Appendix - TDTS Port Scan Detection

TDTS TCP Port Scan Detection

Definition

Attacker sends many crafted TCP packets (*) to a specified destination IP address from a specified source IP address, and check its response to determine what service is available on the specified destination IP address.

Regarding to the crafted TCP packets, per our definition, they could be:

- A TCP packet with only SYN flag asserted: TCP SYN port scan
- A TCP packet with only FIN flag asserted: TCP FIN port scan
- A TCP packet with none flag asserted: TCP NULL port scan
- A TCP packet with FIN, PSH, URG flags asserted: TCP XMAS port scan
- A TCP packet with other than the above flags asserted: TCP port scan

Prerequisite

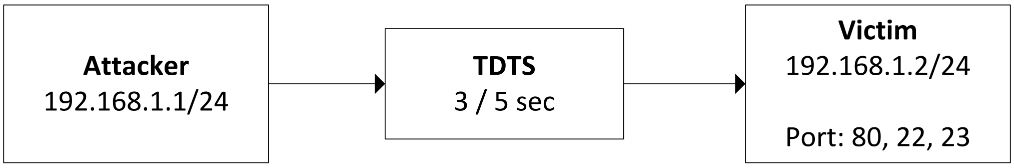
The destination IP address may respond a TCP RST packet for each crafted TCP packet targeting to an unavailable TCP port.

Detection Criteria

During the predefined period of time (i.e. detection period), there are more than predefined amount (i.e. detection threshold) of crafted TCP packets with the same IP pair.

Example

Given the attacker is 192.168.1.1, the victim is 192.168.1.20 and it listens on TCP port 80, 22, 23, the detection period is 5 seconds, and the detection threshold is 3.



Example 1

```
Sec attacker                                         victim
0s TCP SYN:80 ----------------------------->
              <----------------------------- TCP SYN/ACK:80
1s TCP SYN:22 ----------------------------->
              <----------------------------- TCP SYN/ACK:22
2s TCP SYN:81 ----------------------------->
```

```
              <----------------------------- TCP RST:81
   TCP SYN:88 -----------------------------> scan count = 1
   TCP SYN:89 -----------------------------> scan count = 2
   TCP SYN:90 -----------------------------> scan count = 3
              <----------------------------- TCP RST:88
              <----------------------------- TCP RST:89
              <----------------------------- TCP RST:90
   TCP SYN:91 -----------------------------> detected
```

Example 2

```
Sec attacker                              victim
0s TCP SYN:80 ----------------------------->
              <----------------------------- TCP SYN/ACK:80
1s TCP SYN:22 ----------------------------->
              <----------------------------- TCP SYN/ACK:22
2s TCP SYN:81 ----------------------------->
   TCP SYN:88 -----------------------------> scan count = 0
   TCP SYN:89 -----------------------------> scan count = 0
   TCP SYN:90 -----------------------------> scan count = 0
   TCP SYN:91 -----------------------------> no detection
```

Example 3

```
Sec attacker                              victim
0s TCP SYN:81 ----------------------------->
              <----------------------------- TCP RST:81
1s TCP SYN:82 -----------------------------> scan count = 1
              <----------------------------- TCP RST:82
6s TCP SYN:83 -----------------------------> scan count = 2
   TCP SYN:82 -----------------------------> scan count = 2
   TCP SYN:84 -----------------------------> scan count = 3
   TCP SYN:85  -----------------------------> detected
```

Example 4

```
Sec attacker                              victim
0s TCP SYN:81 ----------------------------->
              <----------------------------- TCP RST:81
1s TCP SYN:82 -----------------------------> scan count = 1
              <----------------------------- TCP RST:82
7s  TCP SYN:83 -----------------------------> scan count = 1
    TCP SYN:84 -----------------------------> scan count = 2
    TCP SYN:85 -----------------------------> scan count = 3

    TCP SYN:86 -----------------------------> detected
```

TDTS UDP Port Scan Detection

Definition

Attacker sends many UDP packets to a specified destination IP address from a specified source IP address, and check its response to determine what service is available on the specified destination IP address.

Prerequisite

The destination IP address may respond a ICMP Port Unreachable packet for each UDP packet targeting to an unavailable UDP port.

Detection Criteria

During the predefined period of time (i.e. detection period), there are more than predefined amount (i.e. detection threshold) of UDP packets with the same IP pair.

Appendix - TDTS IP Sweep Detection

Definition

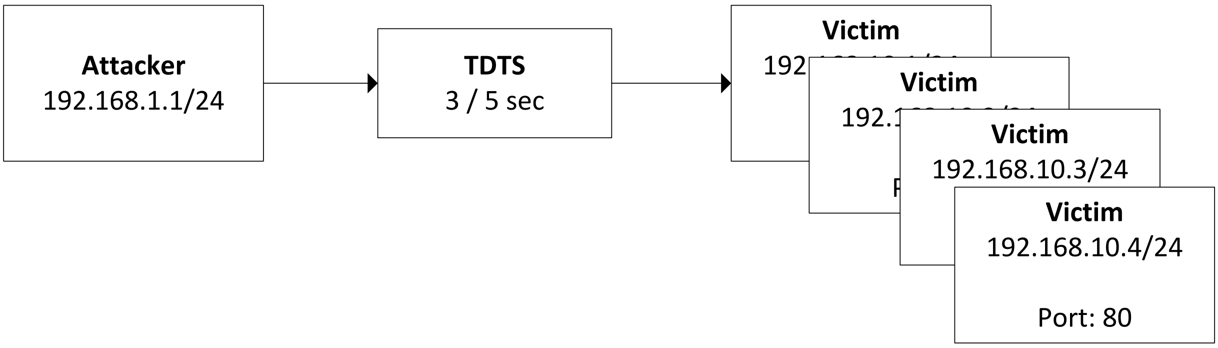
Attacker sends many TCP SYN packets to an IP address range from a specified source IP address, and check its response to determine whether they are available hosts.

Detection Criteria

During the predefined period of time (i.e. detection period), there are more than predefined amount (i.e. detection threshold) of TCP SYN packets from the same IP address to an IP address range.

Example

Given the attacker is 192.168.1.1, the detection period is 5 seconds, and the detection threshold is 3.



Example 1

```
Sec attacker                          victims
0s  TCP SYN     ------------------------------>  192.168.10.1:80 scan count = 1
1s  TCP SYN     ------------------------------>  192.168.10.2:80 scan count = 2
               <------------------------------  TCP SYN/ACK
2s  TCP SYN     ------------------------------>  192.168.10.3:80 scan count = 3
               <------------------------------  TCP RST
    TCP SYN     ------------------------------>  192.168.10.4:80 detected
```

Example 2

```
Sec attacker                          victims
0s  TCP SYN     ------------------------------>  192.168.10.1:80 scan count = 1
1s  TCP SYN     ------------------------------>  192.168.10.2:80 scan count = 2
               <------------------------------  TCP SYN/ACK
6s  TCP SYN     ------------------------------>  192.168.10.3:80 scan count = 3
               <------------------------------  TCP RST
7s  TCP SYN     ------------------------------>  192.168.10.4:80 detected
```

Example 3

```
Sec attacker                                   victims
0s  TCP SYN     -----------------------------> 192.168.10.1:80 scan count = 1
1s  TCP SYN     -----------------------------> 192.168.10.2:80 scan count = 2
               <----------------------------- TCP SYN/ACK
7s  TCP SYN     -----------------------------> 192.168.10.3:80 scan count = 1
               <----------------------------- TCP RST
8s  TCP SYN     -----------------------------> 192.168.10.4:80 scan count = 2
    TCP SYN     -----------------------------> 192.168.10.5:80 scan count = 3
    TCP SYN     -----------------------------> 192.168.10.6:80 detected
```

Example 4

```
Sec attacker                                   victims
0s  TCP SYN     -----------------------------> 192.168.10.1:80 scan count = 1
1s  TCP SYN     -----------------------------> 192.169.10.2:80 scan count = 1
               <----------------------------- TCP SYN/ACK
2s  TCP SYN     -----------------------------> 192.168.10.3:80 scan count = 2
               <----------------------------- TCP RST
    TCP SYN     -----------------------------> 192.168.10.4:80 scan count = 3
3s  TCP SYN     -----------------------------> 192.168.10.5:80 detected
```

Example 5

```
Sec attacker                                   victims
0s  TCP SYN     -----------------------------> 192.168.10.1:80 scan count = 1
1s  TCP SYN     -----------------------------> 192.168.10.2:90 scan count = 1
               <----------------------------- TCP SYN/ACK
2s  TCP SYN     -----------------------------> 192.168.10.3:80 scan count = 2
               <----------------------------- TCP RST
    TCP SYN     -----------------------------> 192.168.10.4:80 scan count = 3
4s  TCP SYN     -----------------------------> 192.168.10.5:80 detected
```

Appendix - TDTS ADP

Definition

Attacker sends one packet with malformed protocol data (i.e. anomaly packet).

Detection Criteria

Any packet with malformed data for supported protocols should be detected.

Detection Criteria of IPv4/IPv6 Anomaly

Type	Description
TDTS_ADP_TYPE_IP_BAD_VER	Invalid IP version number: • eth_type=0x0800 && ip_version != 4 • eth_type=0x86dd && ip_version != 6
TDTS_ADP_TYPE_IP_BAD_LEN	Invalid IP header length: • eth_type=0x0800 && data_len < 20 • eth_type=0x0800 && iphdr->ihl < 5 • eth_type=0x0800 && data_len < (iphdr->ihl * 4) • eth_type=0x0800 && (iphdr->ihl * 4) < iphdr->tot_len • eth_type=0x86dd && data_len < 40
TDTS_ADP_TYPE_IP_TRUNCATED	Invalid IP data length: • eth_type=0x0800 && data_len < iphdr->tot_len • eth_type=0x86dd && data_len < (iphdr->payload_len + 40)
TDTS_ADP_TYPE_IP_OVERSIZE	Invalid IP data length: • eth_type=0x0800 && data_len >= 64 && data_len > iphdr->tot_len • eth_type=0x86dd && data_len >= 64 && data_len > (iphdr->payload_len + 40)
TDTS_ADP_TYPE_IP_BAD_FLAG_UF	Invalid IP fragmentation options or value: • eth_type=0x0800 && (iphdr->frag & 0x8000 != 0)
TDTS_ADP_TYPE_IP_BAD_FLAG_DF_MF	Invalid IP fragmentation options or value: • eth_type=0x0800 && (iphdr->frag & 0x0200) && (iphdr->frag & 0x0400)
TDTS_ADP_TYPE_IP_BAD_OPT	Invalid IP option:

	- eth_type=0x0800 && (opt->len < 2 || opt->len > data_len)

Detection Criteria of TCP Anomaly

Type	Description
TDTS_ADP_TYPE_TCP_BAD_LEN	Invalid TCP header length: • data_len < 20 • data_len < (tcphdr->doff * 4)
TDTS_ADP_TYPE_TCP_BAD_FLAG	Invalid TCP flags: • tcphdr->flag == 0 • tcphdr->syn && tcphdr->fin • tcphdr->ack && tcphdr->rst && (tcphdr->syn || tcphdr->fin) • tcphdr->ack && !(tcphdr->syn || tcphdr->rst) • tcphdr->ack && tcphdr->syn && (tcphdr->urg || tcphdr->psh)
TDTS_ADP_TYPE_TCP_CKSUM	Invalid TCP checksum
TDTS_ADP_TYPE_TCP_WIN	Invalid TCP window size
TDTS_ADP_TYPE_TCP_OVERLAP	Invalid TCP retransmission data
TDTS_ADP_TYPE_TCP_LAND	TCP Landing Attack: • eth_type=0x0800 && iphdr->sip == iphdr->dip && tcphdr->source == tcphdr->dest • eth_type=0x86dd && iphdr->sip == iphdr->dip && tcphdr->source == tcphdr->dest

Detection Criteria of UDP Anomaly

Type	Description
TDTS_ADP_TYPE_UDP_BAD_LEN	Invalid UDP length: • data_len < 8 • data_len != udphdr->len
TDTS_ADP_TYPE_UDP_LAND	UDP Landing Attack: • eth_type=0x0800 && iphdr->sip == iphdr->dip && udphdr->source == udphdr->dest • eth_type=0x86dd && iphdr->sip == iphdr->dip && udphdr->source == udphdr->dest

Detection Criteria of ICMP Anomaly

Type	Description
TDTS_ADP_TYPE_ICMP_BAD_LEN	Invalid ICMP header length: • data_len < 8
TDTS_ADP_TYPE_ICMP_BAD_ERR_MSG	Invalid ICMP error message: • icmph->type == 3 && data_len < 20 • icmph->type == 3 && embed_iphdr->version != 4 • icmph->type == 3 && embed_iphdr->ihl < 5 • icmph->type == 3 && embed_iphdr->saddr != iphdr->daddr

Detection Criteria of ICMPV6 Anomaly

Type	Description
TDTS_ADP_TYPE_ICMP_BAD_LEN	Invalid ICMP header length: • data_len < 8
TDTS_ADP_TYPE_ICMP_BAD_ERR_MSG	Invalid ICMP error message: • icmph->type < 128 && data_len < 48 • icmph->type < 128 && embed_iphdr->version != 6 • icmph->type < 128 && embed_iphdr->saddr != iphdr->daddr

Detection Criteria of IGMP Anomaly

Type	Description
TDTS_ADP_TYPE_IGMP_BAD_LEN	Invalid IGMP header length: • data_len < 8
TDTS_ADP_TYPE_IGMP_BAD_VAL	Invalid IGMP data: • igmph->type == 0x17 && iphdr->dip != 224.0.0.2

Example

In the following IPv4 header, the IP version number is 7.

```
0000  7f 00 00 7c 00 00 40 00  40 01 fd 30 7f 00 00 01   ...|..@. @..0....
0010  7f 00 00 01 86 28 00 00  00 01 01 22 00 01 ae 00   .....(.. ..."....
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 01              ........ ....
```

Appendix - TDTS Flood Detection

TDTS IP/TCP/UDP/ICMP/IGMP Anomaly Flood Detection

Definition

Attacker sends anomaly packets from a specified source IP address, and check for vulnerability or achieve specific attacks.

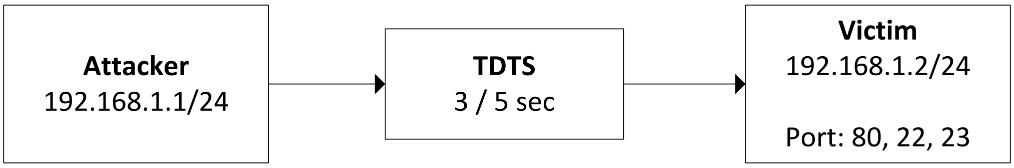
⇒ *Please refer to TDTS ADP section to get more information of anomaly packet detection.*

Detection Critiria

During the predefined period of time (i.e. detection period), there are more than predefined amount (i.e. detection threshold) of anomaly packets from the same source address.

Example

Given the attacker is 192.168.1.1, the victim is 192.168.1.2, the detection period is 5 seconds, and the detection threshold is 3.



```
Sec attacker                               victim
0s UDP Anomaly -----------------------------> scan count = 1
              <-----------------------------
1s UDP Anomaly -----------------------------> scan count = 2
              <-----------------------------
2s UDP Anomaly ------------------------------> scan count = 3
2s UDP Anomaly ------------------------------> detected: UDP Anomaly Flood
```

TDTS TCP SYN Flood Detection

Definition

Attacker attempts a lot of TCP SYN to exhaust TCP connection usage at target victim.

Prerequisite

The destination IP address may respond a TCP RST packet when it's under heavy load.

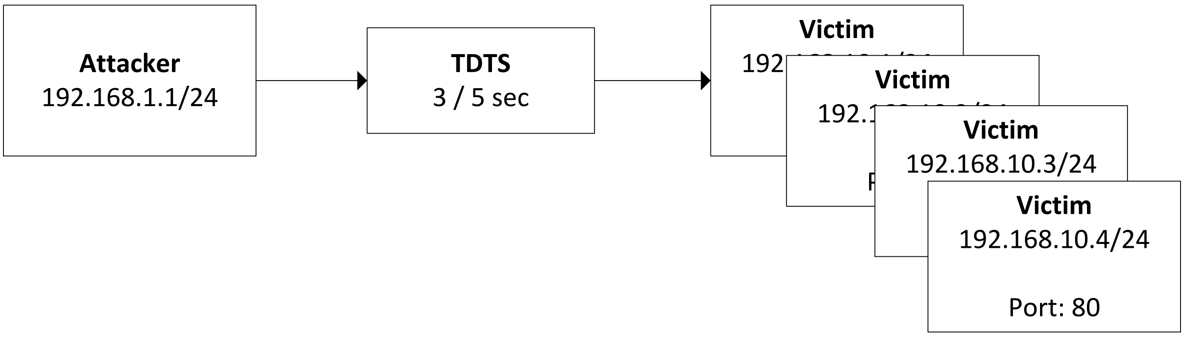
Detection Criteria

- TCP DST SYN Flood: During the predefined period of time (i.e. detection period), there are more than predefined amount (i.e. detection threshold) of TCP RST packets with the same destination IP and TCP port.

- TCP SRC SYN Flood: During the predefined period of time (i.e. detection period), there are more than predefined amount (i.e. detection threshold) of TCP RST packets with same source IP.

Example of TCP SRC SYN Flood

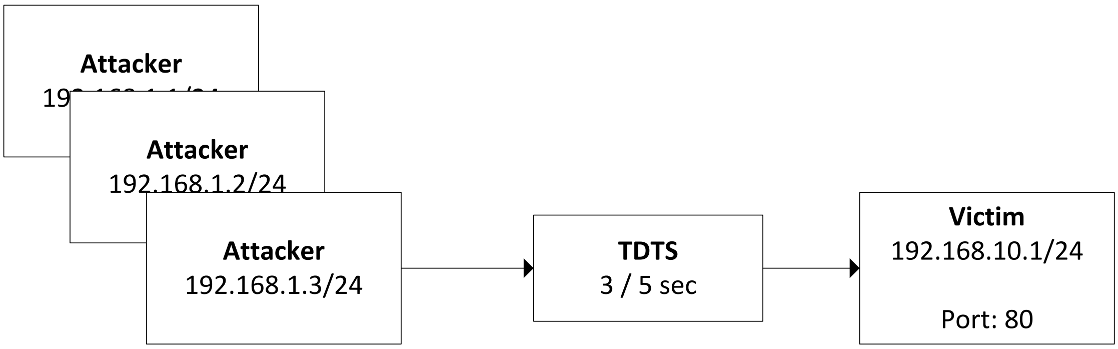
Given the attacker is 192.168.1.1, the detection period is 5 seconds, and the detection threshold is 3.



```
Sec attacker                               victim
0s TCP SYN ----------------------------->
           <----------------------------- SYN ACK
0s TCP SYN ------------------------------>
           <----------------------------- SYN ACK
1s TCP SYN ------------------------------>
           <------------------------------ RST, scan count = 1
1s TCP SYN ------------------------------>
           <------------------------------ RST, scan count = 2
1s TCP SYN ------------------------------>
           <------------------------------ RST, scan count = 3
2s TCP SYN ------------------------------>
           <------------------------------ RST, detected
```

Example of TCP DST SYN Flood

Given the victim is 192.168.10.1, the detection period is 5 seconds, and the detection threshold is 3.



```
Sec attacker                               victim
0s TCP SYN ----------------------------->
```

```
           <----------------------------- SYN ACK
0s TCP SYN ------------------------------>
           <----------------------------- SYN ACK
1s TCP SYN ------------------------------>
           <------------------------------ RST, scan count = 1
1s TCP SYN ------------------------------>
           <------------------------------ RST, scan count = 2
1s TCP SYN ------------------------------>
           <------------------------------ RST, scan count = 3
2s TCP SYN ------------------------------>
           <------------------------------ RST, detected
```