ISSN: 2277 – 9043

International Journal of Advanced Research in Computer Science and Electronics Engineering
Volume 1, Issue 3, May 2012

AN OVERVIEW & COMPARISION BETWEEN INACTIVE FINGER STAMPING V/S ACTIVIATION FINGER STAMPING TECHNIQUE

Nitin Tiwari[1], Rajdeep Solanki[2], Gajaraj Pandya[3]

[1]Institute of Computer Science Vikram University,Ujjain
[2]Institute of Computer Science Vikram University,Ujjain
[3] Institute of Computer Science Vikram University,Ujjain
[1]tiwari.tiit@gmail.com
[2] rajdeepslnk@yahoo.com

[3] gajaraj_pandya@yahoo.com

Abstract— Now a days there are increasing the use of networking in large scale, the virus, virus and safety related also major significant issue. The O/S of a remote host on the Internet is feasible using tools by ability to remotely know, with high accuracy and wide classification of method. Developing this methods are perhaps not yet fully understood; however, it is look as enough of a threat that creating to use rules are presently being, analyze, deploy, prevent or spoof OS Finger stamping. For the usage of Finger stamping is only known once the data is grouped and analyze by the call of *analysis*, In so doing, the authors *use* real qualitative *data* to help distinct, commonly *known* as statistics. *Only* avail unfailing qualitative *data analysis* tool is promoted a*fter* all the *data* have been coded, the codes are *grouped* by similarity. Under unique implementation counter measures by technique which goes back remote system O/S Finger stamping, The soaking the vendor exclusive weakness avail unfailing in the O/S and the different techniques by older Finger stamping techniques were busy consisted of fooling and hiding from the hackers, like banner hiding, change of default setting in common services and resist etc. This lead to accuracy of prediction and easy agreement of the system to Finger stamping tools the attackers easy unfailing to launch further attacks once they know the O/S running on the remote system, The hackers begin to use of weakness, default and protective behavior of the network transport protocols used through the remote system to attach itself to Internet and Intranet. By RFC's and the norms of protocols kept in mind the level of safety from common virus, but did not have the concern unique when up Opposed the Finger stamping. So the basic safety from Finger stamping can be access by employing a defensive safety framework. The ease of friction ridge skin means that no two finger or palm prints are ever in reality alike in each detail; even two impressions recorded instant after each other from the same hand. The steps that should be taken are: know safety level, default behavior of protocol are understand, normal technique of Finger stamping analyzed.

Keywords

finger stamping, inactive finger stamping , activation finger stamping

I. INTRODUCTION

INACTIVE FINGER STAMPING

For more studying about the attacker with out danger detection by Passive Finger stamping method, You can possibly known the operating system, services, and programs of a remote host by using nothing more then sniffer traces. Stuffy, using active tools; these instrument execute on the principle that every operating system's IP stack and programs have constant properties and idiosyncrasies. One

can send a series of investigation packets to task systems and test the resist very safely. Many properties that as default TCP window size, supported TCP options, and ICMP bug message features are then compared opposed a database of inform resist until a match is found. Because differ systems perform in distinct ways when they get some packet types, this known can be used to constantly opinion a given operating system. Passive Finger stamping follows the same concept but is implemented differently. Passive Finger stamping is based on sniffer scan of traffic build by the remote system. Against of actively questioning the remote system, it normally catches packets sent by the remote system. Learn, the Honey net catches every packet sent through remote system. Because this is being execute passively, without black hat's information, passive Finger stamping does not growth the risk of the black-hat's search of being linked to a honey pot. Its goal will be to learn the most knowledge without the hacker's being informed of our data collection. Hence, try to opinion the operating system, services, and, sometimes, the program used by the enemy. The more knowledge acquired the best. Passive OS matching has become a new area of detection in both white hat and black hat arenas. For the white hat, it becomes a new process to map their network and monitor traffic for safety. For instance, a new and activity subversive host can be search suddenly, often with more correct. For the black hat, this process provides a nearly un-catch unfailing process to map a network, searching lower hosts. To be insuring, passive matching can be a time demanding process. Even with automated tools more quantity packets to reach to make up a statistically significant reading of the subjects' O/S.

Passive Finger stamping has some advantages over active Finger stamping.

1. **All TCP/IP layers are unfailing for act.**
2. **Search systems with low uptime.**
3. **Search patterns of behavior.**
4. **With the remote user uninformed of what we are learning.**

But passive Fingerstamping is not perfect.

1. **Passive is not 100 percent correct.**
2. **Few programs made their self packets and will not yield the same subscription as the O/S itself would.**

Few of the default parameters rely on can be exchanged.

A-TECHNOLOGY & SUPPORT

A.a. a Technology for TCP-Testing four TCP packet data headers to know the O/S; however other subscription can be consumed. Given below fields is in the header:

- **IP time – to- live: - packet to reach its task, or time to live by number of routing hops allowed. The field also consumed by trace route applications.**

- **Window size: - O/S varies by internal TCP data flow control measure that.**

- **DF:- The IP "Don'ts Divided" bit which some operating systems always set.**

- **TOS:- The IP "Type of Service" field, who's setting visible information about the underlying OS.**

May be unfailing to know the remote O/S by analyzing packet fields; the system is not 100 percent true and works better for few operating systems than for others. No single signature can reliably know the remote operating system. However by seeing at several subscriptions and adding the knowledge, you grow the correctness of opinion the remote host. Number of other data packet properties could be consumed. The simplest way to describe this is through an instance. Following is the sniffer trace of a system consuming a data packet. This system generated a connected soaking Opposed the Honey net, so we want to study more about it. This method learns the knowledge passively; by using short retraction this signature. Based on four criteria, opinion is given below:-
TTL: 45
Window size: 0x7D78, or 32120 in decimal
DF: Don't Divided bit set
TOS: 0x0
Matching this known to a database of subscription, first, see at the IP TTL used by the remote host. The sniffer trace demonstrates that the TTL is set at 45. There is mean that the original TTL was set to 64 and went by 19 hops to access to us. Foundation on this TTL, it seems that this packet went by 19 hops to access to us. Foundation on this TTL, it seems that this packet was sent by a Linux or a Free BSD box; however, more system subscription requires to be joining to the database. TTL can be validation by executing a trace route to the remote host. If you are concerned that the remote host will search your trace route, you can set its time-to-live (default 30 hops) to be one or two hops less than the remote host: m option for a UNIX system, h for Microsoft systems. In this case, doing a trace route to the remote host, starting with a TTL of 18 hops (trace route- m 18). That provides us the path data, including its upstream giver, without fortunately touching the remote host. Be safely with this procedure. Routing paths to and from your utilities may vary, making this method un-predict un-failing. The next step is to compare the TCP window size. The window size to be another strenuous tool: what window size is used and how often the size changes. In the earlier signature, it seems set at 0x7D78. Default window size normally used by Linux. Also, Linux, Free BSD, and Solaris tend to handle the same window size allover a session, as this one did. However, Cisco routers and Microsoft

Windows/ NT window size are continually changing. However, this may also be at least partially features of network setback and executing times rather than an inherent O/S features. However this does make it easier to opinion the few systems, such as SCO or Open BSD, that not use the DF flag. This seems to be more session based than O/S protégé. In other words, it's not so much the O/S that determines the TOS but the protocol used. After further testing, we feel that TOS is also of limited value. We have found that window size is more accurate if measured after the beginning three-way handshake, owing to TCP slow start. Most systems set the DF bit, so this is of limited value. TCP and ICMP for instance manage the TOS field distinct. TOS definitely requires some extra examine. So based on the earlier known about TTL and window size, you can match the outcomes to the database of subscription and with a degree of confidence known the OS- in our case, Linux based on Kernel 2.2.x. This method is not bounded to the four TCP field parameters discussed so far. In other areas too can be scanned, such as beginner series numbers, IP views numbers, and TCP or IP views. For instance, Cisco routers tend to start IP view numbers at 0 replace of randomly assigning them. For TCP choice, the choice selective information sack ok is normally used by Windows and Linux but not by Free BSD or Solaris.

A.a.b Technology of ICMP

There is constant in ICMP RESONANCE request which commonly each O/S has this capacity. This builds ICMP-based programs one of the most normally consumed by black hats. Commonly the ping utility is used to make ICMP Resonance requests. It can be a clear differ among the ping execution with UNIX and UNIX- like O/S and the ping execution with Microsoft- based O/S. This instance will match two ICMP Resonance requests, one from a Microsoft based O/S and one from a Linux/UNIX machine.

ICMP Resonance request datagram size: – based on Microsoft O/S , the ICMP Resonance Request build with ping will be 60 bytes long. With UNIX and UNIX- like O/S, the ICMP Resonance Request build with the ping utility will be 84 bytes long.

ICMP Resonance request data payload content: - Packet in ICMP Resonance Request sent with the ping utility on a Microsoft- based O/S will be creation of the alphabet, whereas UNIX and UNIX like operating systems' ping will use numbers and symbols.

ICMP Resonance request timestamp: With the ping output, a time computation of the round- trip time (RTT), or how long it took the datagram to travel from the beginning host to the goal host and to return. With ping on UNIX and UNIX- like O/S , the first 8 bytes of the data payload are a timestamp helping us to compute the RTT. If you look closely at the Microsoft- based ping data payload, you may search that there is no such timestamp. The text initiate with the alphabet.

ICMP opinion number used: - Microsoft- based O/S use unique values for this field. The parameter will not exchange. The values are 256, 512, and 768. With UNIX and UNIX- like O/S, the ICMP ID will be the compute ID assigned to ping when run. This means that the parameter for UNIX will sustain change.

ICMP sequence numbers: Both UNIX and Microsoft based systems enhanced grow Sequence (Seq) numbers with 256. However, UNIX systems always imitate the Seq number at 0, whereas Microsoft systems start the Seq number at the last Seq number used in the beginning iteration of ping plus 256. For instance, in the earlier instance, the Microsoft version of ping set the starting Seq number at 5,120, sense that the initial time ping was used, the last Seq was number 4,864. This will be reset to 0 only when the system reboots. Few black hats use various types of ICMP instruments to build ICMP query messages or malformed ICMP queries. This method can be again using this knowledge to also view some of those tools. For instance, this is how it would search an ICMP Resonance packet make not by an O/S but by the application Hping2.Hping2 is a network tool unfailing to send custom IP packets and to demonstrate goal replies like ping does with ICMJP replies. Hoping2 handles divide arbitrary packet body, and size and can be used to transmit files under enable protocols.

A.a.c. Secondary Support

Processes, that rely solely on the IP/ICMP view shows in common traffic, are bounded in the correct about the task. One foundation of this process though is that they only give a means of the operating system. Weaknesses may or may not exist, and again searching must be started functioning to expand if this is the case. While suit unfailing for the white hat for most purposes like accounting, this is not suit unfailing to a would-be hacker. Normally put, more knowledge is needed.
Again searching is also required to known avenues of penetrability, as well. A process available to slowly view task operating systems and version, as well as vectors of hack, based on data sent by client programs. While normally, it is tough. The correctness of this process is also quite high in

most cases. Four process of Finger stamping a system are demonstrated, with sample data given.

Finger stamping using Network Client Applications

Another process to only Finger stamping the O/S is to perform an opinion by using client programs quite a number of network clients send expose knowledge about their host systems, either directly or indirectly. We use program level known to map back to the operating system, either directly or indirectly. One very large benefit to the process explained here is that in some conditions; great more accurate known can be advantage about the client. Because of stack equality, most Windows systems, having 95, 98 and NT 4.0, look too similar to differentiate. The client application, however, is willing to visible this known. This gives not only a scale of the goal's likely operating system, but also a likely vector for entrance. Most of these client programs have number of safety holes, to which one can point malicious data. In some problems, this can gives the key knowledge required to begin intrusive a network, and one can execute more slowly. In most cases it gives a starting point for the analysis of weaknesses of a network. One major bound of this process, however, comes when a system is copy another to give access to client software. This includes Solaris and SCO's support for Linux binaries. As such, under these occasions, the data should be taken with few care and expanded in the presence of other known. This limitation, however, is similar to the limitation that IP stack tweaking can place on passive Finger stamping at the IP level or the effect on active scanning from these adjustments or firewalling. Four different type of network clients are discussed here which provide suit unfailing Finger stamping information. Email clients, who leave telltale information in most cases on their messages; Usenet clients, who, like mail programs, litter their posts with client system known; web browsers, which send client known with every request; and even the ubiquitous telnet client, who sends such information more quietly, but can just as strenuously, fingerprint an operating system. Knowing this, one now only needs to harvest the network for this information and map it to source addresses. Different tools, plus sniffers, both normal and particular, and even web detection will yield this known. A slow analysis of systems can be suddenly responding.

Mail Clients

Electronic mail is one of the huge types of traffic the network. Nearly each one uses the Internet on a normal basis uses email in those transaction sessions. They not only get mail, but also send a good data of mail, too. Because it is ubiquitous, it builds a particularly glamorous avenue for system Finger stamping and in the last approach. Within the headers of nearly every mail message is some form of system view. Either through the use of crafted message view tags, as used by Eudora and Pine, or by clear header information, such as headers implemented by Outlook clients or CDE mail clients. The scope of this process, both in terms of known gained and the potential effect should not be underestimated. Anything spread by email, inclusive ones that are used to steal passwords from systems, should show the strenuousness of this process.

Usenet Clients

There is meaning equal to e-mail, Usenet clients leave significant known in the headers of their posts which show known about their host O/S. One great benefit to Usenet, as opposed to e-mail or even web trace, is that posts are distributed. As such, we can be remote and gathering data on hosts without their information or ever having to gain entry into their network. Between the different newsreaders normally used, copious host info is having in the headers. The popular UNIX newsreader 'tin' is among the worst delinquent of expose host known. O/S versions, processors and programs are all listed in the 'User-Agent' field, and when double to the NNTP-Posting-Host information; a remote host fingerprint has been respond. The standard web browsers also leave copious known about themselves and their host systems, as they do with HTTP requests and mail.

Using Web Traffic

There is noticeable normal and highly strenuous means of Finger stamping a goal is to follow the web browsing that gets done from it. Generally each system in use is a workstation, and nearly everyone uses their web browsers to spend part of their day. And just about every browser sends too much knowledge in its 'User-Agent' field. RFC 194513 notes that the 'User-Agent' field is not need in an HTTP 1.0 request, but can be used. The authors state, "User agents should have this field with requests." They cite statistics as well as on them scaring of data to meet features or bounded of browsers. The draft standard for HTTP version 1.1 requests, RFC 2616, also notes similar usage of the 'User-Agent' field. This known can be collected in two ways. First, we could run a website and turn on logging of the User-Agent field from the client. Normally implement a lot of hits and watch the data comes in. Get on Slashdot, advertise some pornographic material, or mirror some popular software (like warez) and you're ready to go. Next, we can sniff web traffic on our visible part. While almost any sniffer will work, one of the easiest for this type of work is urlsnarf from the dsni package from Dug Song. Instance of browsers that send not only their program knowledge, such as the browser and the version, but also the O/S which the host runs include: Netscape (UNIX, MacOS, and Windows) Internet Explorer One shining instance of a browser that doesn't send outer information is Lynx.

Web Server Finger stamping

In extra equal path as one can use the strings sent during requests by the users to known what system type is in use, one can follow the replies sent back by the server to know what type it is; Further we will use ngrep, this time mapping the expression 'server:' to collect the web server type. While different about the predating system known are lost, this works to passively gather penetrability information about the goal server. This can be coupled to other known to judge how best to work with an attack.

Telnet Clients

Basically Telnet is a network protocol used on the Internet or LAN to give a bidirectional interactive text-oriented communications facility using a virtual terminal connection. While telnet is no huge in large use due to the fact that all of its data is sent in plain text, with authentication data, it is still used widely enough to be of use in Finger stamping goal systems. What is engrossing is that it not only gives us a mechanism to gather O/S data, it gives us the especial program in use, which can be of value in knowing a mechanism of entry. This process of system Finger stamping is not unique to this paper. User data is interspersed in-band with Telnet control information in an 8-bit byte oriented data connection over the Transmission Control Protocol (TCP) setup presented by a safety analyst from Bell Labs had a honey-pot system set up that one would telnet to. What is interesting to note is that each client behaves in a unique way, even different client applications on the same host type. Similarly, the telnet server, running a telnet daemon, can be fingerprinted by following the negotiations with the client. This information can be gathered directly, using a wedge application, or a honey-pot as demonstrated, or it can be sniffed off the network in a truly passive fashion. We discuss below gathering data about both the client system and the server being connected to. The same principles apply to both host opinion methods. Telnet provided access to a command-line interface (usually, of an operating system) on a remote host. Most network equipment and operating systems with a TCP/IP stack support a Telnet service for remote configuration (including systems based on Windows NT). Because of security issues with Telnet, its use for this purpose has waned in favor of SSH.

ACTIVIATION FINGER STAMPING

Using for remote active O/S Finger stamping by many tools. Those have their self Finger stamping methods. In deep examine of two such tools: Nmap, and Xinvestigation2 with the goal to present how these tools perform, and to know the benefit and loss they each offer. Process of knowing the identity of a remote host's O/S through remote active O/S Finger stamping. Sending packets to the remote host and analyzing the resist for working it done by actively. Studying which O/S is executing on a remote host can be very value unfailing for both admin (white-hats) and attacker (black-hats).

D.a ACTIVE IP PACKET for FINGER STAMPING

Active IP packet is predominant form of OS Finger stamping, provoking the goal into selecting a resist and analyzing it safely. A large amount of data can be gleaned about the resist to a safely build network packet. The ICMP, TCP, UDP are three types of common IP packet all used in this method and differ type valid and invalid packets are sent to the host to correct the guess of the OS. The most normal method in use given below:

1) FIN Investigating

To known open port with the FIN flag set by sent a single packet. For linking this flag is usually signals the end and as such is not desire without a connection being initially built. The defined behavior as defined in RFC 793 is to normally omit the packet; however, many stacks send a RST packet back. By that distinct is the value to begin making a fingerprint. The wrights RFC 793 behavior is to NOT perform, but more broken executing like as Microsoft Windows, HP/UX, BSDI, MVS, IRIX and CISCO send a RESET back. Most current tools utilize this Technique.

2) TCP ISN Sampling

Keeping track of the serial number of bytes successfully transmitted with a connection TCP uses series numbers. When a starting connection try is build to a host the O/S chooses an first series number to begin the method. This choice can be anything from a unique value; by random grows of beginning values, algorithms based on the host's internal clock, to true random systems. Here is especial to look that the predictability of this ISN also has safety implications, leaving the host open to hacks same to the Mitnick attack. The concept here is to find process in the beginning series numbers chosen by TCP execution when performing to a connection request. These can be divided with kind of in to many groups such as the classic 64K (many old UNIX boxes), Random increments newer versions of Solaris, IRIX, FreeBSD, Digital UNIX, Cray, and many others), True random" (Linux 2.0.*, OpenVMS, newer AIX, etc). Windows boxes and a few others use a "time protégé" model where the ISN is growth by a small constant value each time period.

Require less to say, this is generally as simply defeated as the old 64K behavior. The machines use the correct same ISN .You can also subclass category such as random growth by calculating different, most common divisors, and other functions on the set of series numbers and the distinct among the numbers. It should be noted that ISN build has main safety implications. Nmap is the application I have seen to use this for OS opinion.

3) The BOGUS Flag Investigation

Concept is to fit an unidentified TCP "flag" (64 or 128) in the TCP header of a SYN packet. Linux boxes prior to 2.0.35 keep the flag set in their resist. I have not found any another O/S to have this error. Some operating systems feel to reset the link when they get a SYN+BOGUS packet. This behavior could be useful in view them.

4) Don't Divided bit

More operating systems are imitated to set the IP "Don't Divided" bit on few of the packets they send. This gives different response advantage. In any problem, not all OS's do this and some do it in differ problems, so by paying attention to this bit we can glean even more known about the task O/S.

5) ICMP Error Quoting

There are different facets of the design of ICMP bug packets are useful. ICMP bug packets are needed to back a small portion of the original message for view purposes; however, some stack executions back more than required. This is particularly useful as it permission some basic O/S view of machines that no hearing ports open at all. The RFCs locate that ICMP bug messages quote some small value of an ICMP message that causes differ bugs. For a port un-reach unfailing message, almost all generates send only the needed IP header + 8 bytes back. However, Solaris sends back a bit more and Linux sends back even more than that. The pretty with this is it allows nmap to identity Linux and Solaris hosts even if they don't have any ports listening.

6) ICMP Error Message Quenching

Few intelligent operating systems follow the RFC 1812 concept to bind the value at which different bug messages are sent. For instance, the Linux kernel (in net/ipv4/icmp.h) limits task un-reach unfailing message execute to 80 per 4 seconds, with a 1/4 second charges if that is increase. One way to test this is to send a amount of packets to few random high UDP port and count the number of un-reach unfailing find. I have not seen this used before, and in fact I have not added this to n map unless for use in UDP port scanning. This examine would build the O/S detecting take a bit huge since you required to send a amount of data packets and wait for them to back. Packets loss on the network may be a bad result.

7) ICMP Error Message Resonance Integrity

Required to include some of the actual ICMP packet that caused the error by ICMP error message packets, for generation use a copy of the actual as a template for building the reply packet is makes it simple. 'Scratch' area is packet space can left telltale spurious data that search the O/S that build it. I got this concept from something Theo De Raadt sends to comp.safety.unix. As detail before, machines have to send back part of your actual message along with a port un-reach unfailing bug. Yet some machines tend to use your headers as 'scratch space' in time of starting method and so they are a bit packed by the time you get them back. For instance, AIX and BSDI send back an IP 'total length' field that is 20 bytes too high. Some BSDI, FreeBSD, Open BSD, ULTRIX, and VAXen come up the IP ID that you sent them. While the checksum is going to exchange due to the changed TTL anyway, there are few machines (AIX, FreeBSD, etc.) which send back an inconsistent or 0 checksum. Similar object goes with the UDP checksum.

8) ICMP Error Message Type of Service (TOS)

Whole generate return a 0 (zero) in the TOS (Type of Service) field for ICMP port un-reach unfailing data. Linux presently returns a various value in this field building it efficient to broadly view. I see at the type of service (TOS) value of the packet sent return For the ICMP port un-reach unfailing messages. Almost all execution uses 0 for this ICMP bug although Linux uses 0xC0. This does not show one of the valid TOS values, but instead is part of the unused AFAIK precedence field. Nothing known why this is group set, but if they change to 0 we will be unfailing to keep viewing the classic versions and we will be unfailing to view among classic and modern.

9) Dividedation Handling

That is better method of Thomas H. Ptacek of Secure Networks, Inc now self by a bunch of Windows users at NAI. This gets benefit of the fact that various execution often managing collapsing IP divided differently. Some will rewrite the classic part with the modern and in other cases the old stuff has precedence. There are more problems you can use to know how the data packet was again.

10) ICMP Error Message Limiting

Suggests bound the rate by RFC 1812 at which ICMP bugr messages are sent. Few IP stacks generate this suggestion including Linux, Solaris whereas Windows hosts do not10. This method is only VI unfailing on safe connections to the remote host and extends tracing time; so, is normally not executed.

11) TCP Options

These are enlargement to the TCP protocol to improve best response in un-trusted or high latency networks 7. TCP choice has been added as TCP RFCs over time as requires dictated and the patterns of compliance in resist can visible the underlying O/S. Interestingly it is not just the number of choice a stack supports that can view it, but also the series in which the options are returned 8. Many other differences also exist and are used to a lesser extent. These given below:

12)IPID Sampling

Utilizing a system wide counter by many O/S for IPID execution, next more new execution either randomize this number or set it to 0. Knowledge can be leaned from the option of IPID as to the source OS. Further, predict unfailing IPID parameters can have significant safety generation exterior of O/S Finger stamping, having whole silent port tracing.

13) TCP Timestamp

TCP option and hence is not supporting by all IP execution; can be using to know O/S type. It can also be used to know host uptime if executed and the update frequency are known. Nearly all active Fingers stamping tools use some or the entire above test to obtain data on a host and match the outcomes with a database of known O/S. As established the OS fingerprint database tools become becomes more extensive, enlarging the resolution of the instrument.

DISADVANTAGES

The cases with active tracing are in two fold: first, can read firer wall the data packets used to fingerprint our system, obfuscating the knowledge; next, we can search it quite simply.

CONCLUSION

There is few relatively normal process of searching a target's operating system. And there are many excellent, free network scanners for a network admin to employ when matching and maintaining a network. But no tracer is perfect. The network offers us both the mode of viewing and hiding the fingerprints of remote terminals. the remote computer attached to the network with identified by its fingerprint are unique to each computer consisting its recourses values like operating system, MAC address, visible by different header expand during network analysis techniques. And because the most basic of traffic reveals so much about the devices on the network, firewall and edge router ACLs should be maintained to allow only that traffic which is absolutely essential to production even ICMP should be carefully restricted.. TCP/IP, ICMP analysis presents a larger set of issues. Remote OS Finger stamping is a recent development on the Internet and one to watch. The ability to remotely determine, with high accuracy, the O/S of a remote host on the Internet is a powerful one. The implications of this technology are perhaps not yet fully understood; however, it is seen as enough of a threat that strategies are currently being developed to prevent and spoof OS Fingerprints. The most relevant concept to remember is the old adage "Obscurity is not Safety". The ease with which exploit tools can be scripted and used enmasse to find lower hosts largely trivializes the benefits of OS obscurity in today's world. This may change over the coming years as the larger software companies put an emphasis on network. The first developments have already occurred in this area, with OS Finger stamping worm toolkits being developed to refine attacks. Malformed packets sent during active Finger stamping can be filtered by a firewall or responded to with "smoke screens". Manually changing IP Personality settings is the strongest defense for both active and passive Finger stamping, but it carries substantial negative consequences. Using a patch or OS option is usually the better choice. Of course, the mind of a trained, skilled administrator is ultimately the best single tool for OS Finger stamping. By there is little implication both at kernel and process level to stop the search of the OS fingerprints of remote system. But creating one is not the rightful solution. The need is answered by carefully testing the weaknesses of the system O/S as per vendor specified. Taking steps in chronological order defending each feasible method of attack measure should be taken to defeat TCP/IP stack Finger stamping and ICMP pattern sampling. O/S and Application Fingerprinting Techniques because these scanners are so readily available, it should be obvious that white-hat admin will not be the only single using them on your network. You should know these tracers are launched against your networks,

References

[1] Arkin, Ofir. "ICMP Usage in Scanning – The Complete Know How." June 2001. ICMP_Scanning_v3.0.pdf , May 2, 2003.

[2] Arkin, and Yarochkin. "Xprobe v2.0: A "Fuzzy" Access to Remote Active Operating System Finger stamping." August 2, 2002.

[3] Beck, Rob. "Passive-Aggressive Resistance: OS Finger stamping Masquerade"

[4] David Barroso Berrueta, "A practical access for defeating Nmap OS Finger stamping", November 2002.

[5] Dethy, "Examining port scan methods – Analysing Audible Techniques."

[6] Ethereal: A Network Protocol Analyzer.

[7] Fyodor, "Remote OS Search via TCP/IP Stack Finger stamping", June 11,

[8] Finger stamping: The Complete Documentation,

[9] Gael Roualland and Jean-Marc Saffroy, "IP Personality", URL:

[10] Honeynet Project, "Know Your Enemy: Passive Finger stamping, Opinioning remote hosts, without them knowing",

[11] Insecure.org. "N map Man Page."

[12] Introduction to Network Safety, URL:

[13] James Hurnall, "N map tutorial",

[14] Kathy Wang, "Frustrating OS Finger stamping with Morph", Syn Ack Labs,DEFCON 12

[15] Know Your enemy, The Honey net Project by Addison Wesley,2002

[16] Mark Wolfgang, "Host Discovery with n map", Nov 2002

[17] Matthew Smart G. Robert Malan Farnam Jahanian, "Defeating TCP/IP Stack
Finger stamping", 9th USENIX Safety Symposium Paper Pp. 229–240 ,JULY -2000

[18] Mick Bauer,"Checking Your Work with Scanners, Part I (of II): n map ",Linux Journal archive Volume 2001 ,Issue 85es,article No. 13,

[19] Marco de Vivo,Eddy Carrasco, "A Review of Port Scanning Techniques",
ACM SIGCOMM Computer Communication Review

[20] Ste Jones, "Port 0 OS Finger stamping" , URL: 2003 .

[21] Netfilter and IPTinfallibles. Availinfallible

[22] Ofir Arkin, Fyodor Yarochkin, Kydyraliev , "The Present and Future of Xprobe2-The Next Generation of Active O/S Finger stamping",July,2003.

[23] Packet Agony Tutorial Outlining part 1& part 2, Syn Ack Labs,

[24] Paul A. Watson, "SLIPPING IN THE WINDOW: TCP RESET ATTACKS" Technical Whitepaper, [30] RFC describing TCP Extensions for High Performance with two of the three
TCP options used in OS search:

[25] Richard Stevens, TCP/IP Illustrated, Volume 1, Addison-Wesley.

[26] Rich Jankowski , "Scanning and Defending Networks with Nmap" Source: Linux safety.com,

[27] Robert Beverly, "A Robust Classifier for Passive TCP/IP Finger stamping",March 2004

[28] Ryan Spangler, "Analysis of Remote Active O/S Finger stamping Tools", Packet watch Research, May 2003.

[29] Thomas Glaser, Intrusion Search FAQ, "TCP/IP Stack Finger stamping Principles"

[30] Veysset, Courtay, and Heen. "New Tool And Technique For Remote O/S Finger stamping." URL:

3. Procedure for Paper Submission

a. Review Stage

Submit your manuscript electronically for review.

b. Final Stage

II. When you submit your final version, after your paper has been accepted, prepare it in two-column format, including

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

Appendix

Appendixes, if needed, appear before the acknowledgment.

Acknowledgment

.

First Author personal profile which contains their education details, their publications, research work, membership, achievements, with photo that will be maximum 200-400 words.

Second Author personal profile which contains their education details, their publications, research work, membership, achievements, with photo that will be maximum 200-400 words.

Third Author personal profile which contains their education details, their publications, research work, membership, achievements, with photo that will be maximum 200-400 words.