Global Information Assurance Certification Paper

GIAC Security Essentials Certification (GSEC) Practical Assignment 1.4b Option 1

Title: **Quarantining DHCP clients to reduce worm infection risk.**

Descriptive-title: A method for checking for worm infected machines connecting via DHCP to an organization's networks.

Author: Paul Blackburn

Table of Contents

1 Abstract

"*An ounce of prevention is worth a pound of cure"*

Today's highly connected computer systems are vulnerable to being attacked by "worm" programs which attempt to exploit software vulnerabilities and spread themselves between computers connected over networks.

"*A computer worm is a self-replicating computer program,*"
(From Wikipedia, the free encyclopedia [10])

Most of these worms are malicious and cause serious problems for computer users.

Estimates for the cost of recovery from damage done by computer worms ranges from the tens of thousands to the millions of dollars [1,9,12] per worm attack.

"Estimated cost of damage caused by the Morris Worm:$15.5 million"
(From: ^Shift [1])

"Bank of America Corp., one of the nation's largest banks, said many customers could not withdraw money from its 13,000 ATMs because of technical problems caused by the attack"
(From: CNN [9])

"The cost of this epidemic, including subsequent strains of Code-Red, is estimated to be in excess of $2.6 billion"
(From "Code-Red: a case study on the spread and victims of an Internet worm" [12])

This paper outlines an approach of preventing machines with infections from being allowed to connect using DHCP [7] into working networks by introducing a quarantine network that shares the same existing physical network already in use.

The focus of this paper is checking for worm problems on DHCP client machines rather than machines with static hostnames and static IP addresses.

This is presented as another layer in the "Defense in depth" approach to protecting an organization's computer resources.

There is no single "silver bullet" that will solve all the problems of worms causing damage but using several techniques including the one described in this paper will collectively reduce the risk.

This paper does not address the problem of dis-infecting a worm infected machine.

2 Introduction

Computer worms are a real problem for users of computer networks because they add extra useless network traffic and often damage the integrity of software and files on a machine.

Often, worms reveal their presence because of anomalous network behaviour. For example: a sudden and overwhelming increase in network traffic that makes normal work difficult or impossible.

If your computer becomes infected and you want to be very sure you have removed all the worm code and other possible problems you may well decide to completely re-install from a known and trusted installation process. This is an expensive process because of the disruption to your work and the time it takes to accomplish.

The prevention method outlined in this paper can be implemented using low-cost technology such as a Linux based DHCP [6] server and some Perl or shell scripting to co-ordinate the operation of the new quarantine and existing DHCP servers.

A benefit of this approach is that the process can be adapted to a particular organization's local needs. For example, it may be that a vulnerability port scan is not considered necessary or that an additional step is needed. These needs can be addressed by the scripting implemented at each site.

3 Worm infection routes

Worms can be introduced into a network by many routes [6]. Here are some:

a) Portable computers being moved between the Internet and an organization's private network.

b) A machine which has both an Internet connection and a VPN connection into an organization's Intranet simultaneously.

c) An email attachment that is automatically executed when the attachment is opened by an unsuspecting user.

d) A downloaded software package that contains hidden worm code.

4 Experience with CodeRed and Nimda worms

Analysis of the logged activities of CodeRed and Nimda worms by using honeypot techniques by the author shows the proportions listed in tables 4.1 and 4.2 (below) of static and dynamic IP addresses exhibiting behaviour of worm infection.

These figures are derived by classifying the client machine as either dynamic or static according to the DNS hostname. DHCP generated hostnames followed a convention that made them easy to identify. Data was collected using “dummy” web server log files that recorded the worm attack signatures of CodeRed and Nimda worms.

Note that in the time frame this data was collected there were other changes such as the growth in use of Dynamic DNS (DDNS) [7] which allows DHCP client machines to use static hostnames.

The following figures are not precise and not 100% representative of all activity but are presented to show that a significant proportion of worm attacks originated from DHCP client machines.

Table 4.1

CodeRed honeypot data showing percentage of DHCP and static hostnames:

Year	% DHCP	% static
2001	54	46
2002	50	50
2003	55	45

Table 4.2

Nimda honeypot data showing percentage of DHCP and static hostnames:

Year	% DHCP	% static
2001	46	54
2002	43	57
2003	29	71

5 Trusting DHCP clients

In my experience, most DHCP servers are configured to hand out full network access to any machine that is plugged into a network. This is a very trusting thing to do. There are no checks.

This leads to some questions:

1. Can we be sure that a DHCP connected machine is not worm infected?

2. Are all DHCP clients bona fide valid machines to connect to our network?

3. If a DHCP client was worm infected, what would be the risk to the network

4. Should we allow any unchecked machine to have full network access?

5. Would it be beneficial to run some basic checks and tests on a DHCP client before it is allowed full network access?

6. If we were to do some checking before allowing a machine to join our network, what would be an acceptable duration for these checks?

There is a balance to be found between being careful about security and not imposing a system that hinders users from doing normal work.

In a work environment that is highly conscious of security and determined to prevent problems as early as possible it may be appropriate to be very thorough about checking for potential problems before allowing DHCP client machines from accessing networks. In less rigorous work environments it may be acceptable to do minimal checking prior to granting full network access.

To achieve this balance, the following factors need to be considered:

- Cost of implementation: how much do we have to pay in manpower, hardware, and systems maintenance?
- Cost of potential damage if not implemented.
- Inconvenience: how long will users be prepared to wait for a connection to the w-net?
- Value of collecting honeypot data from multiple quarantine DHCPservers?
- Value of enforcing a security policy on DHCP clients?

The answers to some of these questions will be different in different organizations and some would form part of the organization's Security Policy.

6 Preventing infected DHCP clients from joining a network

The basic idea is not to allow DHCP client machines that are infected with a worm from being connected to an organization's normal working network.

The author's personal experience has shown that a good proportion of worm infected machines are using DHCP addresses. It is all too easy for someone to move a portable machine from an Internet connection to a private network (e.g. corporate, or university network) without knowing their machine has a worm infection.

If a machine can be checked before it is allowed to join the normal working network then this would block off one of the routes for infection that may be exploited by a worm to reach a private network.

In addition, checks for other problems can be done at the time a connection is requested. For example, tests could be done to identify whether a machine needs to have software updates applied.

This preventative approach provides for another layer of protection in the strategy of "Defense in depth". By itself, this method will not prevent all worms but used in conjunction with other defensive methods it can help restrict access to an organization's network of computer systems.

7 Method

The idea is to introduce a two stage DHCP process.

stage 1 – quarantine, checking for worm infection and other problems
stage 2 - access to network for machines deemed OK after checks

This can be achieved today with two separate DHCP servers using suitably written local scripts which communicate MAC addresses between the DHCP servers and manage the quarantine checking of new clients.

The first, or quarantine, stage DHCP server provides very short lifetime IP address leases in a private unrouted subnet [5].

The second stage DHCP server only serves IP addresses to machines with known MAC addresses (obtained from the quarantine DHCP server). This second stage DHCP server provides IP addresses from the "normal" working network of the organization. This allows client machines that have been through the quarantine process to have access to the organization's network resources.

An interesting property of this scheme is that it should be easy to implement a “fast track” process that enables known MAC addresses to be registered with the second stage DHCP server and thus by-pass the quarantine step.

This “fast track” process would have to be a carefully managed process and may not even be needed at most organizations. It could be compared, by analogy, with “fast track” immigration procedures at some international airports.

Also, it could be possible to revoke a registration and remove a MAC address from the second stage DHCP server so that a particular client’s IP address lease is not renewed thus causing it to be processed by the quarantine DHCP server again.

7.1 Stage 1

The quarantine DHCP server serves newly connected machines but it does so using a private "unrouted" network range [5] which is not the normal network that users connect to do in order to work with the organization’s networked resources.

We will refer to these networks as the quarantine “q-net” and the work “w-net” networks.

For example, if the organization's normal work network was 11.0.0.0/8 then 10.0.0.0/8 could be used as the quarantine (or “q-net”).

Any newly connected machine would be given an IP address in the q-net by the quarantine DHCP server effectively placing that client machine in quarantine.

This means that the new client would have an IP address and be able to network connect with the quarantine DHCP server but it would have no routing information to reach other machines in the normal w-net.

Such a new machine will be instructed to route all its network traffic via the first stage DHCP server. The quarantine DHCP server will also be running a DNS server for the q-net and a web server.

[side note: it is possible for two networks to operate on the same physical network. By having a quarantine network sharing the same wiring as the work network it is possible to isolate any worm infected machines by confining them to the logical quarantine network IP address space. end-side note]

The quarantine DHCP server also checks the newly connected machine using techniques including:

1. Identifying any active attacks using honeypot techniques.
   For example, using IBM Zurich's "Billy Goat" system [2]
   which can "listen" on "virtual" IP addresses in the
   whole of the quarantine subnet (e.g all of 10.)

2. Vulnerability scanning the newly connected machine.
   This is done to identify any problems in the newly connected
   machine such as vulnerability to buffer overflow attacks
   Tools like Nessus [3] (command line mode) could be used for this.

During quarantine checking (see also “Acceptance testing” below), if the user makes any request for web access via a browser, then a web page with a message such as the following would be displayed:

“Quarantine checking your machine, please wait"

On the completion quarantine checks, this message would change.

“Quarantine checking completed. Pass OK.”
“Please wait while your machine is transferred out of quarantine.”

If the newly connected machine fails the quarantine checking then its MAC address and other data are logged. This information can be sent via a systems management reporting system for the attention of the local site network administrators who are normally able to identify physical ports associated with MAC addresses and would (depending on the capability of their network hardware) be able to disable a physical port (if their organization’s security policy required that action).

If the user tries to use a web browser then whatever
URL is tried they will always receive the same web page:

“Quarantine checking completed. Failed acceptance test”
“REASON: your machine seems to be infected with worm: $worm"
“Please contact your help desk to correct this problem."

This could be achieved by a combination of the web server and DNS server running on the quarantine DHCP server with the DNS server redirecting all lookups to the IP address of the quarantine web server.

Note that the client IP address lease time defined for the quarantine DHCP server is defined to be a short time (dependent on the time checks take to run: possibly 2 minutes or less).

7.2 Stage 2

Once a newly connected machine has been cleared as "OK" by the first stage DHCP server, its MAC address is sent to the w-net DHCP server.

The second DCHP server serves "real" addresses but only to MAC addresses that have been checked by the quarantine DHCP server.

The w-net DHCP server would handout IP addresses with longer lease times (eg 2 hours).

If a machine that has been cleared OK leaves the network then its (second stage) DHCP lease on IP address expires and the MAC address is considered "not OK". A “MAC address management script” running on the w-net DHCP server will mark as “do not serve this MAC address” in the w-net DHCP server configuration file and communicate that MAC address back to the quarantine DHCP server in order for this MAC address to be marked “serve this MAC” on the quarantine DHCP server.

So, after disconnection, the w-net DHCP server will not serve that MAC address unless it has been checked again by the quarantine DHCP server.

This behaviour is configurable and could be adapted to meet the needs of an organization. For example, it may be better to hold on to the MAC address in the w-net DHCP server for a period of time (maybe 30 minutes) rather than sending it back to quarantine immediately. This would save clients from having to be quarantined if there was a brief network outage.

8 Possible benefits

Using the method described above would help by reducing the risk of users connecting worm infected machines via DHCP to an organization’s network.

From the user's point of view, they would connect "as normal" to their organization's network. No changes are needed on the client machine.

Users would see some delay while the checks are processed in the quarantine stage and users would not be able to connect to their organization’s network until their machine was checked OK and given a w-net network address by the w-net DHCP server.

In addition, this quarantine method can be used to identify machines that need to have updates applied before granting an IP address in the w-net. This can be done

by identifying some client software versions using vulnerability port scanning techniques (such as using Nessus [3]).

Since this is done at DHCP connection time, there would be frequent quarantine testing. This would raise users' general awareness of the importance of security on their client machines.

A major bonus is that the honeypot software (running on the quarantine DHCP server) could be configured to listen to both the quarantine network and the w-net. A common behaviour of worms is a bias to attempt to infect machines on the local /8 or /16 subnets (e.g. same or similar to the infected machine's local subnet) [11,12].

By configuring the honeypot sensor to listen to both the q-net and w-net, early warning data about anomalous network traffic in the w-net could also be found.

9 Possible problems

This quarantine method checks DHCP clients for problems.

It is possible for a machine with a static IP address to be removed and re-added to the network and to introduce a worm.

This approach would raise user awareness of the importance of the security of most DHCP machines. Thus encouraging that DHCP client machines are kept up-to-date in a regular fashion.

For DCHP client machines that stay net connected for very long periods of time, this method may not be appropriate unless it was acceptable to revoke a w-net IP and force a client machine into quarantine every so often.

It should be noted that this quarantine method does not prevent malicious users from attempting to "steal" a static IP address from w-net and defining that for their client machine. This is generally counter-productive because it skips all the management of IP addresses handled by both the existing DHCP server and whatever process the organization uses for managing static IP addresses. In addition, if the honeypot sensor is configured to use all the *unused* IP addresses available then it will be more difficult for a malicious user to find a usable IP address to steal.

Another issue that is not yet addressed is the problem of client machines attempting to make connections with ports other than port 80/tcp (mentioned above to show quarantine status). For example, how should the quarantine DHCP server deal with SMTP connections from a new client? It is possible that a mail transfer agent (MTA) program on the client has spooled email that it wants to deliver. Since we have a

default route via the quarantine DHCP server we need to refuse a connection for SMTP so that the email stays spooled until a connection into the w-net is established. At the same time, if we want to use honeypot techniques on the quarantine DHCP server it could be useful to “pretend to allow” a connection in order to try to identify worm attack behaviour.

Initially, I think the right approach is for the quarantine DHCP server to refuse such connections but this may make it more difficult to identify worm attack behaviour. This could be a good argument for running the honeypot software on a separate dedicated machine. This issue would need further investigation.

If multiple DHCP clients simultaneously joined the network they would all be placed in the quarantine subnet. However, if one was infected with a worm, it could potentially infect the other quarantined DHCP clients (if vulnerable). In practice however, since all DHCP clients are now being checked at connection time, most clients are likely to be up-to-date with security fixes. In addition, because of the use of the quarantine subnet any worm storm is confined to that quarantine subnet. The quarantine DHCP monitoring script would alert network admins of the worm activity.

Another issue to consider is the type of networking being used. If a modern switched ethernet is in use then the impact of a worm infected machine being active in the quarantine network would be less than if it were a shared media type of network (eg CSMA/CD carrier sense multiple access/collision detect in old bus topology ethernet).

10 Implementation notes

Building a quarantine DHCP system described here would require the addition of a second DHCP server (alongside an existing DHCP server).

For simplicity, it would be useful to have two network interface cards on the quarantine DHCP server. Both would be connected to the same physical network. One would be for the quarantine network and the second would be for the normal w-net.

The quarantine DHCP server would communicate with the w-net DHCP server across the w-net connection. There would be no forwarding of IP packets between the quarantine network and the w-net.

Thus all network traffic between the quarantine DHCP server and quarantine clients would be over a single network interface card.

Scripts needed would include something to exchange MAC addresses between the quarantine and w-net DHCP servers (and update the DHCP configuration files).

There would need to be a script to co-ordinate the quarantine acceptance testing such as port scanning and checking for anomalies in the honeypot data.

The “MAC address management script” that runs on the quarantine DHCP server could work like this: monitor the DHCP state file and identify any newly given out IP address/MAC pair. If a new pair are identified then start the acceptance testing (see below). If the test result is “pass OK” then update the DHCP server configuration file and mark the MAC address as “do not serve”. Check the expiry time of the lease has expired then communicate this MAC address to the w-net DHCP server. If the test result is “fail” keep the MAC address with the quarantine DHCP server and update a web page (or other method to get data back to user) that test failed with explanation and recommended action (call helpdesk).

We are assuming that with a very short DHCP lease time (eg between 30 and 120 seconds) this will mean that when a lease is not renewed (and MAC address passed to the w-net DHCP server) then the user does not have to wait too long for the w-net DHCP IP address.

If the q-net DHCP lease expires before the acceptance testing is done it will just get renewed for another short lifetime lease.

11 Acceptance testing

While in the quarantine network, DHCP client machines are tested with two goals, to check for:

1. worm infections
2. other problems (such as non compliance with organization's Security Policy)

This testing is done automatically and initiated via the "MAC address management script" (see Implementation notes, above).

The nature and type of testing depends on the organization's Security Policy which should define what is and is not acceptable. The contents of the Security Policy will vary from organization to organization.

For example, an organization has decided that, because of the problems with vulnerabilities on various versions of the IIS webserver, it is not acceptable for a DHCP client to run an IIS server on port 80. Therefore, one of the tests done will be to test if IIS is running on port 80 on the new DHCP client machine.

Another example could be the use of the Secure Shell server (sshd daemon). It may have been decided that use of old and vulnerable versions of SSH should be discouraged.

In addition to making specific tests (using Nessus command-line [3] or other tools), the quarantine DHCP server would run honeypot software that will just “listen” passively for any attempts to infect it on various ports and protocols. An example is the IBM BillyGoat sensor [2] which can be configured to recognize different worm attack signatures.

As a result of all this active (eg vulnerability scan with nessus) and passive (eg honeypot) testing on the quarantine DHCP server there could be several possible outcomes:

1. Infected machine.
   Action: Don't allow it to connect to w-net. Alert network admiistrators. Possibly put the MAC address on a “MAC-address-blacklist” to distribute to other DHCP servers.

2. Vulnerable machine. It has some configuration or software version problem.
   Action: warn the user to fix the problem. Start a “countdown” for that MAC address so that the user has (say) 3 more connection attempts otherwise they won't be connected. The user must fix the vulnerability issue if they want to re-connect in future.

3. OK machine.
   Action: mark MAC address as “do not serve” on the q-net DHCP server and pass MAC address to w-net DHCP server.

12 Future possibilities

In this paper I have outlined a method using two DHCP servers. It could easily be implemented by modifying the behaviour of a normal DHCP server to only offer IP address leases to known MAC addresses and adding the quarantine DHCP server which would be used to do the acceptance testing.

The quarantine DHCP server can be implemented with existing technology at low cost.

One interesting possibility is that of introducing a third separate subnet (also sharing the same physical wiring) for the purpose of applying software fixes or configuration changes. We could call this “hospital net” (h-net) where machines are placed to get fixed up. This is separated from q-net because we do not want a vulnerable machine

to wait in the q-net where it could get infected. Also, we don't really want vulnerable machines on the w-net. The h-net would be another unrouted private subnet [5] and would probably require anothet DHCP server.

I believe that a specially written DHCP server could combine the functions of the quarantine and w-net DHCP server on one machine but it would take some software development to achieve that.

13 Conclusion

Some answers to the questions in “5 Trusting DHCP clients” (above):

1. No, without checking, we cannot be sure a DHCP client is or is not infected.

2. Some DHCP clients may not be bona fide. Without checking how could we possibly know?

3. An infected DHCP client would present a great risk which could be mitigated if the client was located in a quarantine subnet.

4. No, we should probably not be letting unchecked DHCP clients access the organization's network.

5. Yes, we should run some checks on DHCP clients.

6. An acceptable duration for quarantine checking? Between 1 and 5 minutes. The less the better.

By combining existing systems of DHCP, honeypot, and vulnerability scanning it is possible to build a system that guards against allowing unchecked machines from joining a network.

This could provide a cost effective way to reduce the impact of worm attacks by refusing connections from worm infected machines. This method also provides a means for DHCP client machines that need to have software updates applied to be identified.

By adding a honeypot sensor, early warning data about anomalous network traffic can be identified and logged both for quarantine and normal working networks.

14 Acknowledgements

I would like to thank the following for supporting me to do this work: my manager Carole Drinkwater, colleagues: Ken Houghton, James Riordan, Diego Zamboni. Fabrice Kah, Florence Adam.

15 References:

[1] Estimated costs of damage done by computer worms
URL: http://www.shift.com/print/9.1/94/1.html

[2] Ricadela, Aaron. “IBM Squashes Worms”. August 2003.
URL: http://www.crn.com/components/Nl/direct/article.asp?ArticleID=44229

[3] Deraison, Renaud. “Nessus port scanner”. 1998-2003.
URL: http://www.nessus.org/

[4] Staniford, Stuart. “The Worm FAQ”. 2003.
URL: http://www.networm.org/faq/

[5] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G.J., Lear, E.
RFC 1918 "Address Allocation for Private Internets". February 1996.
URL: http://www.faqs.org/rfcs/rfc1918.html

[6] Nachenberg, Carey. “Computer Parasitology”
URL:
http://enterprisesecurity.symantec.com/PDF/computerparasitology.pdf?EID=0

[7] Droms, R. RFC 2131 “Dynamic Host Configuration Protocol”. March 1997
URL: http://www.faqs.org/rfcs/rfc2131.html

[8] Vixie, P., Thomson, S., Rekhter, Y., Bound, J.,
RFC 2136 "Dynamic Updates in the Domain Name System (DNS UPDATE)"
April 1997
URL: http://www.faqs.org/rfcs/rfc2136.html

[9] Sieberg, D. and Bash, D. "Computer worm grounds flights, blocks ATMs",
CNN. 26 January 2003
URL: http://www.cnn.com/2003/TECH/internet/01/25/internet.attack

[10] “Computer Worm” definition
URL: http://en2.wikipedia.org/wiki/Computer_worm

[11] Weaver, N., Paxson, V., Staniford, S., and Cunningham, R.,
"A Taxonomy of Computer Worms",
Proc. ACM CCS Workshop on Rapid Malcode, October 2003.
URL: http://www.cs.berkeley.edu/~nweaver/papers/taxonomy.pdf

[12] Moore, D., Shannon, C., Brown, J.
"Code-Red: a case study on the spread and victims of an Internet worm",
Proceedings of the Second the ACM Internet Measurement Workshop, 2002.
URL: http://www.caida.org/outreach/papers/2002/codered/codered.pdf

[13] Staniford, S., Grim, G., Jonkman, R.
"Flash Worms: Thirty Seconds to Infect the Internet"
URL: http://www.silicondefense.com/flash/

16 Terms and abbreviations

DHCP: Dynamic Host Configuration Protocol [7]. This protocol defines a process for client machines to be given IP address (and other information e.g. default route, and DNS server IP address) by a DHCP server. This makes the TCP/IP configuration of the DHCP client machine "automatic" and saves the client user time when joining a client machine to the network.

Dynamic DNS: This protocol [8] defines a process that enables client machines to request a Domain name Service (DNS) update. Dynamic DNS is typically used for DHCP clients (which may have different IP addresses over time) to request and define a "static" DNS hostname rather than a "machine generated" hostname (e.g. dhcp-10-0-0-56.domain.tld).

honeypot: a computer system configured to simply "listen" for inbound connections and report them. Honeypots may be thought of as decoy systems which are just used for identifying unexpected network activity and attacks. Some honeypots are configured with minimal "visibility" by using IP addresses not registered in DNS and not making their presence known in the network.

MAC address: Media Access Control address, a hardware address that uniquely identifies each node of a network. There is normally an association between the MAC address and the IP address of a node. DHCP servers keep track of which IP address they have granted to which MAC address.

Port scanning: A process of attempting to connect to ports on a remote machine to determine what ports are active and to collect information about what servers are running on those ports.

q-net: A term used in this paper to refer to the quarantine network.

w-net: A term used in this paper to refer to the “normal working network” of an organization. This is used to distinguish between the quarantine (q-net) network and the work network (w-net).