SANS PENETRATION TESTING

Use offense to inform defense.
Find flaws before the bad guys do.

Copyright SANS Institute
Author Retains Full Rights

This paper is from the SANS Penetration Testing site. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Techniques, Exploits & Incident Handling (SEC504)"
at https://pen-testing.sans.org/events/

Identifying Load Balancers in Penetration Testing

GIAC (GPEN) Gold Certification

Author: Curt Shaffer, cshaffer@gmail.com

Advisor: David Shinberg

Accepted: February 16, 2010

Abstract

A problematic situation exists when embarking on a penetration test where load balancers are present. The general goal of all penetration tests is to provide accurate information to the company requesting the test. This means that the person performing the test needs to be sure they are as thorough as possible. This will help provide the company with the most realistic insight into how their existing vulnerabilities increase the risk of their most valuable assets. The increasing use of load balancers for web applications and the web servers being used to serve the applications require the increase of high availability. This can cause an issue for penetration testers. When there is a load balancer in place of an asset a penetration tester will be analyzing in their scope, there is the issue that possibly only one of the systems may respond to the test queries. This would give misleading results because only one system would have actually been tested. Another possible issue is that different servers may respond for each run of a different tool. Such a result could in fact cause inconsistency in the testing if the patch levels or configurations are different for each system. This paper addresses four questions: 1) Why is this an issue for the penetration tester? 2) How can you tell if you may be hitting a load balancer in your penetration tests? 3) Is there anything that can do about it and, what tools can be used to assist in this search?

© 2010 SANS Institute, Author retains full rights.

© 2010 The SANS Institute Author retains full rights.

1. Introduction

More and more applications are moving to a web-based platform because there is a need to have applications that can run on multiple platforms without the need to write different code for each. People are using different operating systems and CPU architectures such as 32 or 64 bit. Being able to write code one time to support all of these platforms is invaluable. Businesses are becoming more reliant on their web presence to offer 24-hour access to their services and goods. Thus, it is becoming more important that these applications are highly available. Over the past several years companies have dedicated substantial resources to achieve this flexibility and to use the increased ability to become more productive. One of the first methods used to achieve this was to use DNS load balancing. Using DNS to achieve redundancy is probably the easiest way to give an appearance of load balancing. It then became apparent that a better way to load balance was needed because this method has some serious limitations. The major limitation to this type of load balancing was that the DNS servers do not know if a host that a resource record points to is up and ready to receive requests or not. If someone attempts to connect to a server in this case, the request will not be successful, giving the user an error or not responding properly. Another issue with this is that DNS servers tend to cache requests. If a person's DNS server has cached the record of the server that is down, the request will again fail.

As a further improvement dedicated devices were used to achieve high availability. These devices are often called load balancers or Application Delivery Network (ADN) devices. A load balancer or ADN is a system that listens for requests on a specific port, such as port 80 or 443 in the case of web services. It then analyses what servers behind it can process this request, then sends the request on unbeknownst to the user. In some cases there needs to be some sort of persistence in this communication. This persistence can be in the form of the network that requested the resource, a cookie that was placed on the user's system, the amount of requests the server is taking at the moment or based on user authentication. If a type of persistence is needed, the appliance does this well, when a request comes in matching the persistent value, it sends them to the same node that responded last time.

Two reasons web application owners want to use persistence is to manage the connection levels to their back end systems or to keep a user's preferences as in the case with an ecommerce shopping cart. They may also want to send certain users to a certain group of systems based on

Curt Shaffer, cshaffer@gmail.com

© 2010 SANS Institute, Author retains full rights.

© 2010 The SANS Institute Author retains full rights.

network or geographic location in order to provide the most prompt access to end users as possible.

Systems offering services online require high security. The reason they need high security is because these are assets that a company is choosing to expose to a large group, or everyone on the Internet. Some of these systems hold content in backend databases that are attractive to modern criminals. This information ranges from credit card numbers, social security numbers or health care information to name a few. With this in mind many companies are spending a lot of time and resources on testing and mitigating threats.

One common way of testing if a system is vulnerable to threats is though web application penetration testing. When testing applications for possible vulnerabilities, the tester will often send data, or requests to a URL that provides access to the application. These requests are made to ports that are exposed or available for a user to connect to. The web applications listen to these ports and create socket connections when requests are made. If the application responds in a particular manner, it can provide the tester with complete access to the backend of the system. This backend access can lead to leakage of personal information such as social security numbers, account numbers or sensitive company information. It can also offer a method for the tester to pivot though the network gaining access to other systems and information.

More important than finding the holes in the application is proper understanding of the likelihood of reoccurrence and reporting this information accurately to the application owner. This is where the issue of load balancing can enter the picture. Some of the issues that are found during a penetration test are host specific including lack of patching on the host or inconsistent code versions. If the systems are behind a load balancer, you might not know which node actually has the vulnerability.

This paper will review the different methods of load balancing, and explain how to recognize them. Then it will discuss what can be done about it so the problems that are outlined can be mitigated. This should lead the reader to knowing why this information is important to take into consideration when performing penetration tests and should arm testers with the techniques to know when load balancing is in place.

Curt Shaffer, cshaffer@gmail.com

© 2010 SANS Institute, Author retains full rights.

© 2010 The SANS Institute Author retains full rights.

2. DNS Load Balancing

One of the easiest methods of load balancing is through DNS. The RFC for DNS (RFC 1034) states that it is valid for an A record to contain multiple entries of IP addresses. (The Internet Engineering Task Force (IETF)). Most DNS systems support what is known as round robin DNS. This will allow each query for a hostname to return a different IP address. This gives the appearance of load balancing. It does in fact divide up the requests for a web page. There is a fundamental flaw in this approach as far are true load balancing is concerned.

With DNS load balancing, the DNS server is not capable of knowing if a host with an IP address that is listed for a particular name is up and ready to process requests. Thus, if a request for a page comes in, the DNS server responds with the next IP address in line if it's ready or not. This is far from ideal, and can lead to pages not being displayed when the IP address of the host is down is returned to a client requesting the record. While this does provide the function of load balancing, it should be noted that it is not providing high availability or any real type of redundancy.

There are some devices that use this technique along with other methods as well. The way they do this is by utilizing DNS to enter in a number of possible systems that can respond. In some cases these are actual hosts and in some cases, they are just local load balancers. The core systems that control the DNS records are a bit more intelligent than just DNS servers though. One such product is the F5 Global Traffic Manager (F5 Technologies). Such equipment is created for the sole purpose of providing redundancy and availability across a large geographic area. In many cases, this type of load balancing provides responses to resources based on the origination of the request. Companies such as Google and Microsoft use these technologies in delivering content and updates. One could only imagine how inefficient it would be if all requests from around the world going to Google went to Mountain View California, or if all request from around the world for Microsoft updates went to Redmond, Washington. This class of equipment is able to know via certain checks, such as ICMP, content requests or port availability, and determine if one of the systems is unavailable. When it sees that one of them is unavailable, it removes the entry from the DNS record response.

© 2010 The SANS Institute Author retains full rights.

DNS cache is how a DNS server attempts to make queries faster. It holds the same value that was received when it queried either the authoritative or root DNS zones. When a client asks for a record that has been cached by the DNS server, the DNS server can respond directly rather than having to request a new value from another source. There can cause an issue with this because this value may have changed, or the system that responds to the address that was given last time, may not be available now. This is fixed by using a low TTL value. The TTL or "time to live" value tells the requester how long the record is valid. If the record has a low TTL value that means that they should ask for an updated value again sooner. This does not always work but it can be effective if utilized properly. This type of configuration is outside of the scope of this document.

It is relatively easy to know if this type of load balancing is going on. If you ping the host from your system 2 times clearing your DNS cache in between commands as shown in figure 2-1 below. In this case the command that is used is "ping –a www.google.com". This will send an ICMP echo request to the resolved value of the name www.google.com. In this case that first value is 64.233.169.104. Then the next command issued, on Microsoft Windows, is "ipconfig /flushdns". This command clears the local DNS cache. When this is done, the next time a request for a resolution of that name, it will request it from DNS servers rather than just using the local cache. Next issue another request with the command "ping –a www.google.com". As you can see in figure 2-1, this time the value received is the IP address of 64.233.169.103:

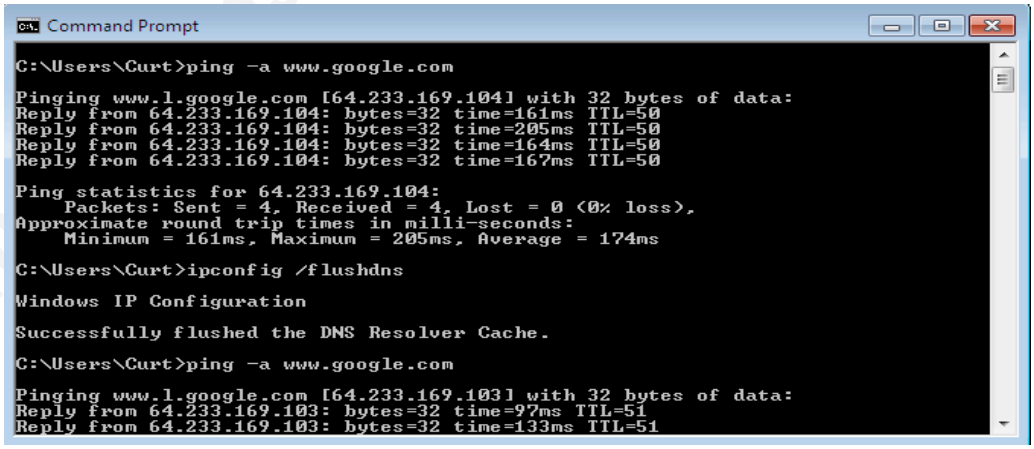

Figure 2-1: DNS Load Balance Example

Curt Shaffer, cshaffer@gmail.com

© 2010 The SANS Institute Author retains full rights.

Figure 2-1 shows that there is some DNS load balancing going on with www.google.com. This can also be seen by requesting the A record of www.google.com with nslookup as shown below. In this example the command that is issued is "nslookup". This will drop into the nslookup command window. From there the DNS server is set to a public DNS server. In this case the command that is typed is "server 4.2.2.2". This command tells the nslookup command line to use that specific server for resolution. Next, the value that is typed is the record of which to resolve. In this case the value that was typed is "www.google.com". The result shows six different IP address that are set to respond to the resource record at this time.

```
> server 4.2.2.2
Default Server:  vnsc-bak.sys.gtei.net
Address:  4.2.2.2

> www.google.com
Server:  vnsc-bak.sys.gtei.net
Address:  4.2.2.2

Non-authoritative answer:
Name:    www.l.google.com
Addresses:  74.125.53.106
          74.125.53.147
          74.125.53.99
          74.125.53.103
          74.125.53.104
          74.125.53.105
Aliases:  www.google.com

>
```

Figure 2-2: DNS A Records

Armed with this information, a penetration tester can now take necessary steps in changing their method of testing this particular web page. One approach is to use IP address rather than names, though we may still be hitting a virtual IP address on a load balanced system. The tester can also attempt to find a direct hostname to node mapping through further DNS enumeration. You can also ask the company's IT staff if any of the web applications are using load balancing techniques. This can be achieved by some manual or automatic discovery process or obtained from methods of social engineering if that is in scope of the test. Also, this can be accomplished by requesting the information in the scoping meeting of the penetration testing contract. The final option here is probably the best to fully ensure that we are giving accurate

© 2010 The SANS Institute Author retains full rights.

results, though depending on the type of test, this information may not be given as in a black box testing scenario.

3. Load Balancing Appliances

The next thing a penetration tester can come across in this scenario is the use of load balancing appliances. The use of load balancing appliances in the industry is becoming more and more of a reality. These devices provide companies with high availability and fail over capability for their web applications.

An issue exists when a penetration tester come across these types of appliances when performing penetration testing. This issue is the fact that when a request is being received and handled by such a device, the penetration tester cannot know if they are hitting the same target every time. This may lead to inconsistent results in the test. The general way that these appliances work is much like a proxy. A request is sent, for a web page as an example, to a DNS address which is translated to an IP address. The load balancing appliance is configured to respond to all HTTP requests to this IP address. The load balancer then makes a determination, based on load balancing methods, on the best server to send this request to at this time.

There are a number of different methods used in load balancing. Some of the most common are round robin, least connections, and cookie persistence. In a round robin configuration each subsequent request is sent to the next server in a list of available servers in the load balancer configuration. This is much like the DNS round robin in that each subsequent request is sent to the next system in line. The way that it differs from DNS round robin is that the load balance device knows if the members are available to process the request or not. If they are not, they are skipped in the rotation. This is the easiest configuration of load balancing and probably one of the most used. For a penetration tester, this is one of the more difficult methods to deal with. The reason is because there is no persistence, so each request could be or is hitting a new member server in the load balancing pool.

Curt Shaffer, cshaffer@gmail.com

© 2010 The SANS Institute Author retains full rights.

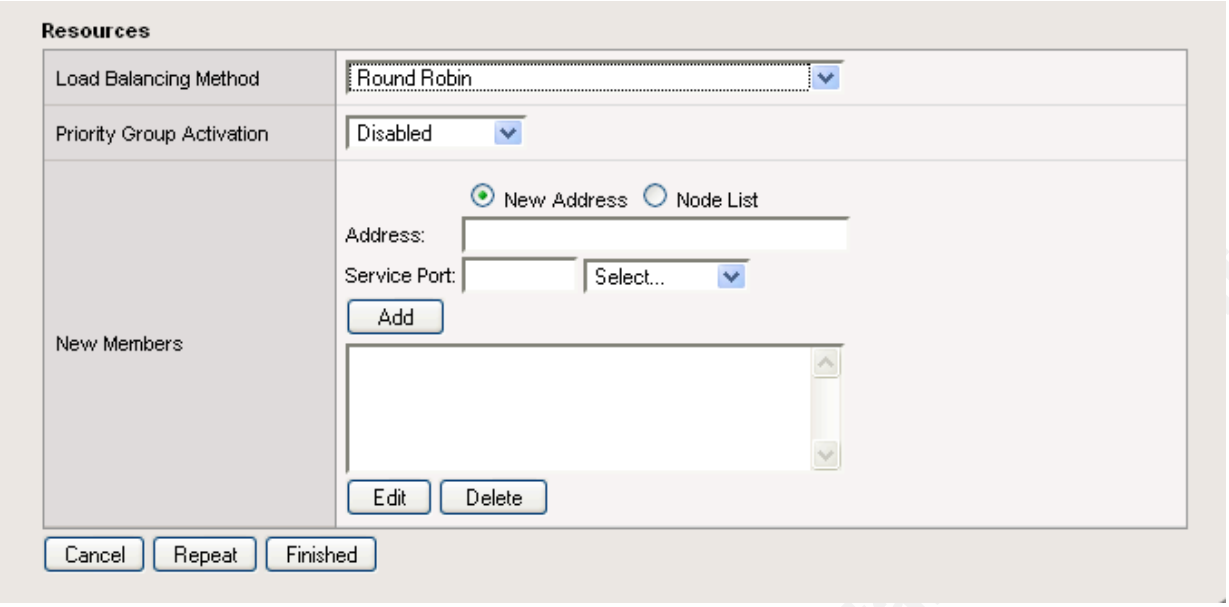

Figure 3-1: F5 Round Robin Load Balance Configuration

In least connection configurations, the subsequent request is sent to a server in the configured list which is currently serving the least amount of connections to the resource. These connections may also be weighted to favor certain servers over others. New servers may come online in the load balancing pool and the administrator may want the newer systems to handle more connection. This could pose an issue to the penetration tester because some systems my only process a few connections and be missed in the examination of the test.

Curt Shaffer, cshaffer@gmail.com

© 2010 The SANS Institute Author retains full rights.

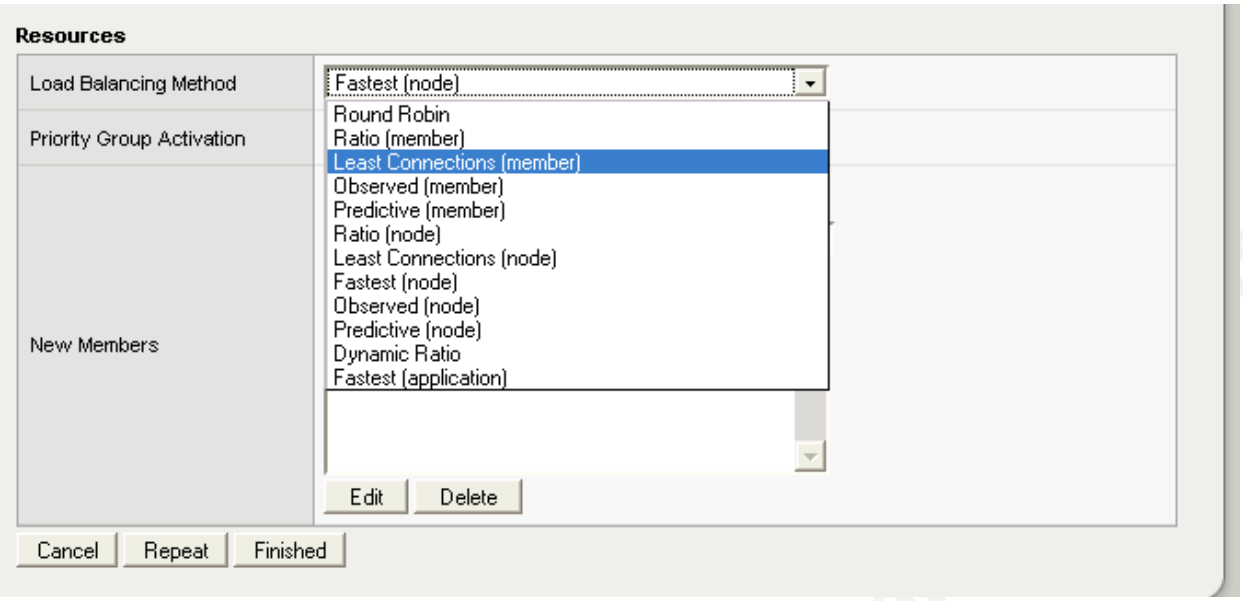

Figure 3-2: F5 Least Connections Example

Finally cookie persistence is used in cases such as ecommerce where the company may want the user to stay on one server if possible to keep information such as shopping cart items or preferences for the lifetime of the session. These cookies can be created by the load balance appliance or the server behind them. In some cases, this will reveal the existence of load balancing. This method may not be that bad for a penetration tester if they do not clear out the cookie value set, thus they will be hitting the same system every time. On the other hand, this is not a complete test as there is normally a number of members behind the appliance that need to be examined as well.

Curt Shaffer, cshaffer@gmail.com

© 2010 The SANS Institute Author retains full rights.

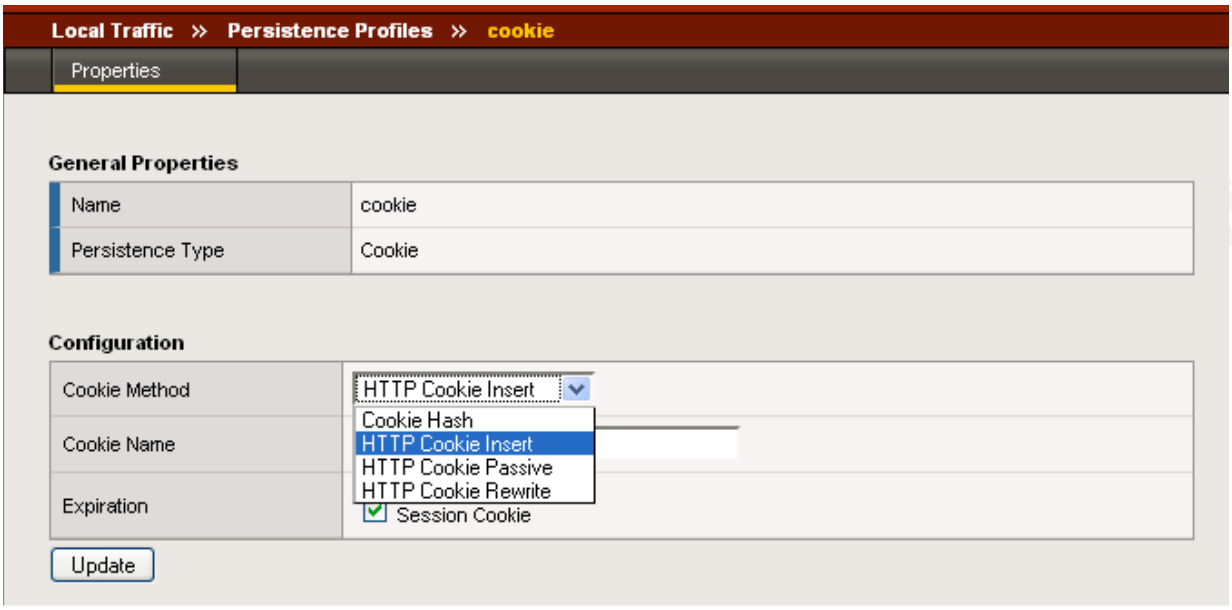

Figure 3-3: F5 Cookie Persistence Configuration Example

Although, there are many other methods used, these are the only options we have discussed to illustrate the point. If we are sending requests to an IP or DNS address for a resource, and that request is being sent to a different server for each or a number of subsequent requests, the penetration tester will get a mix of results.

4. Operating System Load Balancing

Load balancing can be achieved with most modern operating systems as well. The methods are similar to the load balancing appliances, but their configurations are often less granular. Microsoft includes NLB (Network Load Balancing) with their operating system by default. According to their TechNet article on the subject "Network Load Balancing is a clustering technology offered by Microsoft as part of all Windows 2000 Server and Windows Server 2003 family operating systems" (Microsoft, 2004).

Other operating systems such as Linux or any flavor of BSD also have options for load balancing. There has been a lot of development of this type of functionality in the open source

© 2010 The SANS Institute Author retains full rights.

world. The Linux Virtual Server (LVS) is one of the load balancing options on Linux operating systems. This is not to be confused with high availability (HA) or operating system load balancing. LVS is a smart load balancing application much like the load balancing appliances such as F5 or Radware. There are a lot of new systems on the market that are running Linux under the hood that are utilizing either this technology or some derivative. Some flavors of Linux also have built in technologies like Microsoft such as the Red Hat Cluster Suite which is included in their Enterprise software as well as the open community version of Centos. (Greenseid)

5. What does load balancing mean for pen testers?

Some may ask, what does load balancing mean to pen testers, and why should they care? This has been the reason for the writing of this paper. There are a number of concerns that a pen tester should have when performing a penetration test for a company. The main goal of a penetration test is to provide the client with a valid explanation of the potential security holes in their network. The information that is given to them must be accurate as the recommended fixes can cost a lot of money and time for a company to implement. If the problem that is presented to them is not really exploitable, then this time and/or money may be spent for nothing. Also, if the results of a penetration test say that a client's system(s) are not vulnerable when they really are, then the company can suffer a breach which could lead to many issues legally and operationally.

One of the first steps in a penetration test is reconnaissance. This is where the penetration tester will start to enumerate the potential targets on the network in the scope. This will generally start by analyzing the IP address space and services associated with each system in the scope of the engagement. If any of the servers or services that are being enumerating are behind some sort of load balancing, then there may be inconsistent results. If the pen tester is not careful in their examination of the results of this enumeration, it could lead them to miss important services or systems.

After systems and services on the network in scope have been identified, steps are taken to find potential vulnerabilities in them. This will often be done in a number of ways which may include identifying versions of applications or operating systems, attempts to identify patch level

© 2010 The SANS Institute Author retains full rights.

or existence for applications and operating systems, and potential response analysis through fuzzing the systems and services.

As these steps are taken, careful notes should be kept to come back to when we enter the exploitation phase of the penetration test. If the penetration tester is running their scans and fingerprinting against systems which may be behind a load balancer, they may be only hitting certain systems in the case of a persistence method, or may hit different systems with each pass of the scan(s). In either of these cases, one system that is hit may have a valid vulnerability which will be noted, but this may be the only system out of the cluster that has that particular vulnerability.

One of the final objectives of a penetration test before the report generation is to attempt to exploit the vulnerabilities they have found. The pen tester will normally go back at this time through the detailed notes they have taken and attempt different methods of exploiting a found vulnerability. This may take a long time depending on the requirement of the test. It may also take the testing of a number of different methods until finding one that works.

This process is normally stressful enough in some cases when exploits need to be modified to actually exploit a known vulnerability consistently. If you add a load balancer to this equation, you may be thinking that the exploit is not working, when it fact, it may be that you are hitting a different system which may not have the vulnerability. This could lead to a lot of wasted time that could have been eliminated if it was known that load balancing was in place. It is important then, that it is known if this is the situation, and to identify these things before the time is spent.

6. How to identify load balancing

There are a few ways in which to tell if the request is hitting such a load balancing system. One way earlier in the discussion of DNS load balancing. A simple series of ICMP messages sent interspersed with flushing out DNS cache would easily reveal if this was the case. If, DNS load balancing is present, you could attempt to do your testing on the IP addresses rather than the DNS name in order to lower the chance of running into any of the issues discussed.

© 2010 The SANS Institute Author retains full rights.

Another way is due to a mis-configuration of the load balancing devices. Some devices put values into cookies that identify them clearly. For example, I logged into the community section at http://communities.vmware.com. After logging in I took at look at the cookie value that was set by their server. Figure 6-1 shows the relevant portions of the cookie:

```
BIGipServercommunities-prod-pool1379823882.36895.0000communities.vmware.com
```

Figure 6-1:F5 Cookie from http://communities.vmware.com

Notice the entry of BIGipServer in this cookie. This would let us know that VMware uses the F5 brand of load balancer at least for their community site, it also lets us know that they are probably using some sort of cookie based persistence for users who log in. In some cases this value is actually just an encoded value which represents the pool name and the IP address and port number of the responding system behind it. This is a known bug and has been addressed. If the penetration tester is using Tenable's Nessus for one of their testing tools, there is a NASL script number 20089 that can be used to identify this information. (Tenable Network Security)

Another thing that can be checked in attempt to see if there is load balancing going on is revealed in the IP ID field in the response from the server in question. Hping3 is able to craft network packets and is used by many penetration testers in examining behavior of certain systems. The following output has been captured from a series of hping3 packets being sent to www.microsoft.com. The first command is telling hping3 to craft a SYN packet, with the –S, to port 80

```
hping3 www.microsoft.com -S -p 80
HPING www.microsoft.com (eth0 207.46.193.254): S set, 40 headers + 0 data bytes
len=46 ip=207.46.193.254 ttl=242 id=64800 sport=80 flags=SA seq=0 win=8190 rtt=106.0 ms
len=46 ip=207.46.193.254 ttl=242 id=6989 sport=80 flags=SA seq=1 win=8190 rtt=106.3 ms
len=46 ip=207.46.193.254 ttl=242 id=45431 sport=80 flags=SA seq=2 win=8190 rtt=106.9 ms
  len=46 ip=207.46.193.254 ttl=242 id=27555 sport=80 flags=SA seq=3 win=8190 rtt=105.5 ms
len=46 ip=207.46.193.254 ttl=242 id=19407 sport=80 flags=SA seq=4 win=8190 rtt=108.1 ms
len=46 ip=207.46.193.254 ttl=242 id=17401 sport=80 flags=SA seq=5 win=8190 rtt=106.5 ms
```

Figure 6-2: HPING3 IPID Example Output

Curt Shaffer, cshaffer@gmail.com

© 2010 The SANS Institute Author retains full rights.

As seen from the output the IPID field, listed as just id= in the results, not only is the number drastically different, but the values are very much out of order. The IPID value is incremental from each system. You might expect not to see a liner pattern for a busy site such as Microsoft; however, you would expect that each value is larger than the previous if you were hitting the same system every time. In the results this is not the case. There are lower values for later packets listed which would lead us to believe that www.microsoft.com is using load balancing of some sort.

7. What can be done about load balancing?

After discussing all of the potential ways in which load balancing is being used, one might wonder what can be done about it. As discussed, this can cause problems or inconsistencies in penetration testing. With that in mind, after finding out that requests seem to be hitting load balancing of some sort, a penetration tester would want to take proper steps in ensuring that the test results are not only valid, but helpful for the target company.

In dealing with each of these types of load balancing, one thing normally remains true. Each system behind each of these methods normally have an IP address that is truly their own and not part of the load balancing. These IP addresses are normally used to perform maintenance on the system, directory service identification, and in some cases identification of pool association in the load balancing appliances. When looking at DNS load balancing, it is pretty simple to see what IP addresses are available for each host involved. By using nslookup for the A record for the DNS name, it is apparent as in the example below. Again the command "nslookup" is issued which will drop the command line into the nslookup command line tool. At this time the value of www.google.com is entered. This method of gathering information is normally done in the reconnaissance portion of a penetration test. However, this information along with all information gathered in that stage can and will be used elsewhere in the test. This value is asking nslookup to ask the DNS servers to return the A record results for the value. The response shows multiple addresses as shown in Figure 7-1 below:

© 2010 The SANS Institute Author retains full rights.

```
> www.google.com
Server:  vnsc-bak.sys.gtei.net
Address:  4.2.2.2

Non-authoritative answer:
Name:    www.l.google.com
Addresses:  72.14.204.104
          72.14.204.147
          72.14.204.99
          72.14.204.103
Aliases:  www.google.com

>
```

Figure 7-1: Google A Record Result Example

From the example above we use nslookup looking for A or host records. This is accomplished by typing set type=a in the nslookup command line window as shown in Figure 7-2 below:

```
C:\Users\Curt>nslookup
Default Server:  UnKnown
Address:

> server 4.2.2.2
Default Server:  vnsc-bak.sys.gtei.net
Address:  4.2.2.2

> set type=a
> www.google.com
Server:  vnsc-bak.sys.gtei.net
Address:  4.2.2.2

Non-authoritative answer:
Name:    www.l.google.com
Addresses:  72.14.204.104
          72.14.204.147
          72.14.204.99
          72.14.204.103
Aliases:  www.google.com

>
```

Figure 7-2: NSLOOKUP Set Type Example

The results show that there are four separate IP addresses that respond to www.google.com. It should be noted that Google appears to use multiple methods of load balancing so this example may change, but it shows the point. This does not reveal if these IP addresses are load balancing devices as well or just multiple addresses configured to respond to the request. We could assume, knowing the size of Google, that this is probably addresses of load balancers, but that may not be the case for every penetration test.

The next issue discussed was cookie configuration or mis-configuration. This method is a little more difficult to get around by the results that are achieved from looking at the cookie that

© 2010 The SANS Institute Author retains full rights.

is presented to the browser. If the cookie is really mis-configured, the pool name or some other identifying value may reveal possible IP addresses for the host. In most cases this will not be true. With that in mind, the only real way to get around this would be to attempt to do some reconnaissance on the DNS of the domain in attempt to find a direct host name to IP address mapping. This also is a problem in that network address translation (NAT), will generally be performed on hosts, and the internal IP addresses will not be available to the general public. If the penetration tester is successful in getting to the inside of the network, this may be more possible.

The final thing that the penetration tester can do to get around the potential issues of load balancing is to simply ask the company IT staff if any load balancing is being done. In some cases, they may not know. If that is the case, the tester can use some of the techniques discussed here to gather what they feel might or is being load balanced. With that information it can be taken to the IT staff and a request can be made to either get the IP address that is mapped directly to each host or they may be able to set up a series of pools or settings in which each host can be addressed directly. Another technique that can be done is by social engineering. A call can be made to certain individuals in charge of network equipment posing as a developer having issues or possibly the posing as a network engineer requesting information from the application development team. A Third option could be calling and pretending to be of a certain load balance device vendor asking about what solutions they currently use. They could then add details such as if these devices are facing the Internet, only internal or both. The staff member may also tell the social engineer that they only use something like DNS load balancing, giving useful information to the penetration tester. Social engineering is not always in scope, so this needs to be used with caution and only if explicitly allowed in the test scope.

8. Conclusion

This paper discussed an issue that exists for penetration testers with the introduction of load balancing in network infrastructures today. Each method was looked at in detail, so recognizing them would be possible. The paper started with a discussion of load balancing in general and why it is becoming more of an issue.

Curt Shaffer, cshaffer@gmail.com

© 2010 SANS Institute, Author retains full rights.

© 2010 The SANS Institute Author retains full rights.

The next topic that was covered was DNS. It was shown by itself; this is not a robust load balancing method. Some companies still use it despite this fact. If DNS load balancing is added to a load balancing appliance, it can add a level of functionality that DNS load balancing by it does not have. DNS load balancing is easy to see with the nslookup or dig commands.

Load balancing appliances are more commonly used for web applications. These appliances can be configured to use cookie persistence, least connections or round robin in their methods. Recognizing these types of methods is a little more difficult than DNS load balancing. By analyzing the cookies that may be presented, often will reveal information leading us to believe that a load balance appliance is in place. Another method is to use a tool, such as HPING3. When using HPING3, it was evident that www.microsoft.com was using load balancing by reviewing the IPID field in the results.

The discussion then brought up the issues with load balancers when present in a penetration test. The main issues are that the tester may get inconsistent results in their vulnerability assessment, payload or exploitation creation, and reporting validation for the company at hand. Thus, it is important that care is taken during a penetration test to recognize when some sort of load balancing is being used.

Armed with this information, it should be easier for penetration testers to recognize when they may be hitting a load balancing method in their penetration tests. These easy steps could easily be incorporated into a Perl script or possibly a NMAP extension, or Nessus plugin so that the penetration tester would have an automated way to know load balancing is being done. This should be a tool or steps in any penetration tester's arsenal. This is an easy addition to the processes already followed which could help false positive reduction and/or increase the quality of the end report for the customers.

Curt Shaffer, cshaffer@gmail.com

© 2010 SANS Institute, Author retains full rights.

© 2010 The SANS Institute Author retains full rights.

9. References

F5 Technologies. (n.d.). *F5 Technologies*. Retrieved January 18, 2010, from F5 Technologies: http://www.f5.com/products/big-ip/product-modules/global-traffic-manager.html

Greenseid, J. (n.d.). *LCIC*. Retrieved January 3, 2010, from Load Balancing Linux Clusters: http://lcic.org/load_balancing.html

Linux Virtual Server. (n.d.). *Linux Virtual Server*. Retrieved January 18, 2010, from Linux Virtual Server: http://www.linuxvirtualserver.org/index.html

Microsoft. (2004, February 10). *Microsoft TechNet*. Retrieved January 3, 2010, from Windows Server TechCenter: http://technet.microsoft.com/en-us/library/cc780254%28WS.10%29.aspx

Tenable Network Security. (n.d.). *Tenable Network Security*. Retrieved January 31, 2010, from http://www.nessus.org/plugins/index.php?view=viewsrc&id=20089

The Internet Engineering Task Force (IETF). (n.d.). Retrieved January 18, 2010, from The Internet Engineering Task Force (IETF): http://www.ietf.org/rfc/rfc1034.txt

Curt Shaffer, cshaffer@gmail.com

© 2010 SANS Institute, Author retains full rights.

Last Updated: May 26th, 2013

Upcoming SANS Penetration Testing

SANS PENETRATION TESTING

Click Here to {Get Registered!}

Community SANS SEC560 Bergen	Bergen, Norway	May 27, 2013 - May 31, 2013	Community SANS
Mentor Session - SEC504	Seaside, CA	May 30, 2013 - Aug 01, 2013	Mentor
SANS Malaysia @ MCMC 2013	Cyberjaya, Malaysia	Jun 03, 2013 - Jun 08, 2013	Live Event
SANS Pen Test Berlin 2013	Berlin, Germany	Jun 03, 2013 - Jun 08, 2013	Live Event
SANS vLive - SEC542: Web App Penetration Testing and Ethical Hacking	SEC542 - 201306,	Jun 03, 2013 - Jul 10, 2013	vLive
Mentor Session - SEC560	Owings Mills, MD	Jun 04, 2013 - Aug 27, 2013	Mentor
Mentor Session - SEC504	Omaha, NE	Jun 04, 2013 - Aug 06, 2013	Mentor
Mentor Session - AW - SEC560	Kedron, Australia	Jun 06, 2013 - Jul 25, 2013	Mentor
Community SANS Ottawa @ CANHEIT	Ottawa, ON	Jun 08, 2013 - Jun 09, 2013	Community SANS
Community SANS Paris @ HSC - SEC542 Spring	Paris, France	Jun 10, 2013 - Jun 14, 2013	Community SANS
Mentor Session - SEC 560	Sau Paulo, Brazil	Jun 11, 2013 - Jun 29, 2013	Mentor
SANSFIRE 2013	Washington, DC	Jun 14, 2013 - Jun 22, 2013	Live Event
SANSFIRE 2013 - SEC504: Hacker Techniques, Exploits & Incident Handling	Washington, DC	Jun 17, 2013 - Jun 22, 2013	vLive
Community SANS Paris @ Lexsi - SEC560 Societe Generale	Paris, France	Jun 17, 2013 - Jun 21, 2013	Community SANS
SANS vLive - SEC617: Wireless Ethical Hacking, Penetration Testing, and Defenses	SEC617 - 201306,	Jun 24, 2013 - Aug 14, 2013	vLive
Community SANS Albany	Albany, NY	Jun 24, 2013 - Jun 29, 2013	Community SANS
Community SANS Paris @ HSC - SEC560 - Summer	Paris, France	Jun 24, 2013 - Jun 28, 2013	Community SANS
Community SANS Paris @ Lexsi - SEC504 Spring	Paris, France	Jun 24, 2013 - Jun 28, 2013	Community SANS
Community SANS Charlotte	Charlotte, NC	Jun 24, 2013 - Jun 29, 2013	Community SANS
Mentor SEssion - AW- SEC 660	Shanghai, China	Jun 25, 2013 - Jul 30, 2013	Mentor
SANS Canberra 2013	Canberra, Australia	Jul 01, 2013 - Jul 13, 2013	Live Event
St. Louis Community College	St Louis, MO	Jul 08, 2013 - Jul 13, 2013	Community SANS
SANS London Summer 2013	London, United Kingdom	Jul 09, 2013 - Jul 16, 2013	Live Event
Mentor Session - SEC 504	Gainesville, FL	Jul 09, 2013 - Aug 08, 2013	Mentor
Mentor Session - SEC504	Niceville-Eglin Afb, FL	Jul 09, 2013 - Aug 22, 2013	Mentor
Mentor Session - SEC 504	Minneapolis, MN	Jul 11, 2013 - Sep 12, 2013	Mentor
SANS Rocky Mountain 2013	Denver, CO	Jul 14, 2013 - Jul 20, 2013	Live Event
Mentor Session - AW - SEC 504	Hanover, MD	Jul 17, 2013 - Jul 24, 2013	Mentor
Mentor Session - SEC 560	Denver, CO	Jul 18, 2013 - Sep 19, 2013	Mentor
SANS Mumbai 2013	Mumbai, India	Jul 22, 2013 - Jul 27, 2013	Live Event
Mentor Session - SEC 504	Dallas, TX	Jul 23, 2013 - Sep 24, 2013	Mentor