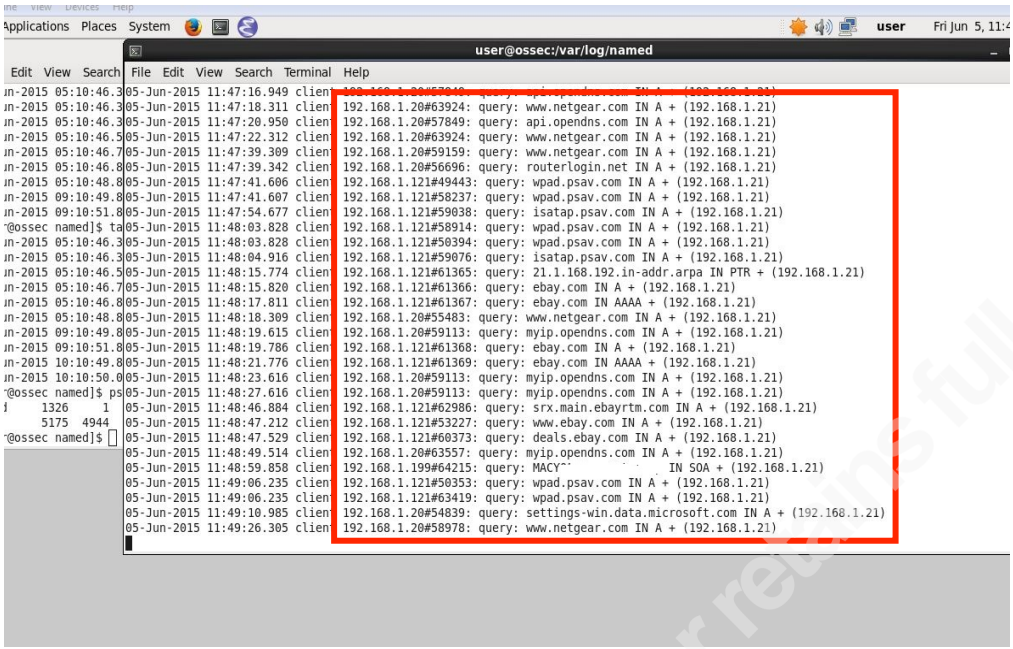



Figure 13: DNS Query logs

These query logs can be passed to OSSEC with the list of malicious and blacklisted site, and it can trigger alerts when any client attempts to connect to the known blacklisted server.

## 4.2 Solution Test

The figure below shows an attempt to connect to a known malware site being thwarted.

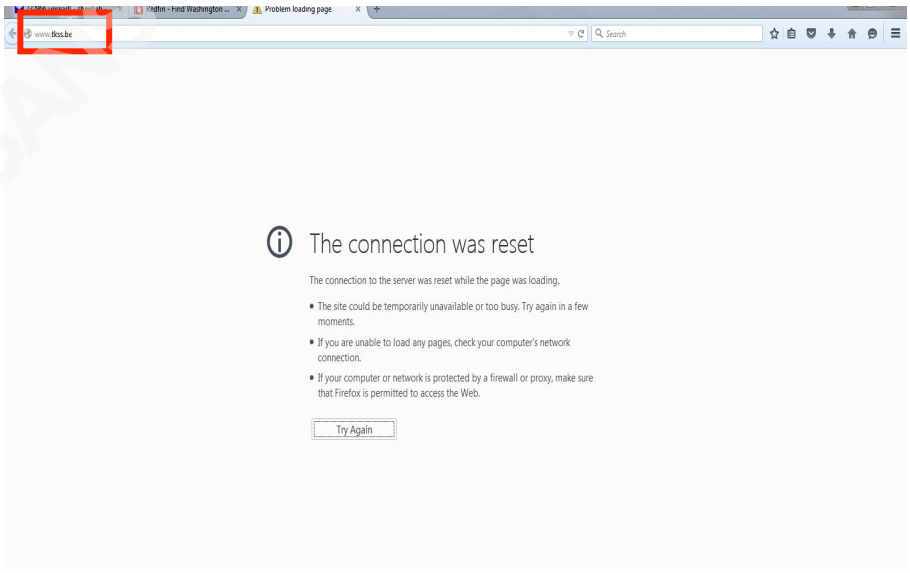


Figure 15: Attempt to connect to malware site thwarted

The figure below identifies the client (192.168.1.199) or MACY0138 (from figure 10), a laptop in the home network attempting to connect to the malicious site on 25th June at 18:36:24.


## 5. Conclusion

The solution is able to protect home systems from connecting to known malicious servers and personally blacklisted sites. Individual devices can easily be identified from the logs. Reviewing the logs periodically will help in fine-tuning the personal black list. Gratuitous advertising sites can easily be blocked by adding them to personal blacklist. The solution complements the protection provided by OpenDNS. In addition to blocking the sites based on categories, the user can configure egress filtering of unlimited sites. A Virtual Machine setup as mentioned in this design with a simple user interface for configuration update can help home users to protect their network from Malware. It would be ideal if home routers, which are becoming very powerful these days, could have the features as designed in-built in them.

Sumesh Shivdas, msumesh@hotmail.com

# References

[1] Barcena, M., & Wueest, C. (2015). Insecurity in the Internet of Things. Retrieved June 8, 2015 from http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/insecurity-in-the-internet-of-things.pdf

[2] Brown, B. (2011). How to stop e-mail spam, spyware, malware, computer viruses, and hackers from ruining your computer or network: The complete guide for your home and work. Ocala, Fla.: Atlantic Pub. Group.

[3] Doherty, S., & Tsapakis, N. (2015). Analysis of malware targeting the Boleto payment system. Retrieved January 8, 2015 from http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/boleto-malware.pdf

[4] Symantec Internet Security Web Threat Report. (2015). Retrieved June 8, 2015 from https://www4.symantec.com/mktginfo/whitepaper/ISTR/21347932_GA-internet-security-threat-report-volume-20-2015-social_v2.pdf

[5] Complete Malware Domain List. (2015). Retrieved June 8, 2015 from http://www.malwaredomainlist.com/hostslist/hosts.txt

[6] Bad domain List. Retrieved June 8, 2015 from https://zeustracker.abuse.ch/blocklist.php?download=baddomains

[7] Phishing sites. Retrieved June 8, 2015 from https://openphish.com/feed.txt

[8] Malware domain list. Retrieved June 8, 2015 from https://www.malwarepatrol.net/lists.shtml

[9] Network Address Translation, Frequently Asked Questions. Retrieved June 8, 2015 from http://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/26704-nat-faq-00.html

[10] OSSEC Server Virtual Appliance 2.8.1. Retrieved June 11, 2015 from http://www.ossec.net

[11] Oracle Virtual Box. Retrieved June 11, 2015 from https://www.virtualbox.org/wiki/Downloads

[12] Open DNS. Retrieved June 11, 2015 from https://www.opendns.com/home-internet-security

Sumesh Shivdas, msumesh@hotmail.com

Sumesh Shivdas, msumesh@hotmail.com