Global Information Assurance Certification Paper

The Hping2 Idle Host Scan
Erik J. Kamerling

Introduction

I recently read an online Scientific American article about the hunt for new planets. It outlined the difficulty astronomers faced in determining if a star has any orbiting planets. Those difficulties were caused by great distances, technological limitations, and the fact that planets emit very little light. Inevitably, scientists realized that they needed a new method for detecting the presence of planets outside of our own solar system.

Doppler Planet Detection is the method that was devised. Using this method, astronomers analyze variations in the light being emitted from a star. If the light appears to shift from the blue to the red sections of the spectrum, there is a possibility that a planetary body is orbiting the star. A planet will exert a certain amount of gravitational pull on a star, which causes the star, and the light which its emits, to wobble to and fro. This is quite an ingenious detection method, and has proven invaluable in modern day astronomy.

The thing that truly interested me about this article was the fact that scientists can determine a planet's characteristics with great certainty, without ever actually seeing the planet itself. An orbiting body's mass and the distance from its sun, can be determined by analyzing the effect it has on the star, which it orbits. I found this type of analysis quite extraordinary, and was surprised to find out that there was a theoretically equivalent analysis method in the hacking community. There is a method of determining the characteristics of a target computer, by analyzing the behavior of a third party, and the effect that the target has on it. It is called Idle Host Scanning. Although much easier to understand than Doppler Planet Detection, at it's core, this method of intelligence gathering is very similar indeed.

Overview of Idle Host Scanning

Let me begin by giving a brief explanation of Idle Host Scanning. It is a clever combination of different information gathering techniques, which are in wide use on the Internet today. It combines IP spoofing and network, or host scanning, into an effective way to perform an anonymous probe for services. It is similar in many ways to the scientific method outlined above, and is best described by building a direct comparison between astronomers, and those who would use an Idle Host Scan. The astronomers object of desire is an invisible planet, and the attackers, an Internet connected host. The scientist will perform a direct analysis of a star's light properties in order to glean information on it's orbiting planet. A hacker will analyze the behavior of an idle host, in an attempt to gain information on their target. Both methods, because of the indirect nature of their information gathering techniques, go hand in hand.

As with any type of scan, the attacking party has a desire to gather information on the characteristics of a network, or Internet connected computer. The information could be

what services that host offers, what ports those services run on, or even the operating system of the target. Typically, such scans are a precursor to either heavier probing, or outright attacks. The appeal of the Idle Host Scan to the attacking party, is the fact that a scan of this type, can be accomplished anonymously.

This leads us to the truly insidious nature of this scanning method. With the relative anonymity that can be maintained while using this method, through the use of unsuspecting third party computers, this type of network probe is very difficult to trace. Tracking down offending parties in an Idle Host Scan can be a tall order for the network security practitioner. Not only would one need to contact innocent third party host owners, but logs would also need to be compared, in the hopes that the true origin of the scan could be revealed. You could only hope that the originating address was not a compromised system, which the attacker had used as a launch pad. This type of scan poses a definite threat to networks and Internet connected computers, and offers a promisingly anonymous method of information gathering for attackers.

Breakdown of the Scan

A small sample network was used to gather information for this paper. The following is information on the hosts involved, and the sequence by which the sample attack takes place. This information is the breakdown of the idle host scan concept and does not include specific details. The section of this document dealing with Hping analyzes this type of scan on a much more detailed level.

	Attacking Host	**Idle Host**	**Target Host**
Operating System	Red Hat Linux 6.2	Windows NT 4 SP5	Windows 98
IP Address	192.168.0.4	192.168.0.5	192.168.0.6

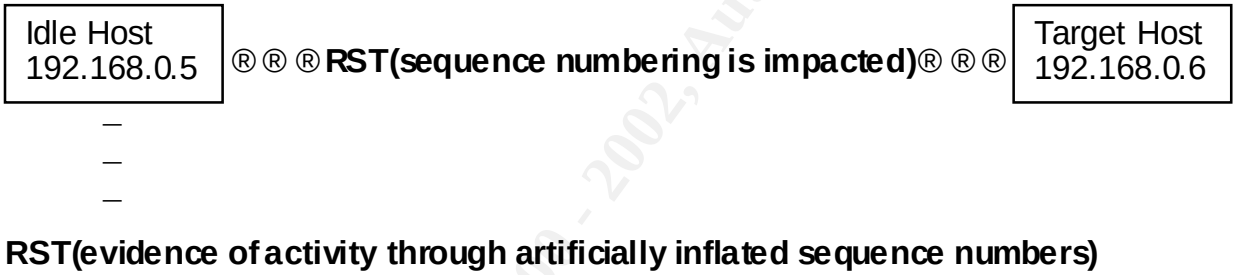
Step 1. The first step is for the attacking host to send the idle host some packets with no flags set, to port 0. Typically, this type of packet should not be making its way around, and we should expect a RST packet from the idle host.

Step 2. The second phase in this attack occurs when the idle host acknowledges the strange packets with RST packets bound for the attacker.

Step 3. In the third phase of the attack, the attacking host sends spoofed SYN packets to the target, with the originating address specifically crafted as the idle host.

Step 4. Once the target host responds to the spoofed packets, and sends that response to the idle host, the fourth phase is complete.

Step 5. The fifth step in this scan is the most complex. With the session still running from the attacker to the idle host, the attacker hopes to see a reflection in the sequence numbers of the RST packets that the idle host is sending back. This would imply that the idle host is sending RST packets to the target for an unexpected ACK. These RST packets will take up sequence numbers, therefore increasing the sequence number in the RST packets that the idle host is sending to the attacker.



From an attackers point of view, this example scan was highly successful. It is reflected in the fact that the sequence number generation from idle host to attacker was affected. The attacker would know at this point, that the target host was responding to spoofed SYN packets sent to a particular port. This would signify that the target offered a service on that port.

This scanning method depends heavily on one characteristic of the idle host. That characteristic being, a predictable TCP sequence number generation scheme. Idle Host Scanning depends on analysis of activity at a packet level, and uses the IP sequence number as a benchmark. It would be virtually impossible to gauge a host's activity level if it produced truly randomized sequence numbers, regardless of activity level.

In 1989, Steven Bellovin outlined the dangers of predictable sequence number generation, in his paper, "Security Problems in the TCP/IP Protocol Suite". Given the

amount of time that has passed since that date, it is interesting to note that there are still operating systems, which exhibit this behavior. Predictable sequence number attacks are deserving of a book in and unto themselves, so I will not be going into detail in this paper. Yet, I would like to point out that highly successful security compromises, which depended on this characteristic, have happened in the past. The Kevin Mitnick-Tsutomu Shimomura attack stands as the perfect example. Unfortunately, attacks still exist today, which partially depend on predictable sequence numbers. Idle Host Scanning being the perfect example.

Enter Hping

Hping is an application written by Salvatore SanFilippo, an Italian programmer. He is also known as Antirez. Mr. SanFilippo is accredited with the discovery and development of this scanning technique. This discovery was outlined in a post he made to Bugtraq on December 18, 1998. A copy of the whole document can be found at the following URL. http://www.kyuzz.org/antirez/papers/dumbscan.html

As it's name would imply, Hping is a program to perform a type of ping. In Salvatore SanFilippo's own words, "Hping is software to do TCP/IP stack auditing, to uncover firewall policy, to scan TCP ports in a lot of different modes, to transfer files across a firewall and many other things." Although Hping has the ability to use the ICMP protocol as with a conventional ping, this paper will only deal with Hping over the TCP protocol. Hping has the capability to deliver a packet, or packets, over the most commonly used protocols today.

Hping is GPL software that will run on GNU/Linux, FreebSD, and OpenBSD, as well as other operating systems. The main URL for Hping is http://www.kyuzz.org/antirez/hping. The source code for this application is included in the package.

This paper is not an analysis of Hping itself, but instead looks at how an Idle Host Scan can be accomplished with Hping. It is not only an Idle Host Scanning tool, but also a very savvy network administration program. For those interested in the wide variety of functions that Hping supports, you are encouraged to read the documentation that comes with the downloadable source code.

Although my experience with Hping is limited to the following test scan. I am very convinced of its versatility as a network administrator's tool. A few examples would be, testing firewall rules, vulnerability assessment, and other network security related tasks. Hping has evidently has made its way into the FreeBSD ports collection, which would indicate that quite a few people are using it for administrative purposes. Unfortunately, the state of affairs in Information Security today, is that the most powerful freely available tools are used for legitimate, as well as nefarious purposes. Needless to say, Hping makes a powerful attack program as well.

The Hping Idle Host Scan

The following information is a direct copy of an Hping Idle Host Scan, performed on a test network. In order to provide better insight into the data that follows, I have included definitions of the switches used for Hping in this particular type of scan. A full text of Hping options is available with the Hping documentation.

Another thing to note is that this type of scan is most easily accomplished within X-Windows, where two occurrences of Hping can be run on the same desktop. Two occurrences of Hping must be run simultaneously. This is due to the fact that within the same time frame, the attacker must be using Hping to monitor one host, while sending spoofed packets to another.

This information is copied directly from the Hping help file. Any information from the documentation, or any direct output from Hping itself is intentionally left in its original form.

```
usage: hping host [options]

-r  --rel        relativize id field          (to estimate host traffic)
-W  --winid      use win* id byte ordering
-a  --spoof      spoof source address
-S  --syn        set SYN flag
-p  --destport   [+][+]<port> destination port(default 0) ctrl+z inc/dec
```

The following data is from the perspective of an attacking party in an Idle Host Scan. Not only will it show the offender's point of view in this attack, but also what this attack looks like on the target machine. It should be noted that the target is protected by a personal host based firewall.

Below is the first step in the attack, the initiation of an Hping session with the idle host.

```
[root@test sbin]# ./hping 192.168.0.5 -r
eth0 default routing interface selected (according to /proc)
HPING 192.168.0.5 (eth0 192.168.0.5): NO FLAGS are set, 40 headers + 0 data
bytes
46 bytes from 192.168.0.5: flags=RA seq=0 ttl=128 id=6416 win=0 rtt=0.3 ms
46 bytes from 192.168.0.5: flags=RA seq=1 ttl=128 id=+256 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=2 ttl=128 id=+256 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=3 ttl=128 id=+256 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=4 ttl=128 id=+256 win=0 rtt=0.3 ms

--- 192.168.0.5 hping statistic ---
5 packets tramitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.3 ms
```

As can be seen, the output from the simplest of Hping commands is very similar to conventional ICMP output. The –r switch is used in this initial connection, which is telling Hping to only show the attacker what the difference is in the id field. So, as opposed to seeing the whole, sometimes hard to manage id value, Hping only shows you the difference in that field.

There are two important points to note here. The first being, that this is the start of the process of determining if a host is idle, or active. By analyzing the id field the attacker can tell if the host is sending packets to and from the Internet. If the id field values change frequently, and not in a predictable manner, the attacker will assume that the host in question is experiencing some sort of packet exchange. If the host's id generation is very active, the host would be inappropriate for participation in this type of scan. The output above, however, shows that we have a very inactive host with which we are dealing. This makes it a prime candidate as the third party in this scan. Please keep in mind that any Internet connected device with a TCP/IP stack can be used in this manner. This includes workstations, printers, and routers, to name a few.

The second point, is that the value in the id field increases by a standard +256 for every packet sent. According to the Hping documentation, this is an indicator of a Windows host. Hping has proven itself already as a valuable information gathering tool, since the attacker has already determined that the idle host is using a Microsoft operating system.

The output below, is the same exact Hping session, but with the –W switch used in order to compensate for the +256 Windows behavior. Please note the difference in the following output, which is caused by the –W option.

```
[root@test sbin]# ./hping 192.168.0.5 -r -W
eth0 default routing interface selected (according to /proc)
HPING 192.168.0.5 (eth0 192.168.0.5): NO FLAGS are set, 40 headers + 0 data
bytes
46 bytes from 192.168.0.5: flags=RA seq=0 ttl=128 id=4126 win=0 rtt=0.4 ms
46 bytes from 192.168.0.5: flags=RA seq=1 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=2 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=3 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=4 ttl=128 id=+1 win=0 rtt=0.3 ms
46 bytes from 192.168.0.5: flags=RA seq=5 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=6 ttl=128 id=+1 win=0 rtt=0.2 ms

--- 192.168.0.5 hping statistic ---
7 packets tramitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.3/0.4 ms
```

The above data is easier to handle than the previous data set, because the id field has now been normalized with the compensation for the MS Windows +256 behavior. By watching the id field, the attacker can determine with some confidence, if a host is sending packets. Accordingly, the attacking party would be able to draw the conclusion from the above data, that this host is idle.

Assuming that the attacker has already obtained the IP address of their intended target, they can then proceed with the Idle Host Scan.

What follows is the output of an attacker sending spoofed SYN packets to a target host. The spoofed originating address is that of the idle Windows NT host as defined earlier. The target is a Windows 98 host that offers no services but HTTP on port 80. Keep in

mind that the attacker does not know this, and will send packets to the FTP service on port 21 first.

```
[root@test sbin]# ./hping -a 192.168.0.5 -S -p 21 192.168.0.6
eth0 default routing interface selected (according to /proc)
HPING 192.168.0.6 (eth0 192.168.0.6): S set, 40 headers + 0 data bytes

--- 192.168.0.6 hping statistic ---
8 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

You should notice that there was 100% packet loss in this, the third stage in the attack. The reason for this is obvious. When the scanning party sends packets spoofed with the originating address as that of the Windows NT computer, the response from the Windows 98 machine will be to the idle host. This would cause a complete packet loss on the part of the attacker. Yet, by maintaining their initial monitoring session with the Windows NT host, they have not lost the ability to see the impact of their spoofed SYN packets.

Herein lies the turning point in an idle host scan. If the attacker has an ongoing Hping session with the idle host, the attacker should be able to determine if the target has sent packets as a response, to the idle host, by monitoring the values in the id field of the Hping output. Below, is the output from the idle host session with the NT workstation. If there were a marked change in id values, the attacker would assume that FTP is available.

```
[root@test sbin]# ./hping 192.168.0.5 -r -W
eth0 default routing interface selected (according to /proc)
HPING 192.168.0.5 (eth0 192.168.0.5): NO FLAGS are set, 40 headers + 0 data
bytes
46 bytes from 192.168.0.5: flags=RA seq=0 ttl=128 id=4141 win=0 rtt=0.4 ms
46 bytes from 192.168.0.5: flags=RA seq=1 ttl=128 id=+1 win=0 rtt=0.3 ms
46 bytes from 192.168.0.5: flags=RA seq=2 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=3 ttl=128 id=+1 win=0 rtt=0.3 ms
46 bytes from 192.168.0.5: flags=RA seq=4 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=5 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=6 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=7 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=8 ttl=128 id=+1 win=0 rtt=0.2 ms

--- 192.168.0.5 hping statistic ---
9 packets tramitted, 9 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.3/0.4 ms
```

Please note that the id value did not change in this hping session with the idle host. Considering that the values stayed uniform for the given time frame that the spoofed packets were sent, one would assume that the Windows 98 host offers no FTP services. If the target host did offer FTP on port 21, the transaction would be as follows.

- The target would receive the spoofed SYN packets.

- It would then respond to the packets with a SYN/ACK bound for the NT idle host
- The Windows NT host would then receive the SYN/ACK packets, and respond appropriately with a RST, because it did not make a request for FTP.

If that were the case, the RST packet generation would change the values in the idle host's id numbering scheme. Since this was not the case according to the previous data set, the attacking party could assume that the FTP service is not available on the Windows 98 target host.

In order to provide some in-depth information on what an Hping spoofed scan for an unavailable service looks like from the targets point of view, please see the following information. This log data was obtained from ZoneAlarm, a freely available host based firewall product, for Microsoft Windows operating systems. Remember, the only service that the Windows 98 machine offers, is HTTP on port 80. The personal firewall is configured to stop requests for service, to any port other than 80. This should be reflected in the following log output, which shows denied traffic on the target host.

```
ZoneAlarm Basic Logging Client v2.1.44
Windows 98-4.10.2222- A -SP
type,date,time,source,destination,transport
FWIN,2001/01/30,14:12:10 -5:00 GMT,192.168.0.5:3014,192.168.0.6:21,TCP
FWIN,2001/01/30,14:12:12 -5:00 GMT,192.168.0.5:3015,192.168.0.6:21,TCP
FWIN,2001/01/30,14:12:12 -5:00 GMT,192.168.0.5:3016,192.168.0.6:21,TCP
FWIN,2001/01/30,14:12:14 -5:00 GMT,192.168.0.5:3017,192.168.0.6:21,TCP
FWIN,2001/01/30,14:12:14 -5:00 GMT,192.168.0.5:3018,192.168.0.6:21,TCP
FWIN,2001/01/30,14:12:16 -5:00 GMT,192.168.0.5:3019,192.168.0.6:21,TCP
FWIN,2001/01/30,14:12:16 -5:00 GMT,192.168.0.5:3020,192.168.0.6:21,TCP
FWIN,2001/01/30,14:12:18 -5:00 GMT,192.168.0.5:3021,192.168.0.6:21,TCP
```

This information explains why there was no marked id value change on the part of the Windows NT workstation. The firewall on this computer denied inbound traffic to the FTP service. Of interest is that the originating address for the FTP request is showing up in the above log, as the address of the idle host, and not the attacking party. By blindly trusting logs on this machine, the administrator of this computer could potentially accuse the wrong person of scanning them for FTP.

The following output will reflect an Hping Idle Host Scan, which is successful in the sense that the target computer shows that it has a service available by responding to the idle host. The attacker could, with some confidence, assume that the target computer is offering a service on port 80, if they had obtained this output.

The attacker sends spoofed SYN packets to port 80 of the intended target.

```
[root@test sbin]# ./hping -a 192.168.0.5 -S -p 80 192.168.0.6
eth0 default routing interface selected (according to /proc)
HPING 192.168.0.6 (eth0 192.168.0.6): S set, 40 headers + 0 data bytes
```

```
--- 192.168.0.6 hping statistic ---
7 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Meanwhile, the attacker has been monitoring the idle host with another Hping session.

```
[root@test sbin]# ./hping 192.168.0.5 -r -W
eth0 default routing interface selected (according to /proc)
HPING 192.168.0.5 (eth0 192.168.0.5): NO FLAGS are set, 40 headers + 0 data
bytes
46 bytes from 192.168.0.5: flags=RA seq=0 ttl=128 id=4170 win=0 rtt=0.3 ms
46 bytes from 192.168.0.5: flags=RA seq=1 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=2 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=3 ttl=128 id=+1 win=0 rtt=0.3 ms
46 bytes from 192.168.0.5: flags=RA seq=4 ttl=128 id=+2 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=5 ttl=128 id=+2 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=6 ttl=128 id=+2 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=7 ttl=128 id=+2 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=8 ttl=128 id=+2 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=9 ttl=128 id=+2 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=10 ttl=128 id=+2 win=0 rtt=0.3 ms
46 bytes from 192.168.0.5: flags=RA seq=11 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=12 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=13 ttl=128 id=+1 win=0 rtt=0.2 ms
46 bytes from 192.168.0.5: flags=RA seq=14 ttl=128 id=+1 win=0 rtt=0.2 ms

--- 192.168.0.5 hping statistic ---
15 packets tramitted, 15 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.3 ms
```

The most important thing to note about the above data, is the change in the id field of the idle host. Armed with the above information, the attacker could assume that the seven spoofed SYN packets that were sent to the target, were acknowledged with seven ACK packets. The ACK packets were sent to the idle host, which in turn sent seven RST packets. Those seven RST packets occupied id numbers, which were reflected in the monitoring session output. With seven spoofed packets sent, and the impact on the id values of the idle host being approximately seven, the data is strongly indicative of a service being offered on the target host. The most interesting fact, is that the attacker originating from 192.168.0.4 has just successfully scanned the target at 192.168.0.6, anonymously spoofed as 192.168.0.5.

As with any method or tool for scanning hosts or networks, this could be used as an inverse network mapping tool. Obviously the lack of reaction in the idle hosts id sequence would indicate the absence of a service. The service could be firewalled, not available, or configured with any number of third party tools to limit connectivity. This data is valuable to an attacker, in that they can profile whole networks by cataloging what is not available, be it either hosts, or services.

Hping advancements, and other tools

Currently the most recent version of Hping, Hping2-beta54, has reached the end of it's life. Salvatore SanFilippo, and other project volunteers, are rewriting Hping from the ground up. According to his website, Hping3 will be a far superior tool to the current version. There will be installation improvements, more readable output, and Hping3 will be scriptable. The current status of the project can be followed at http://www.kyuzz.org/antirez/hping3.html.

Hping is not the only tool that can perform an Idle Host Scan. Idlescan is a tool, which has elaborated on the concept behind Hping Idle Host Scanning. It has simplified the concept of using idle hosts on the Internet, to scan for services. It does not come with the large variety of options that Hping does, but for those attackers who are only interested in performing this type of scan, it makes a tempting pre-canned scanning tool. This too is a freely available program, and the URL where it can be obtained is in the reference section of this document.

Common Defenses

Since Idle Host Scanning is a combination of information gathering techniques and hacking methods, defending networks from such scans requires a multi-tiered approach. Defenses for such an attack should be broken into two different categories. The first category consists of host level defensive mechanisms. The second is network level defense. Tightening security at a host level will ensure that a machine under your administrative reach can not be used as a third party host in this type of scan. At a network level, you are ensuring that Idle Host Scanning doesn't originate from your network. Also, signatures of this type of scan can be alerted upon, and dealt with appropriately.

Defense Section 1 – Host Level

Predictable TCP sequence number generation on the part of the idle host is one of the primary variables in this type of attack. Therefore, randomization of these numbers should be an appropriate defense. According to Steven Bellovin's previously referenced paper, using cryptography in initial sequence number generation is an effective way to ensure that sequence numbers are truly random, and not predictable. As an example, this has already been implemented in OpenBSD, which uses an algorithm to alter its initial sequence numbers as it communicates. Due to this security conscious enhancement, a computer running OpenBSD makes a very unattractive idle host.

OpenBSD only stands as an example of the many operating systems today, which either employ cryptographically generated sequence numbers, or can be patched to do so. Along with the above requirements, proper host based security, log management, and host based firewalls are necessary in order to ensure that one's hosts are not used in this type of attack. This leads us to the next step in defending systems against this type of scan, the implementation of proper network based security.

Defense Section 2 – Network Level

Since IP spoofing is an integral part of an Idle Host Scan, following the countermeasures to IP spoofing according to CERT advisory CA-1995-01, is highly advisable. Where it is stated,

"The best method of preventing the IP spoofing problem is to install a filtering router that restricts the input to your external interface (known as an input filter) by not allowing a packet through if it has a source address from your internal network. In addition, you should filter outgoing packets that have a source address different from your internal network in order to prevent a source IP spoofing attack originating from your site."

The information above clearly provides guidelines for preventing an outgoing Idle Scan from your network, yet seems to fall short of providing an explanation of how to defeat incoming spoofed packets for this type of scan. It is probable that spoofed packets in a scan of this type would originate from the Internet at large, and not from your own network. Implementation of input filters does however; effectively combat many other types of IP spoofing attacks. Even though it may not harden your systems against an Idle Host Scan, it is good practice to employ this countermeasure.

This leads us to the next section of the advisory, which approaches a solution, yet still does not completely define it.

"We suggest that you use TCP wrappers to allow access from only a select few machines. Although this is not a complete solution, it does reduce your susceptibility to attack. Alternatively, change the configuration of your Internet gateway so that rlogin and rsh from the Internet to hosts in your domain are blocked. If that is not possible, disable the rlogin and rsh services on all of your hosts."

The use of TCP Wrapper, and other third party tools, can be a very comprehensive solution to defending against many types of scans and attacks. I believe that a combination of practices will provide an acceptable solution for network based defense in this scenario. The number one defense is diligence on the part of the administrator. This would include reading logs, and responding to scans and attacks with the appropriate follow up procedures. In addition, ensuring that those systems are up to date with patches and hotfixes, is imperative.

The implementation of Intrusion Detection Systems at a network level is important as well. There are quite a few commercial, and freely available Intrusion Detection Systems in wide use today. Their capabilities range from passive traffic analysis, to all out active defense. It will be left as an exercise for the reader, to decide what IDS would provide an appropriate solution for you environment. A great starting point for those interested is, http://www.sans.org/newlook/resources/IDFAQ/ID_FAQ.htm.

Since Idle Host Scanning is a combination of different attack methods, defending systems against such a scan must be approached from a multi layered viewpoint. In an attempt to "harden" against Idle Host Scanning, system and network administrators must be dynamic in their choice and implementation of defensive measures.

Conclusion

Some would argue that network scanning does little to no harm. Indeed from a scientific viewpoint, astronomers use comparable information gathering techniques in order to retrieve data on very distant planets. Hackers are simply utilizing alternative methods and procedures to gather information on computers, which interest them. Of course, the primary difference between both methods, is that a planet being studied from Earth, is not under threat of being attacked. As would be the case, if someone using the above procedure were to scan your network.

In today's ever changing environment of interconnected networks, it is increasingly difficult to maintain the confidentiality, availability, and integrity of our computing resources. This is especially true in a world where hackers are constantly devising new ways to gather information on, probe, and attack our networks. The Idle Host Scan using Hping, as it was defined in this document, is not inherently an active attack. It is an information gathering technique that is probably being used as an anonymous way to scan our networks, and compile intelligence for an impending assault.

References

[1] Marcy, Jeffrey. Butler, Paul. "Giant Planets Orbiting Faraway Stars." Scientific American, Volume 9, Number 1, 1998. March 1998.
URL: http://www.sciam.com/1998/0398cosmos/0398marcy.html
(February 12, 2001)

[2] CERT Advisory CA-1995-01. "IP Spoofing Attacks and Hijacked Terminal Connections." September 23, 1997.
URL: http://www.cert.org/advisories/CA-1995-01.html (January 21, 2001)

[3] CERT Advisory CA-1996-21. "TCP SYN Flooding and IP Spoofing Attacks." November 29, 2000.
URL: http://www.cert.org/advisories/CA-1996-21.html (January 21, 2001)

[4] Securiteam. "A New Stealth Port Scanning Method." December 23, 1998.
URL:
http://www.securiteam.com/securitynews/A_new_stealth_port_scanning_method.html (February 12, 2001)

[5] Bellovin, Steven M., "Security Problems in the TCP/IP Protocol Suite." Computer Communications Review, 2:19, pp. 32-48, April 1989.
URL: http://www.research.att.com/~smb/papers/ipext.ps (February 19, 2001)

[6] de Raadt, Theo. Hallqvist, Niklas. Grabowski, Artur. Keromytis, Angelos D.. Provos, Niels. "Cryptography in OpenBSD: An Overview." Usenix 1999. URL: http://www.openbsd.org/papers/crypt-paper.ps (February 19, 2001)

[7] SanFilippo, Salvatore. "Original Bugtaq Posting." December 18, 1998. URL: http://www.kyuzz.org/antirez/papers/dumbscan.html (February 25, 2001)

[8] SanFilippo, Salvatore. "HPING2-HOWTO.txt." August 10, 1999.

[9] SANS Institute. "Intrusion Detection FAQ." Version 1.42. URL: http://www.sans.org/newlook/resources/IDFAQ/ID_FAQ.htm (February 25, 2001)

Links to Downloadable Source Code

Hping2-Beta54 Source Code - http://www.kyuzz.org/antirez/hping-src/

Idlescan-v0.1 Source Code - http://www.hackers-pt.org/ptstuff/

TCP Wrapper Source Code - ftp://ftp.porcupine.org/pub/security/index.html

Last Updated: May 25th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event