2.4.5 Bounce Scan Attack ၏အခြေခံသဘောတရား

Attacker တွေအတွက် အရေးအကြီးဆုံး အချက်တစ်ခုကတော့ မိမိရဲ့လှုပ်ရှားမှုကိုဖုံးကွယ်ထားနိုင်စွမ်းရှိဖို့ပဲဖြစ်ပါတယ်။ ဒါကြောင့်မို့လို့ Attacker တွေအနေနဲ့ သူတို့ရဲ့တိုက်ခိုက်မှုအောင်မြင်စေဖို့အတွက် Internet အတွင်းမှာကြားခံအဖြစ်ထားသုံးမယ့် System (သို့မဟုတ်) Server တစ်ခုရှိဖို့လိုအပ်ပါတယ်။ FTP bounce Scanning tool ဟာ FTP protocol ရဲ့အားနည်းချက်ကို အသုံးချပြီး တိုက်ခိုက်တဲ့ tool တစ်ခုပဲဖြစ်ပါတယ်။ FTP protocol ဟာ proxy FTP connection ကို support လုပ်ပေးဖို့အတွက် အသုံးပြုရတဲ့ protocol ပဲဖြစ်ပါတယ်။ ဒီလိုမျိုး FTP Server ရဲ့ အားနည်းချက်ကြောင့် ၎င်းကတစ်ဆင့် တိုက်ခိုက်တဲ့ attacker တွေရဲ့တည်နေရာကိုပြန်လည်ပြီးရှာဖွေဖို့ရာ ခက်ခဲပါတယ်။ ဒါကြောင့် FTP bounce ဟာ IP Spoofing နဲ့ ခပ်ဆင်ဆင်တူပါတယ်။ ဘာလို့လည်းဆိုတော့ ဒီ Attack နှစ်ခုစလုံးဟာသူတို့ဘယ်ကလာသလဲဆိုတာကိုလျှို့ဝှက်ထားနိုင်ကြလို့ပါပဲ။ ဥပမာအနေနဲ့ပြောရရင် Badwebsitexyz.Com မှာ FTP Server Connection ကိုထိန်းချုပ်ဖို့အတွက်PI (Protocol Interpreter) ကိုအသုံးပြုပါတယ်။ ၎င်းPIကို Badspiterbites.com လို့ အမည်ပေးထားပါတယ်။ ဒါ့ကြောင့်ဒီ FTP Server ကနေ Request တစ်ခုကိုတစ်နေရာရာကိုပို့တော့မယ်ဆိုရင်၎င်း PI ကနေ စတင်အလုပ်လုပ်ပါတယ်။ ဒီ PI ကနေမှတစ်ဆင့် DTP (Data Transfer Process)Server ကို Active လုပ်ပြီး၎င်း DTP ကနေပြီးတော့မှ Internet အတွင်းရှိကြိုက်တဲ့နေရာကို file တွေပို့နိုင်မှာဖြစ်ပါတယ်။

ဒါကြောင့်မို့လို့တိုက်ခိုက်ခံရသူအနေနဲ့ပြန်ပြီး Trace လိုက်တဲ့အခါမှာ Badspiterbites.com ကိုပဲရှာတွေ့မှာ ဖြစ်ပါတယ်။ မူရင်းကြားခံဖြစ်တဲ့ Badwebsitexyz.com ကိုသော်လည်းကောင်း၊ တိုက်ခိုက်တဲ့ source ကိုသော် လည်းကောင်း ရှာတွေ့မှာမဟုတ်ပါဘူး။

အထက်မှာရေးသားခဲ့တဲ့ FTP server ရဲ့အားနည်းချက်တွေကြောင့်မို့လို့ port scanner အနေနဲ့ FTP server ရဲ့ proxy အတွင်းရှိ TCP port တွေကို scan လုပ်နိုင်ပါတယ်။ ၎င်းနည်းလမ်းကို အသုံးပြုပြီး firewall နဲ့ ကာထားတဲ့ FTP server တွေနဲ့လည်းချိတ်ဆက်နိုင်ပါတယ်။ ဒါ့အပြင် block လုပ်ထားတဲ့ port တွေ၊ ဥပမာ -port 139 လိုမျိုး ပိတ်ထားတဲ့ port တွေကိုလဲ scan လုပ်နိုင်ပါတယ်။ port 139 ဆိုတာကတော့ Net BIOS session service (TCP) port ပဲဖြစ်ပါတယ်။ Window 98, ME နဲ့ NT တွေမှာ resource sharing(file and printer) အတွက်အသုံးပြုပါတယ်။ User ရဲ့ 10% လောက်ဟာ internet အသုံးပြုနေစဉ် File တွေ Folder တွေ Share ပေးထားတတ်ကြပါတယ်။ အဲဒီလို Share ပေးထားတဲ့အတွက်၎င်း Port139 ဟာပွင့်နေပါတယ်။ ဒီအတွက်ကြောင့် hacker အများစုဟာ target နဲ့ connection စယူဖို့ကြိုးစားတဲ့အခါမှာ၎င်း port ကိုပထမဆုံး ပစ်မှတ် ထားကြပါတယ်။ firewall ကတော့ ၎င်း port 139 ကိုအမြဲ block လုပ်ပေးပါတယ်။ဒါကြောင့် user တွေဟာ ၎င်း port ကို internet မှာ access မဖြစ်စေရန် firewall ကိုအသုံးပြုကြပါတယ်။

FTP bounce scanning ရဲ့အားသာချက်တွေကတော့ trace လိုက်ဖို့ခက်ခဲခြင်း၊ firewall ကိုကျော်နိုင်ခြင်း တို့ပဲဖြစ်ပါတယ်။ အဓိကအားနည်းချက်ကတော့ ဒီ နည်းလမ်းကိုအသုံးပြုရတာနေးကွေးခြင်းနှင့် FTP server အများစုက မိမိအသုံးပြုနေတာကို နောက်ဆုံးမှာတွေ့ရှိပြီး proxy

"feature" ကို disable လုပ်ကြတဲ့အတွက် ကြားခံ FTP server တစ်ခုတည်းကို ရေရှည်အသုံးမပြု နိုင်ခြင်းပဲဖြစ်ပါတယ်။

2.4.6 Nmap ဆိုတာဘာလဲ?

Nmap ဆိုတာကတော့ကြီးမားတဲ့ network တွေအတွင်းမှာ port တွေကို scan လုပ်ဖို့အသုံးပြုတဲ့ network mapper tool တစ်ခုပဲဖြစ်ပါတယ်။ ဒီ tool ဟာ host တစ်ခုထဲ အတွက်ပဲကောင်းကောင်းအလုပ်လုပ်နိုင်ပေမဲ့လူသုံးများတဲ့ tool တစ်ခုပဲဖြစ်ပါတယ်။ Nmap ရဲ့ဆောင်ပုဒ်ကတော့ TMTOWTDI(There's More Than One Way To Do It) ပဲဖြစ်ပါတယ်။ ဒီ tool ကိုတော့ perl programming language နဲ့ရေးထားတာဖြစ်ပေမဲ့ Scanner အဖြစ် အသုံးပြုရတာ အထူးကောင်းမွန်ပါတယ်။ သင်ဟာ Hacker တစ်ဦးဖြစ်ခဲ့မယ်ဆိုရင် တစ်ခါ တစ်ရံမှာ သင်ဟာလျှင်မြန်ဖို့လိုအပ်မယ်။ တစ်ခါတစ်ရံမှာ သင်ဟာလျှို့ဝှက်ဖို့လိုအပ်မယ်။ တစ်ခါ တစ်ရံမှာ firewall ကို ကျော်ဖြတ်ဖို့လည်း လို အပ်နိုင်ပါတယ်။ ဒါ့အပြင်သင် Scan လုပ်ချင်တဲ့ Protocol တွေဟာလည်း UDP, TCP ,ICMP စသည်ဖြင့်အမျိုးမျိုးကွဲပြားနိုင်ပါတယ်။ ဒါကြောင့် သင့်အနေနဲ့ Scanning mode တစ်ခုတည်းကို ပဲအသုံးပြုလို့မရနိုင်ပါဘူး။ ဒါ့အပြင် Interface အမျိုးမျိုးနဲ့ Scanner tool ၁၀ မျိုးလောက်ကို အသုံးမပြုချင်ကြပါဘူး။ ဒါကြောင့်မို့လို့ Nmap ကို တီထွင်ခဲ့ကြပါတယ်။ Nmap အနေနဲ့ အောက်မှာဖော်ပြထားတဲ့ Attack တွေကို Support လုပ်နိုင်ပါတယ်။

၁။ Vanilla TCP Connect () Scanning
၂။ TCP SYN (half open) Scanning
၃။ TCP FIN (stealth) Scanning
၄။ TCP FTP Proxy (bounce attack) Scanning
၅။ SYN/FIN Scanning
၆။ UDP recvfrom() Scanning
၇။ UDP raw ICMP port unreachable Scanning
၈။ ICMP Scanning (Ping Sweep)
၉။ Reverse-ident Scanning

Nmap မှာ တစ်ခြားစွမ်းဆောင်ရည်တွေလည်းပါရှိပါတယ်။ အဲဒါတွေကတော့ Delay Time ကို တွက်ချက်ပေးတာ၊ Pack တွေ Timeout ဖြစ်သွားရင်ပြန်ပို့ပေးနိုင်တာ၊ Port တွေ အများကြီးကို တစ်ချိန်တည်းမှာ Scan လုပ်နိုင်တာ၊ Ping တွေအများကြီးပို့ပြီး Host ကို down အောင် လုပ်နိုင်တာ၊ Target နဲ့ Port ကို ရွေးချယ်ပေးနိုင်တာ စတာတွေကိုလည်း လုပ်ဆောင်ပေးနိုင် ပါတယ်။ အောက်မှာဖော်ပြထားတဲ့ ဥပမာတွေကတော့ Nmap 4.53 မှာ အသုံးပြုနိုင်တဲ့ Scan attack command တွေပဲဖြစ်ပါတယ်။

```
Nmap 4.53 ( http://insecure.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
```

```
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file

HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sP: Ping Scan - go no further than determining if host is online
  -PN: Treat all hosts as online -- skip host discovery
  -PS/PA/PU [portlist]: TCP SYN/ACK or UDP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO [protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver

SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
  --traceroute: Trace hop path to each host
  --reason: Display the reason a port is in a particular state

PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports consecutively - don't randomize
  --top-ports <number>: Scan <number> most common ports
```

--port-ratio <ratio>: Scan ports more common than <ratio>

SERVICE/VERSION DETECTION:
-sV: Probe open ports to determine service/version info
--version-intensity <level>: Set from 0 (light) to 9 (try all probes)
--version-light: Limit to most likely probes (intensity 2)
--version-all: Try every single probe (intensity 9)
--version-trace: Show detailed version scan activity (for debugging)

SCRIPT SCAN:
-sC: equivalent to --script=safe,intrusive
--script=<Lua scripts>: <Lua scripts> is a comma separated list of
directories, script-files or script-categories
--script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
--script-trace: Show all data sent and received
--script-updatedb: Update the script database.

OS DETECTION:
-O: Enable OS detection
--osscan-limit: Limit OS detection to promising targets
--osscan-guess: Guess OS more aggressively

TIMING AND PERFORMANCE:
Options which take <time> are in milliseconds, unless you append 's'(seconds), 'm'
(minutes), or 'h' (hours) to the value (e.g. 30m).
-T[0-5]: Set timing template (higher is faster)
--min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
--min-parallelism/max-parallelism <time>: Probe parallelization
--min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies probe round
trip time.
--max-retries <tries>: Caps number of port scan probe retransmissions.
--host-timeout <time>: Give up on target after this long
--scan-delay/--max-scan-delay <time>: Adjust delay between probes

FIREWALL/IDS EVASION AND SPOOFING:

-f; --mtu <val>: fragment packets (optionally w/given MTU)
-D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
-S <IP_Address>: Spoof source address
-e <iface>: Use specified interface
-g/--source-port <portnum>: Use given port number
--data-length <num>: Append random data to sent packets
--ip-options <options>: Send packets with specified ip options
--ttl <val>: Set IP time-to-live field
--spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address
--badsum: Send packets with a bogus TCP/UDP checksum

OUTPUT:

-oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rIpt kIddi3, and Grepable format, respectively, to the given filename.
-oA <basename>: Output in the three major formats at once
-v: Increase verbosity level (use twice for more effect)
-d[level]: Set or increase debugging level (Up to 9 is meaningful)
--open: Only show open (or possibly open) ports
--packet-trace: Show all packets sent and received
--iflist: Print host interfaces and routes (for debugging)
--log-errors: Log errors/warnings to the normal-format output file
--append-output: Append to rather than clobber specified output files
--resume <filename>: Resume an aborted scan
--stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
--webxml: Reference stylesheet from Insecure.Org for more portable XML
--no-stylesheet: Prevent associating of XSL stylesheet w/XML output

MISC:

-6: Enable IPv6 scanning
-A: Enables OS detection and Version detection, Script scanning and Traceroute
--datadir <dirname>: Specify custom Nmap data file location
--send-eth/--send-ip: Send using raw ethernet frames or IP packets
--privileged: Assume that the user is fully privileged
--unprivileged: Assume the user lacks raw socket privileges

-V: Print version number

-h: Print this help summary page.

EXAMPLES:

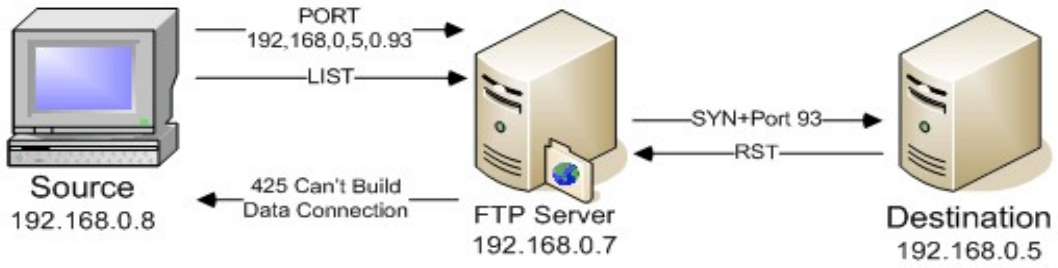
nmap -v -A scanme.nmap.org

nmap -v -sP 192.168.0.0/16 10.0.0.0/8

nmap -v -iR 10000 -PN -p 80

2.4.7 Nmap ကို Bounce scan attack တွင်အသုံးပြုခြင်း

Bounce attack လုပ်တော့မယ်ဆိုရင် ပထမဆုံး nmap အနေနဲ့ middleman (ကြားခံ) အနေနဲ့အသုံးပြုမည့် FTP Server ထဲကို Login လုပ်ဖို့လိုအပ်ပါတယ်။ ဒါကတော့ Nmap အနေနဲ့ ကြားခံ Server နဲ့ Connection စယူလိုက်တာပါ။ Connection လုပ်လို့ရပြီဆိုတာနဲ့ Nmap ကနေ ကြားခံ FTP Server ရဲ့ IP address တွေနှင့် TCP port အားလုံးရဲ့ Data connection ကို ထိန်းချုပ်ဖို့အတွက် PORT command တွေပေးပို့ပါတယ်။ ဒီ Port command မှာ တူညီတဲ့ Syntax တွေပါဝင်ပါတယ်။ Port command ဟာ ဂဏာန်း ၆ လုံးပါဝင်ပါတယ်။ ပထမ ၄ လုံး ကတော့ FTP server ရဲ့ IP address ကို ကိုယ်စားပြုပါတယ်။ နောက်ဆုံး ၂ လုံးကတော့ port number ကိုကိုယ်စားပြုပါတယ်။ Port number ကို decimal အဖြစ် တွက်ချက်ဖို့အတွက် ဒုတိယ နောက်ဆုံးဂဏာန်းကို 256 နဲ့ မြောက်ပြီး နောက်ဆုံးဂဏာန်းကိုပေါင်းပေးရမှာဖြစ်ပါတယ်။ ဥပမာ port command ဟာ 192,168,0,5,2,44 ဖြစ်ခဲ့ရင်192.168.0.5 ဟာ IP address ဖြစ်ပြီး port ကတော့ (2*256)+44=556 ဖြစ်ပါတယ်။ Nmap အနေနဲ့ PORT command ကိုသတ်မှတ်ပြီးတာနဲ့ LIST command ကိုလဲ FTP server ဆီပို့ပါတယ်။ သတ်မှတ်ထားတဲ့ IP address နဲ့ TCP port တွေကို data connection ရယူဖို့ဖြစ်ပါတယ်။ ဒီအခါမှာ ကြားခံ FTP server ဟာ nmap ကပို့တဲ့ PORT command တွေရဲ့ခိုင်းစေချက်အတိုင်း လိုက်လုပ်ဖို့ အဆင်သင့် ဖြစ်နေပါတယ်။ အောက်ဖော်ပြပါပုံကတော့ FTP server ကနေ တိုက်ခိုက်တဲ့သူဆီကို destination server နဲ့ connection ယူလို့မရဘူးဆိုတာ replay ပြန်တဲ့ပုံဖြစ်ပါတယ်။

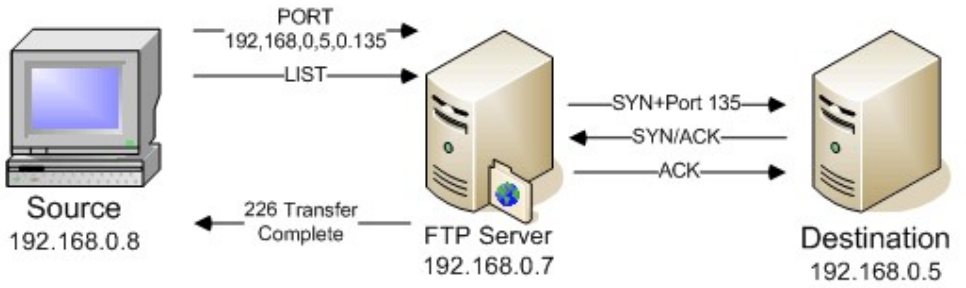


ပုံ ၁

သူ့ရဲ့ data ကတော့အောက်မှာပြထားတဲ့အတိုင်းတွေ့ရမှာပါ။

```
    Source        Destination   Summary
-------------------------------------------------------------------------------
[192.168.0.8] [192.168.0.7] FTP: C PORT=37205   PORT 192,168,0,5,0,93
[192.168.0.7] [192.168.0.8] FTP: R PORT=37205   200 PORT command successful.
[192.168.0.8] [192.168.0.7] FTP: C PORT=37205   LIST
[192.168.0.7] [192.168.0.5] TCP: D=93 S=20 SYN SEQ=474501024 LEN=0 WIN=65535
[192.168.0.5] [192.168.0.7] TCP: D=20 S=93 RST ACK=474501025 WIN=0
[192.168.0.7] [192.168.0.8] FTP: R PORT=37205   425 Can't build data connection: Connection
refused.
```

အောက်ပါပုံကတော့ သွားရောက်ရှိတ်ဆက်တဲ့ port ဟာ မိမိဆီက request ကိုလက်ခံလိုက်တဲ့ အတွက် connection တည်ဆောက်တဲ့ပုံဖြစ်ပါတယ်။



ပုံ ၂

```
Source        Destination   Summary
-------------------------------------------------------------------------------
[192.168.0.8] [192.168.0.7] FTP: C PORT=37205   PORT 192,168,0,5,0,135
[192.168.0.7] [192.168.0.8] FTP: R PORT=37205   200 PORT command successful.
[192.168.0.8] [192.168.0.7] FTP: C PORT=37205   LIST
[192.168.0.7] [192.168.0.5] TCP: D=135 S=20 SYN SEQ=4240951199 LEN=0 WIN=65535
[192.168.0.5] [192.168.0.7] TCP: D=20 S=135 SYN ACK=4240951200 SEQ=2193395373 LEN=0
WIN=65535
[192.168.0.7] [192.168.0.5] TCP: D=135 S=20     ACK=2193395374 WIN<<1=65700
[192.168.0.7] [192.168.0.8] FTP: R PORT=37205   150 Opening ASCII mode data connection for
'/bin/ls'.
[192.168.0.7] [192.168.0.8] FTP: R PORT=37205   226 Transfer complete.
[192.168.0.7] [192.168.0.5] FTP: R PORT=135   Text Data
```

အောက်မှာပြထားတာတွေကတော့ nmap ကနေ ရရှိထားမဲ့ FTP bounce scan ရဲ့ result တွေပဲဖြစ်ပါတယ်။ဒီမှာထူးခြားတာကတော့ nmap ဟာ firewall ကိုကျော်နိုင်ပေမဲ့ လိုလိုမယ်မယ် command line မှာ (-P0) ဆိုပြီး "don't ping " ဆိုတဲ့ reminder တစ်ခုထပ်ထည့်ထားတာပဲ ဖြစ် ပါတယ်။

```
# nmap -v -b anonymous:anon@192.168.0.7 192.168.0.5
Hint: if your bounce scan target hosts aren't reachable from here, remember to
use -P0 so we don't try and ping them prior to the scan
```

```
Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2005-04-23 20:37 EDT
Resolved ftp bounce attack proxy to 192.168.0.7 (192.168.0.7).
Attempting connection to ftp://anonymous:anon@192.168.0.7:21
Connected:Login credentials accepted by ftp server!
Initiating TCP ftp bounce scan against 192.168.0.5 at 20:37
Discovered open port 6969/tcp on 192.168.0.5
Discovered open port 135/tcp on 192.168.0.5
Discovered open port 139/tcp on 192.168.0.5
Discovered open port 445/tcp on 192.168.0.5
Scanned 1663 ports in 9 seconds via the Bounce scan.
Host 192.168.0.5 appears to be up ... good.
Interesting ports on 192.168.0.5:
(The 1659 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
6969/tcp open  acmsoda
MAC Address: 00:11:43:43:A8:34 (Dell   (WW Pcba Test))
Nmap finished: 1 IP address (1 host up) scanned in 20.602 seconds
                   Raw packets sent: 2 (68B) | Rcvd: 1 (46B)
#
```

ထပ်တင်ပြချင်တာကတော့ bounce attack မှာအသုံးပြုတဲ့ nmap command တွေပဲ ဖြစ်ပါတယ်။ ပထမဆုံး ကျွန်တော်တို့အနေနဲ့ vsftpd packet တွေကို FTP server ထဲကို ထည့်သွင်းမှာဖြစ်ပါတယ်။ ဒီအခါမှာ ကျွန်တော့်အနေနဲ့ FTP server ထဲကို upload လုပ်နိုင်တဲ့ anonymous user တစ်ယောက်ဖြစ်ရပါမယ်။ဒီအခါမှာ FTP server ဆီကိုဖော်ပြပါ command ပို့မှာဖြစ်ပါ တယ်။

```
ftp> passive
Passive mode off.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    8 0        0            4096 Aug 30 12:10 pub
226 Directory send OK.
ftp>
```

ဒီအခါမှာ ကျွန်တော်တို့အနေနဲ့ FTP bounce attack ကိုစတင်အသုံးပြုဖို့ အတွက် အောက်ပါ command ကို အသုံးပြုပါတယ်။

nmap -b anonymous:""@ w.x.y.z <http://172.16.1.251/> a.b.c.d

တကယ်လို့ connection မရခဲ့ရင် အောက်ပါ error ကို တွေ့ရမှာဖြစ်ပါတယ်။

[root () gdrd5 ~]# nmap -v -b anonymous:""@w.z.y.z a.b.c.d

ဒီအခါမှာ -OP ကို အသုံးပြုရမှာဖြစ်ပါတယ်။ ဘာလို့လဲဆိုတော့မိမိရဲ့တည်နေရာကနေ target ဆီကိုချိတ်ဆက်လို့မရတဲ့အတွက် အသုံးပြုတာဖြစ်ပါတယ်။ ၄င်းနောက် အောက်ပါ command အ တိုင်းတွေ့ရမှာဖြစ်ပါတယ်။

Starting Nmap 4.20 at 2007-08-31 15:15 IST
Resolved ftp bounce attack proxy to w.x.y.z
Initiating ARP Ping Scan at 15:15
Scanning a.b.c.d [1 port]
Completed ARP Ping Scan at 15:15, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:15
Completed Parallel DNS resolution of 1 host. at 15:15, 0.06s elapsed
Attempting connection to ftp://anonymous:@w.x.y.z:21<ftp://anonymous:@172.16.1.251/>
Connected:220 Welcome to blah FTP service.
Login credentials accepted by ftp server!
Initiating TCP ftp bounce scan against somedomain.in <http://hgdrd1.cdacbangalore.in/>(a.b.c.d) at 15:15
Your ftp bounce server doesn't allow privileged ports, skipping them.
Your ftp bounce server doesn't allow privileged ports, skipping them.
Your ftp bounce server doesn't allow privileged ports, skipping them.
Your ftp bounce server doesn't allow privileged ports, skipping them.
Your ftp bounce server sucks, it won't let us feed bogus ports!
[root () gdrd5 ~]#

ဒီအခါမှာ ကျွန်တော်တို့အနေနဲ့ /etc /vsftpd / vsftpd.conf ထဲမှာအောက်ပါအတိုင်း configuration file တစ်ခုရရှိပြီဖြစ်ပါတယ်။

anonymous_enable=YES
write_enable=YES
anon_upload_enable=YES
connect_from_port_20=YES

2.3 TCP FIN (stealth) Scanning

TCP scan ရဲ့လုပ်ဆောင်ချက်ကတော့ connection တစ်ခုရယူဖို့ကြိုးစားတာပဲဖြစ်ပါတယ်။ နောက်လုပ်ဆောင်ချက်တစ်ခုကတော့ packet အမှားတွေကို target port ဆီကို ပို့လိုက်တာပါ။ ဒီအခါမှာ၄င်းပို့တာ(target port အနေနဲ့မိမိပို့လိုက်တဲ့ packet တွေကိုလက်ခံပြီး error message ပြန်လာအောင်) error message ပြန်ပို့ပြီးပိတ်သွားမှာဖြစ်ပါတယ်။ Scanner အနေနဲ့ပွင့်နေတဲ့ connection တစ်ခုကိုပိတ်သွားအောင် FIN package တွေပို့ပေးရပါတယ်။ ၄င်းနောက်ပိတ်နေတဲ့ ဒီ port ကနေ ဒီ FIN packet တွေကို RST နဲ့ reply ပြန်ပေးပါတယ်။ ဒီအခါမှာ တခြားပွင့်နေတဲ့ post တွေဟာမိမိဆီကပို့တဲ့ FIN packet အပေါ်ဂရုပြုမိခြင်းမရှိတော့ပါဘူး။ ဒါဟာ TCP အတွက်လိုအပ်တဲ့အလေ့အထ တစ်ခုပါ။

တကယ်လို့သာ target port အလုပ်မလုပ်ခဲ့ရင် operating system အနေနဲ့ error message တစ်ခုထုတ်ပေးပါလိမ့်မယ်။ တကယ်လို့အလုပ်လုပ်ခဲ့ရင် operation system အနေနဲ့ ဂင်လာတဲ့ packet တွေကိုတိတ်တဆိတ်ချထားပေးမှာဖြစ်ပါတယ်။ ဒါကြောင့်မို့လို့ ကျွန်တော်တို့ အနေနဲ့ target port ရဲ့အခြေအနေကို သိရှိရမှာဖြစ်ပါတယ်။ အကယ်၍သာဒီပို့လိုက်တဲ့ pocket ဟာ သတိပေးစနစ် တစ်ခုခုအပေါ်သွားထားမိတာ ဒါမှမဟုတ် firewall ရဲ့ block လုပ်တာခံခဲ့ ရရင်တော့ scan လုပ်တာအောင်မြင်မှာမဟုတ်ပါဘူး။

နောက်နည်းလမ်းတစ်ခုကတော့ TCP packet တွေထဲမှာ XMAS scans တွေကို flag တွေ အနေနဲ့ ထည့်သွင်းအသုံးပြုတာနဲ့ bits တွေဘာမှမထည့်ပဲအသုံးပြုတဲ့ NULL scans တို့ပဲဖြစ် ပါတယ်။ တကယ်လို့သာဒီ scan အပေါ်မှာ operating system အမျိုးမျိုးကနေ response အမျိုးမျိုးပြန်လာခဲ့ရင်တော့ ဒီ response အပေါ်မူတည်ပြီး ၄င်း OS တွေဟာဘာတွေလဲ၊ Version ဘယ်လောက်လဲ၊ သူ့ရဲ့ patch level ကဘယ်လောက်လဲ စတာတွေကိုသိရှိနိုင်မှာဖြစ်ပါတယ်။

အထက်ဖော်ပြခဲ့တဲ့ TCP scan တွေဖြစ်ကြတဲ့ FIN scan (-sf), Xmas Tree Scan (-sX) နဲ့ Null Scan (-sN) တွေ အသုံးပြုဖို့အတွက် လိုအပ်ချက်တွေကတော့ အောက်ပါအတိုင်း ဖြစ်ပါတယ်။

၁။ Require Privileged Access : YES

၂။ Identifies TCP Ports : YES

၃။ Identifies UDP Ports : NO

ဒီ scan နည်းလမ်း ၃ ခုကို group တစ်ခုထဲအဖြစ် သတ်မှတ်ထားပါတယ်။ ဘာလို့လဲ ဆိုတော့ သူတို့တစ်ခုချင်းဆီရဲ့ လုပ်ဆောင်ချက်က တူညီတဲ့အတွက်ပဲ ဖြစ်ပါတယ်။ သူတို့ကို "stealth" scan တွေလို့ခေါ်ကြပါတယ်။ ဘာကြောင့်လဲဆိုတော့ သူတို့အနေနဲ့ single frame တစ်ခုကို TCP hand shaking ဒါမှမဟုတ် တစ်ခြား packet တွေမပါပဲ TCP port ဆီကို ပေးပို့ကြပါတယ်။ ဒီ scan type ရဲ့ ရည်ရွယ်ချက်ကတော့ single frame ပို့ပြီး single response ပြန်ရဖို့ပါပဲ။

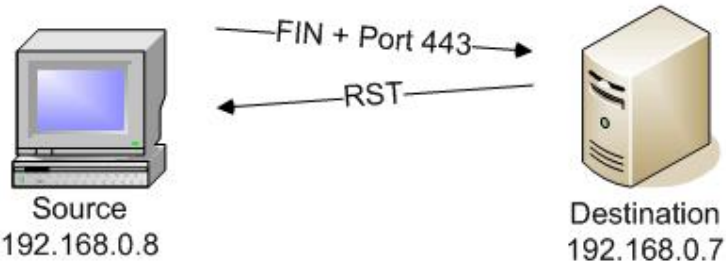
RFC 793 အရ Transmission Control Protocol ဟာ ပိတ်နေတဲ့ TCP port ဆီကို information တစ်ခုရောက်လာခဲ့ရင် RST frame တစ်ခု reply ပြန်ပေးရပါတယ်။ ဒီ TCP port ကတော့ဘာမှအလုပ်လုပ်ခြင်းမရှိပါဘူး။ ဒီ stealth scan တွေအတွက် nmap ကို အသုံးပြုတဲ့အခါ

မှာ၄င်း reply ပြန်လာတဲ့ RST frame ပေါ်မူတည်ပြီးဒီ port ဟာ ပွင့်နေလား၊ ပိတ်နေလား ဆိုတာ ဆုံးဖြတ်ပေးပါတယ်။ ဒါပေမဲ့လည်း တစ်ခါတစ်လေမှာပို့လိုက်တဲ့ FIN packet ကို firewall က တွေ့ရှိ ပြီး response မပြန်ပဲ drop လုပ်ထားတာရှိပါတယ်။ ဒါကြောင့်လဲ reply မပြန်တာဟာ port အလုပ်လုပ်နေလို့လား ဒါမှမဟုတ် firewall က ဖမ်းယူသွားတာလားဆိုတာခွဲခြားဖို့ ခက်ပါ တယ်။ ဘာ့ကြောင့်လဲဆို တော့ post ဟာပွင့်နေရင်လဲ reply မပြန်တတ်ပါဘူး။

TCP/IP stacks အမျိုးမျိုးဟာ ဒီ scan တွေကို အမျိုးမျိုးလက်ခံကြပါတယ်။ဒါကြောင့်မို့လို့ တစ်ခြား non-stealth scan တစ်ခုကိုလဲ response အပေါ်အကောင်းဆုံးခွဲခြမ်းစိတ်ဖြာနိုင်ဖို့အ တွက်တွဲဖက်အသုံးပြုဖို့လိုအပ်ပါတယ်။

အောက်မှာဖော်ပြထားတဲ့ ဥပမာမှာ အထက်ပါ scan ၃မျိုးလုံးအတွက် open ဖြစ်နေတဲ့ port အတွက်ရော close ဖြစ်နေတဲ့ port အတွက်ပါပုံနှင့်အတူ trace file ကိုတင်ပြထားပါတယ်။ သတိထားရမှာ တစ်ခုကတော့ TCP flags ရဲ့ bits ပမာဏဟာ scan type တစ်ခုချင်းစီအပေါ်မူ တည်ပြီးခြားနားပါတယ်။

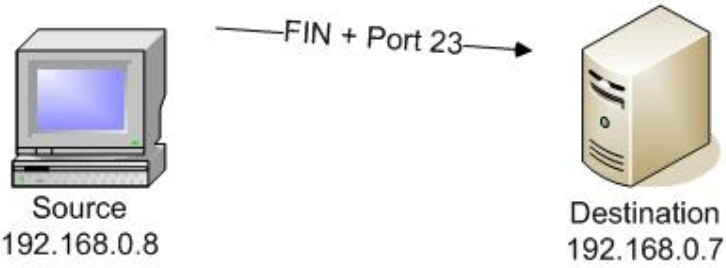
ဒီပုံမှာ FIN scan အတွက် TCP port 443 ဟာပိတ်နေတာတွေ့ရမှာဖြစ်ပါတယ်။ဒါကြောင့် destination 192.168.0.7 ကနေ source 192.168.0.8 ကို RST responce ပြန်ပေးတာ တွေ့ရမှာပါ။



ပုံ ၁

```
 Source          Destination     Summary
-------------------------------------------------------------------------------
[192.168.0.8] [192.168.0.7] TCP: D=443 S=62178 FIN SEQ=3532094343 LEN=0 WIN=2048
[192.168.0.7] [192.168.0.8] TCP: D=62178 S=443 RST ACK=3532094343 WIN=0
```

အကယ်၍ port ဟာ open ဖြစ်ခဲ့ရင် destination ကနေ reply ပြန်လာတာမရှိပါဘူး။



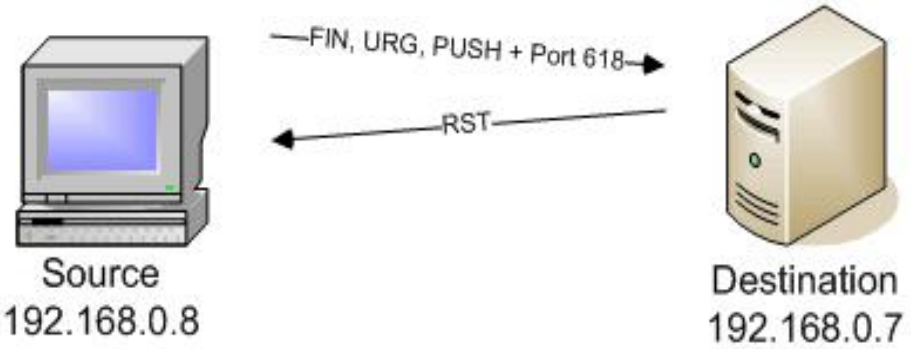
ပုံ ၂

```
 Source          Destination     Summary
-------------------------------------------------------------------------------
[192.168.0.8] [192.168.0.7] TCP: D=23 S=62178 FIN SEQ=3532094343 LEN=0 WIN=2048
```

အောက်မှာဖော်ပြထားတာကတော့ open port အတွင်းရှိ FIN scan ရဲ့ output ကို nmap မှာဖော်ပြထားတာပါ။

```
# nmap -sF -v 192.168.0.7
Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2005-04-23 21:17 EDT
Initiating FIN Scan against 192.168.0.7 [1663 ports] at 21:17
The FIN Scan took 1.51s to scan 1663 total ports.
Host 192.168.0.7 appears to be up ... good.
Interesting ports on 192.168.0.7:
(The 1654 ports scanned but not shown below are in state: closed)
PORT      STATE         SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
79/tcp    open|filtered finger
110/tcp   open|filtered pop3
111/tcp   open|filtered rpcbind
514/tcp   open|filtered shell
886/tcp   open|filtered unknown
2049/tcp open|filtered nfs
MAC Address: 00:03:47:6D:28:D7 (Intel)
Nmap finished: 1 IP address (1 host up) scanned in 2.276 seconds
           Raw packets sent: 1674 (66.9KB) | Rcvd: 1655 (76.1KB)
#
```

အခုတင်ပြမဲ့အကြောင်းအရာကတော့ Xmas Tree Scan (-sX) အကြောင်းပါပဲ။ Xmas tree scan အနေနဲ့ အောက်ဖော်ပြပါအတိုင်း destination ဆီကို TCP frame တွေပို့တဲ့ အခါမှာ URG ၊ PUSH နဲ့ FIN flag တွေကိုပါတွဲဖက်ပေးပို့ပါတယ်။ Xmas tree scan လို့ ဘာကြောင့် ခေါ်သလဲဆိုတော့ Christmas သစ်ပင်မှာ အလှဆင်သော မီးလုံးတွေလို scanner ရဲ့ flag ဟာ byte (00101001) ရောက်တိုင်း တစ်လှည့်စီ အဖွင့်အပိတ် လုပ်ပေးတဲ့ အတွက်ကြောင့်ပါပဲ။ ခုအောက်မှာ ဖော်ပြထားတဲ့ပုံကတော့ request သွားလုပ်တဲ့ port ဟာပိတ်နေတဲ့အတွက် Xmas tree scan ဆီကို RST ပြန်ပို့ပေးတဲ့ပုံပါပဲ။

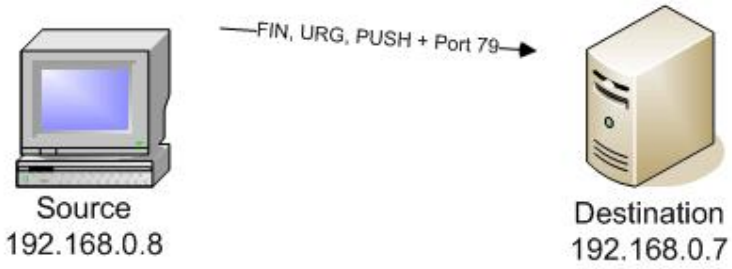


Source Destination Summary

[192.168.0.8] [192.168.0.7] TCP: D=618 S=36793 FIN URG PUSH SEQ=3378228596 LEN=0 WIN=1024

[192.168.0.7] [192.168.0.8] TCP: D=36793 S=618 RST ACK=3378228596 WIN=0

တကယ်လို့ port ဟာ open ဖြစ်နေတယ် ဆိုရင်တော့ FIN scan ကို အသုံးပြုရင် တွေ့ရသလိုမျိုးပဲ reply ပြန်လာတာမရှိတာကိုအောက်ပါပုံအတိုင်းတွေ့ရမှာပါ။



ပုံ ၄

Source Destination Summary

[192.168.0.8] [192.168.0.7] TCP: D=79 S=36793 FIN URG PUSH SEQ=3378228596 LEN=0 WIN=2048

Xmas scan ကနေ ရလာတဲ့ output ဟာလည်း FIN scan မှာကဲ့သို့ပဲ အောက်ပါအတိုင်း တွေ့ရမှာ ဖြစ်ပါတယ်။

nmap -sX -v 192.168.0.7

Starting nmap 3.81 (http://www.insecure.org/nmap/) at 2005-04-23 21:18 EDT

Initiating XMAS Scan against 192.168.0.7 [1663 ports] at 21:18

The XMAS Scan took 1.55s to scan 1663 total ports.

Host 192.168.0.7 appears to be up ... good.

Interesting ports on 192.168.0.7:

(The 1654 ports scanned but not shown below are in state: closed)

PORT STATE SERVICE

21/tcp open|filtered ftp

22/tcp open|filtered ssh

23/tcp open|filtered telnet

79/tcp open|filtered finger

110/tcp open|filtered pop3

111/tcp open|filtered rpcbind

514/tcp open|filtered shell

886/tcp open|filtered unknown

2049/tcp open|filtered nfs

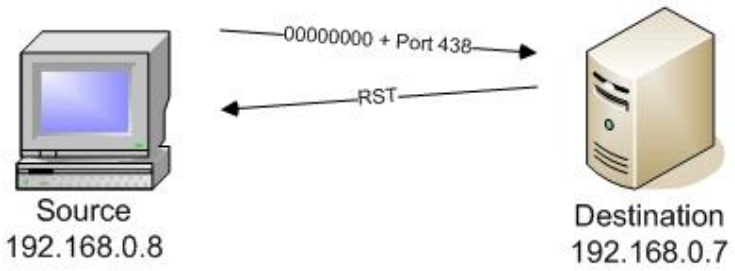
MAC Address: 00:03:47:6D:28:D7 (Intel)

Nmap finished: 1 IP address (1 host up) scanned in 2.432 seconds

Raw packets sent: 1674 (66.9KB) | Rcvd: 1655 (76.1KB) #

နောက်ဆုံးတစ်ခုကတော့ Null scan (-sN) အကြောင်းပါပဲ။ Null scan မှာကျတော့ flag တွေမှာ ဘာတန်ဘိုးမှထည့်မထားပါဘူး။ ဒါကြောင့် flag တွေကို ပိတ်တယ်လို့ဆိုရမှာပါ။ အဲဒီလို flag တန်ဘိုးတွေမထည့်ထားတဲ့အတွက်လက်တွေ့မှာသူ့ရဲ့ flag တွေဟာမမြင်နိုင်ပါဘူး။

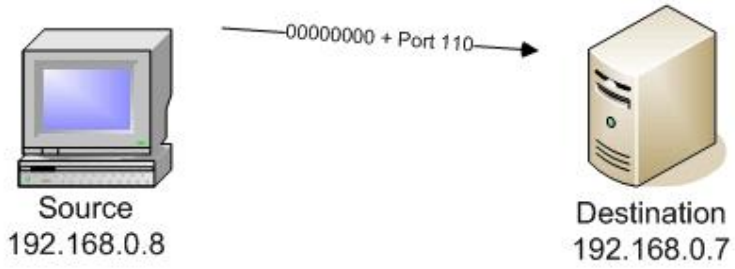
အောက်ဖော်ပြပါပုံမှာ port ဟာပိတ်နေတဲ့အတွက် RST frame return ပြန်လာတာကို တွေ့ရမှာ ဖြစ်ပါတယ်။



ပုံ ၅

```
  Source          Destination     Summary
-------------------------------------------------------------------------------
[192.168.0.8]  [192.168.0.7]  TCP: D=438 S=36860     WIN=4096
[192.168.0.7]  [192.168.0.8]  TCP: D=36860 S=438 RST ACK=2135565682 WIN=0
```

ဒီပုံမှာတော့ port ဟာပွင့်နေတဲ့အတွက် response ပြန်မလာတာကိုတွေ့နိုင်ပါတယ်။



ပုံ ၆

```
  Source          Destination     Summary
-------------------------------------------------------------------------------
[192.168.0.8]  [192.168.0.7]  TCP: D=110 S=36860     WIN=1024
```

Null scan ရဲ့ output ဟာလည်းပဲ FIN scan ၊ Xmas tree scan တို့နဲ့တူတာကို တွေ့ရပါတယ်။

```
# nmap -sN -v 192.168.0.7
Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2005-04-23 21:19 EDT
Initiating NULL Scan against 192.168.0.7 [1663 ports] at 21:19
The NULL Scan took 1.42s to scan 1663 total ports.
Host 192.168.0.7 appears to be up ... good.
Interesting ports on 192.168.0.7:
(The 1654 ports scanned but not shown below are in state: closed)
PORT     STATE         SERVICE
21/tcp   open|filtered ftp
22/tcp   open|filtered ssh
```

```
23/tcp   open|filtered telnet
79/tcp   open|filtered finger
110/tcp  open|filtered pop3
111/tcp  open|filtered rpcbind
514/tcp  open|filtered shell
886/tcp  open|filtered unknown
2049/tcp open|filtered nfs
MAC Address: 00:03:47:6D:28:D7 (Intel)
Nmap finished: 1 IP address (1 host up) scanned in 2.251 seconds
                  Raw packets sent: 1674 (66.9KB) | Rcvd: 1655 (76.1KB)
#
```