APPENDIX 1

DDoS RULES

1. alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDoS TFN Probe"; id: 678; itype: 8; content: "1234"; reference: arachnids, 443; classtype: attempted-recon; sid: 221; rev: 1;)

2. alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDoS tfn2k icmp possible communication"; itype: 0; icmp_id: 0; content: "AAAAAAAAAAAA"; reference: arachnids, 425; classtype: attempted-dos; sid: 222; rev: 1;)

3. alert udp $EXTERNAL_NET any -> $HOME_NET 31335 (msg:"DDoS Trin00\:DaemontoMaster(PONGdetected)"; content: "PONG"; reference: arachnids, 187; classtype: attempted-recon; sid: 223; rev: 1;)

4. alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDoS TFN client command BE"; itype: 0; icmp_id: 456; icmp_seq: 0; reference: arachnids, 184; classtype: attempted-dos; sid: 228; rev: 1;)

5. alert tcp $EXTERNAL_NET any -> $HOME_NET 20432 (msg:"DDoS shaft client to handler"; flow:established; reference: arachnids, 254; classtype: attempted-dos; sid: 230; rev: 2;)

6. alert udp $EXTERNAL_NET any -> $HOME_NET 31335 (msg:"DDoS Trin00\:DaemontoMaster(message detected)"; content: "l44"; reference: arachnids, 186; classtype: attempted-dos; sid: 231; rev: 1;)

7. alert udp $EXTERNAL_NET any -> $HOME_NET 31335
   (msg:"DDoSTrin00\:DaemontoMaster(*HELLO*detected); content:
   sid:232; rev:2;)

8. alert tcp $EXTERNAL_NET any -> $HOME_NET 27665
   (msg:"DDoS Trin00\:Attacker to Master default startup password";
   flow:established,to_server; content:"betaalmostdone"; reference:
arachnids,197; classtype:attempted-dos; sid:233; rev:2;)

9. alert tcp $EXTERNAL_NET any -> $HOME_NET 27665
   (msg:"DDoS Trin00 Attacker to Master default password";
   flow:established,to_server; content:"gOrave"; classtype:attempted-
dos; sid:234; rev:2;)

10. alert tcp $EXTERNAL_NET any -> $HOME_NET 27665
    (msg:"DDoS Trin00 Attacker to Master default mdie password";
    flow:established,to_server; content:"killme"; classtype:bad-unknown;
    sid:235; rev:2;)

11. alert udp $EXTERNAL_NET any -> $HOME_NET 27444
    (msg:"DDoSTrin00\:MastertoDaemon(defaultpassdetected!)"
    content:"l44adsl"; reference:arachnids,197; classtype:attempted-dos;
    sid:237; rev:1;)

12. alert icmp $HOME_NET any -> $EXTERNAL_NET any
    (msg:"DDoS TFN server response"; itype:0; icmp_id:123;
    icmp_seq:0; content: "shell bound to port"; reference:arachnids,182;
    classtype:attempted-dos; sid:238; rev:4;)

13. alert udp $EXTERNAL_NET any -> $HOME_NET 18753
    (msg:"DDoS shaft handler to agent"; content: "alive tijgu"
    reference:arachnids,255; classtype:attempted-dos; sid:239; rev:1;)

14. alert udp $EXTERNAL_NET any -> $HOME_NET 20433
    (msg:"DDoS shaft agent to handler"; content: "alive"
    reference:arachnids,256; classtype:attempted-dos; sid:240; rev:1;)

15. alert tcp $HOME_NET any <> $EXTERNAL_NET any (msg:"DDoS shaft synflood"; flags: S,12; seq: 674711609; reference:arachnids,253; classtype:attempted-dos; sid:241; rev:3;)
16. alert udp $EXTERNAL_NET any -> $HOME_NET 6838 (msg:"DDoS mstream agent to handler"; content: "newserver"; classtype:attempted-dos; sid:243; rev:1;)
18. alert udp $EXTERNAL_NET any -> $HOME_NET 10498 (msg:"DDoS mstream handler ping to agent" ; content:"ping"; reference:cve,CAN-2000-0138; classtype:attempted-dos; sid:245; rev:1;)
19. alert udp $EXTERNAL_NET any -> $HOME_NET 10498 (msg:"DDoS mstream agent pong to handler" ; content: "pong"; classtype:attempted-dos; sid:246; rev:1;)
24. alert icmp $EXTERNAL_NET any -> $HOME_NET any
   (msg:"DDoS - TFN client command LE"; itype: 0; icmp_id: 51201;
    icmp_seq: 0; reference:arachnids,183; classtype:attempted-dos;
    sid:251; rev:1;)

25. alert icmp 3.3.3.3/32 any -> $EXTERNAL_NET any
   (msg:"DDoS Stacheldraht server spoof"; itype: 0; icmp_id: 666;
    reference:arachnids,193; classtype:attempted-dos; sid:224; rev:2;)

26. alert icmp $HOME_NET any -> $EXTERNAL_NET any
   (msg:"DDoS Stacheldraht gag server response"; content: "sicken";
    itype: 0; icmp_id: 669; reference:arachnids,195; classtype:attempted-
    dos; sid:225; rev:3;)

27. alert icmp $HOME_NET any -> $EXTERNAL_NET any
   (msg:"DDoS Stacheldraht server response"; content: "ficken";
    itype: 0; icmp_id: 667; reference:arachnids,191; classtype:attempted-
    dos; sid:226; rev:3;)

28. alert icmp $EXTERNAL_NET any -> $HOME_NET any
   (msg:"DDoS Stacheldraht client spoofworks"; content: "spoofworks";
    itype: 0; icmp_id: 1000; reference:arachnids,192; classtype:attempted-
    dos; sid:227; rev:3;)

29. alert icmp $EXTERNAL_NET any -> $HOME_NET any
   (msg:"DDoS Stacheldraht client check gag"; content: "gesundheit!";
    itype: 0; icmp_id: 668; reference:arachnids,194; classtype:attempted-
    dos; sid:236; rev:3;)

30. alert icmp $EXTERNAL_NET any -> $HOME_NET any
   (msg:"DDoS Stacheldraht client check skillz"; content: "skillz"; itype:
    0; icmp_id: 666; reference:arachnids,190; classtype:attempted-dos;
    sid:229; rev:2;)

31. alert icmp $EXTERNAL_NET any <> $HOME_NET any
   (msg:"DDoS Stacheldraht handler->agent (niggahbitch)";
    content:"niggahbitch"; itype:0; icmp_id:9015; reference:url,staff.washington.edu/dittrich/misc/stacheldraht.analysis;
    classtype:attempted-dos; sid:1854; rev:2;)

32. alert icmp $EXTERNAL_NET any <> $HOME_NET any (msg:"DDoS Stacheldraht agent->handler (skillz)"; content:"skillz"; itype:0; icmp_id:6666; reference:url,staff.washington.edu/ditrich/misc/stacheldraht.analysis; classtype:attempted-dos; sid:1855; rev:2;)

33. alert icmp $EXTERNAL_NET any <> $HOME_NET any (msg:"DDoS Stacheldraht handler->agent (ficken)"; content:"ficken"; itype:0; icmp_id:6667; reference:url,staff.washington.edu/ditrich/misc/stacheldraht.analysis; classtype:attempted-dos; sid:1856; rev:2;)
APPENDIX 2

STACHELDRAHT

Stacheldraht is more stable and sophisticated than Trin00, TFN, TFN2K and mstream DDoS tools. It is capable of launching attacks in ICMP, UDP and TCP protocols.

The attack traffic generated in the testbed is a mixture of TCP SYN flood, ICMP_ECHO flood and UDP flood packets.

A.2.1 STACHELDRAHT SOURCE FILES

DDoS tools can be easily located in the Internet. Each tool contains different files but the file structures are similar among DDoS tools. Users need to identify the files for client (Source attack machine), handler and agents of their chosen tools.

For StacheldrahtV4, the source files are in the main folder “Stacheldrahtv4”. The handler and agent source files reside directly under the “Stacheldrahtv4” main directory.

Two additional folders “leaf” and “telnetc” are created under the main directory. The folder “leaf” contains the source files for agents while the master server client files are stored under “telnetc”.

The main files for master server client, handler and agents are “client.c”, “mserv.c” and “td.c” respectively. These file names constitute the executable filenames after compilation.
### Table A 2.1  Stacheldrahtv4 Files

<table>
<thead>
<tr>
<th>Client</th>
<th>Handler</th>
<th>Agent</th>
<th>Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stacheldrahtv4</td>
<td>Stacheldrahtv4</td>
<td>Stacheldrahtv4</td>
<td>Stacheldrahtv4</td>
</tr>
<tr>
<td>/telnetc/</td>
<td>/</td>
<td>/leaf/</td>
<td></td>
</tr>
<tr>
<td>bf_tab.h</td>
<td>telnetc</td>
<td>bf:tab.h</td>
<td>blowfish.c</td>
</tr>
<tr>
<td>client.c</td>
<td>bf_tab.h</td>
<td>config.h</td>
<td>blowfish.h</td>
</tr>
<tr>
<td>makefile</td>
<td>config.h</td>
<td>config.h.in</td>
<td></td>
</tr>
<tr>
<td></td>
<td>makefile</td>
<td>control.h</td>
<td></td>
</tr>
<tr>
<td>mserv.c</td>
<td>mmserv.c</td>
<td>icmp.c</td>
<td></td>
</tr>
<tr>
<td>tubby.h</td>
<td>tubby.h</td>
<td>makefile</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>syn.c</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>td.c</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>tubby.h</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>udp.c</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Executables</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>client</td>
<td>mserv</td>
<td>td</td>
<td>--</td>
</tr>
</tbody>
</table>

### A.2.2  ATTACK TRAFFIC GENERATION

The Stacheldraht tool can be freely downloaded from the internet and slightly modified for testbed conditions as below:

1. The communication port between client machine (the MASTERSERVERPORT in client.c) and handler machine (MSERVERPORT in mserv.c) were set to the same port number.

2. The IP address of the handler (LOCALIP value in the mserv.c) and handler address of agent (MSERVER1 value in td.c) were set to the same IP address.
3. The communication port between agent and handler (COMMANPORT in both mserv.c and td.c) were set to the same value.

4. A password was encrypted using blowfish.c. The handler password (SALT in mserv.c) was set to this encrypted value and is used for the communication between the agents and handler.

5. The three executables – client for attack source, mserv for handler and td for agent are created and placed in the appropriate machines.

Start the client, handler and agent programs. Check if the handler is running or not. (Command : ps -e | grep mserv). The handler checks the number of alive and dead agents.

Launch attack against target machine using UDP flooding attack (Command : .mudp <target address>), ICMP flooding attack (Command : .micmp <target address>) and TCP SYN attack (Command : .msyn <target address>).

To end attack execute stop command at all the agent machines (Command : .mstop all) and exit Stacheldraht (Command : .quit).