Network Defense Applications Using Stationary and Event-Driven IP Sinkholes

Defeating Denial of Service, Decreasing False Positives, and Enriching Network Intelligence using IP Sinkholes

What this presentation covers

- Sinkhole Background and Function
- Decoy Network Deployment
- Denial of Service Avoidance Strategy
- Backscatter Analysis

VOSTROM

What's an **IP sinkhole**?

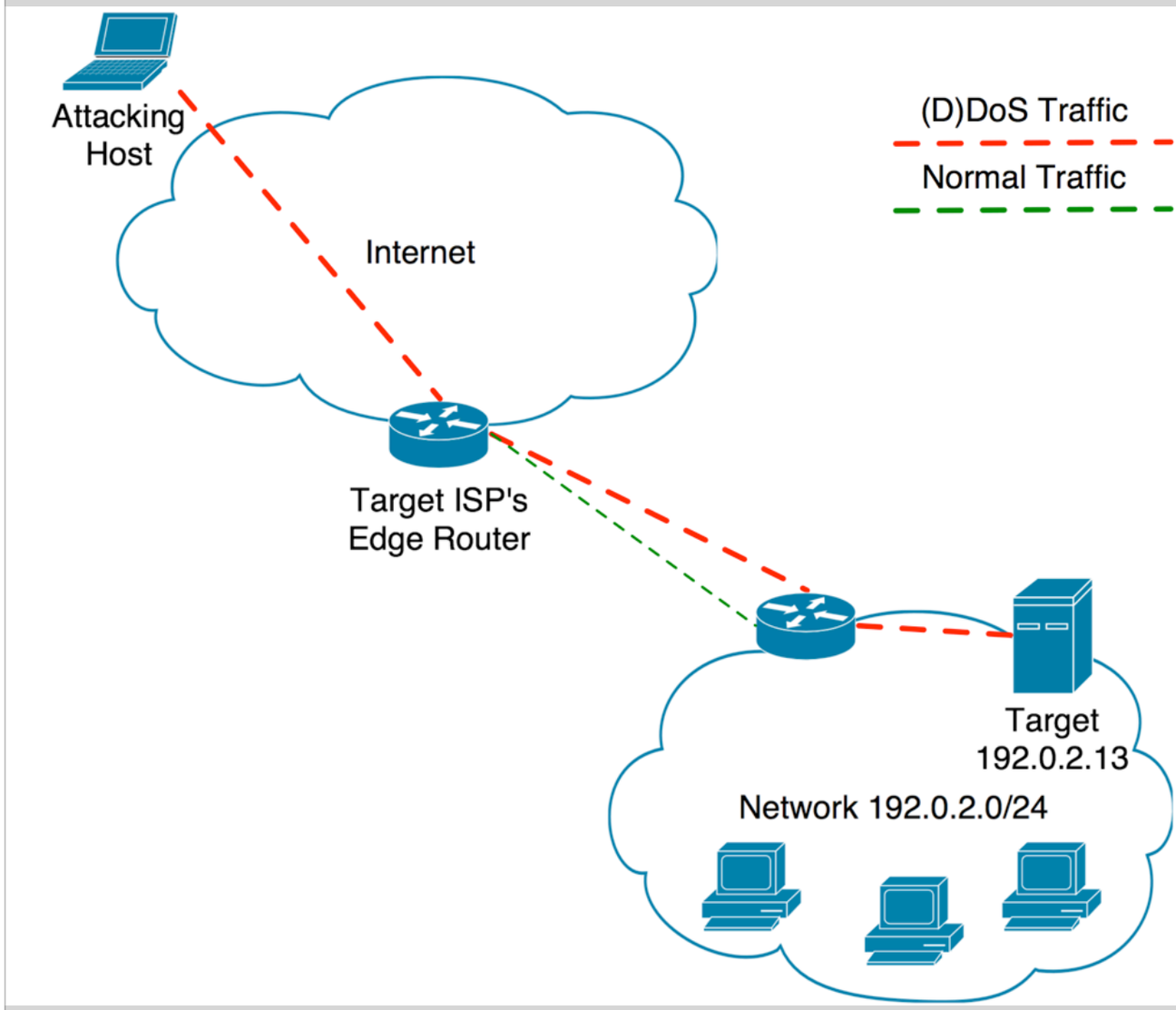
Sinkhole: Definition

- Means of redirecting specific IP network traffic for different security-related purposes including analysis and forensics, diversion of attacks, and detection of anomalous activities

VOSTROM

Sinkhole: Example

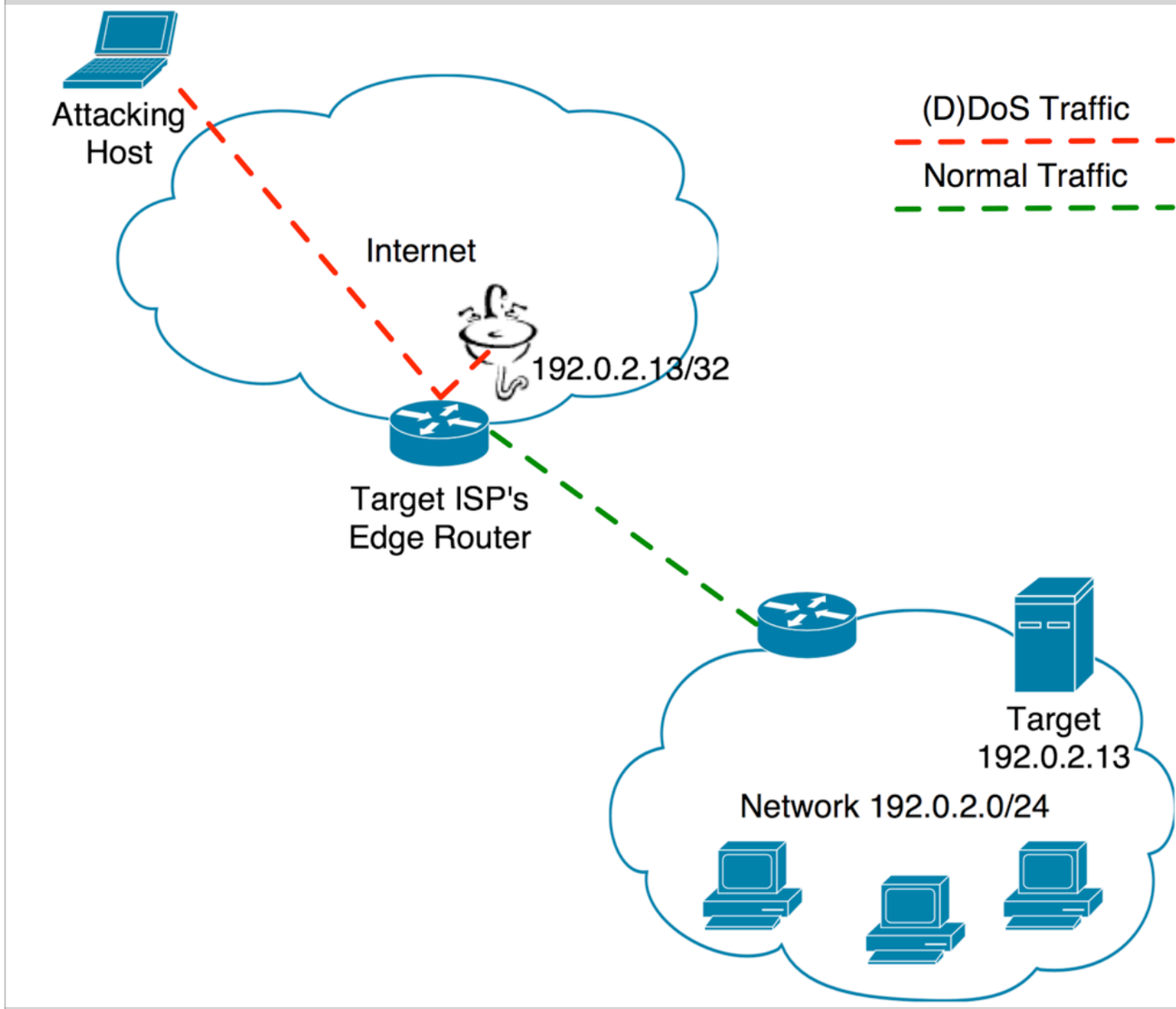
- Lots of DDoS traffic sourced from various networks is destined for host 192.0.2.13
- The organization that owns 192.0.2.13 (or its ISP) announces 192.0.2.0/24 as its network
- The attack becomes debilitating, impacting business operations, etc. The upstream ISP is also impacted, causing problems for adjacent customers. Action must be taken...



Sinkhole: Example

- The ISP reacts and temporarily initiates a blackhole-type sinkhole by injecting a more specific route for the target (192.0.2.13/32) inside their backbone, whose next-hop is the discard interface on their edge router (also known as null0 or the “bit bucket”)

- Result: Organization isn’t flooded anymore and can operate, server 192.0.2.13 cannot talk to outside world, but it works internally and services can be moved to another server (until it gets attacked also/ instead)



That’s one kind of sinkhole

How else can we use sinkholes?

Using sinkholes to deploy **decoy networks**

- Another modern use of sinkholes is in the deployment of various kinds of decoy networks for entrapment, exposure, and intelligence-gathering purposes.

*Decoy \De*coy"\, n. Anything intended to lead into a snare; a lure that deceives and misleads into danger, or into the power of an enemy; a bait.*

VOSTROM

Deploying Decoy Networks

- Two types of decoy networks we'll discuss
  - **darknet**
  - **honeynet**

VOSTROM

What’s a darknet?

Darknet: Definition

- Many definitions ... IRC related, Peer2Peer, etc.

  But in this case....

- **A darknet is an “unlit” or unused region of an existing IP network.** One simple rule: *No services are advertised (DNS, etc) as being available in a darknet and nothing you place in a darknet may respond to requests*

VOSTROM

Only 2 reasons packets fall into a darknet

1. A misconfiguration of some sort (a host with the wrong netmask or a discovery protocol searching the aether, for example)
2. Malware (scanning)

VOSTROM

If there's nothing to talk to, what falls into a darknet?

- Worms and all types of scanning malware don't know there's nothing there, so they fall in
- Systems with incorrect netmasks broadcast packets that land in darknets
- Routers without proper routes for internal traffic redirect to the darknet instead of leaking it upstream (Internet)

VOSTROM

Don’t I need a sophisticated IDS with anomaly detection and statistical analysis to catch zero-day worms and other malware?

Isn’t something like bayesian analysis required to along with patterns/sigs/rules?

No.

1. Modern-day scanners don't necessarily trip even the most sophisticated anomaly detectors.

2. Putting NIDS-capable sensors everywhere in the network can be **expensive** -- especially when there's another way...

You can use a commodity PC and a darknet

VOSTROM

Yes, with a darknet you can detect malware:

- WITHOUT expensive software/hardware to do statistical analysis (anomaly detection) ($$)
- BEFORE waiting for an IDS (Snort, Bro, etc.) or anti-virus signature to come out
- WITHOUT analyzing 100% of your network/traffic ($$)
- WITHOUT false positives (I don’t like that term, but that’s another story)

VOSTROM

But wait, there’s more!

External darknets do even more...

- By redirecting some “outside” Internet addresses (unfiltered) into your darknet, you can detect:
  - Footprinting/Scanning/Probing
  - Backscatter (we’ll discuss this later)
  - You can also perform tracebacks, but we won’t cover those in this presentation

VOSTROM

Why darknets?

- As with any network monitoring solution, there are lots of strategies that do “more” or “less” but with the darknet, think:
  - Cost savings
  - Data reduction
  - Speed of reporting / ease of workflow

VOSTROM

Re-cap the advantages of a darknet

- Detect scanning malware inside our network -- quickly and without false positives
- Detect footprinting/probes from outside our network
- Detect misconfigurations internally
- Find out when we're being implicated in an attack and who's being attacked (backscatter comes later)
- *Glean more network intelligence by reducing noise*

VOSTROM

Great, how do we deploy a darknet?

Darknet Deployment: 5 easy steps

- **First, select one or more regions of unused IP/network space from your internal network (external IPs can be added to the darknet later).**

- Can be /16 down to /32 in size. More = better. From more subnets (more distributed) = even better.

- Routing follows specificity. So, if you use 10.1.0.0/24 - 10.100.0.0/24 internally, make the whole 10.0.0.0/8 into a darknet. As soon as something scans anything in the 10/8 network that isn't specifically routed, wham!

VOSTROM

Darknet Deployment: 5 easy steps

- **Set up the physical topology**
  - Need 1 existing layer-3 device carrying your traffic to forward packets into the darknet (your gateway router)
  - Need 1 system/server/collector box with lots of storage and 2 NICs to act as the packet vacuum/collector (we'll use FreeBSD in our example)
  - Need 1 layer-2 Ethernet switch to connect the devices. A nice switch with span/mirror capability is nice to connect lots of other sensors later on

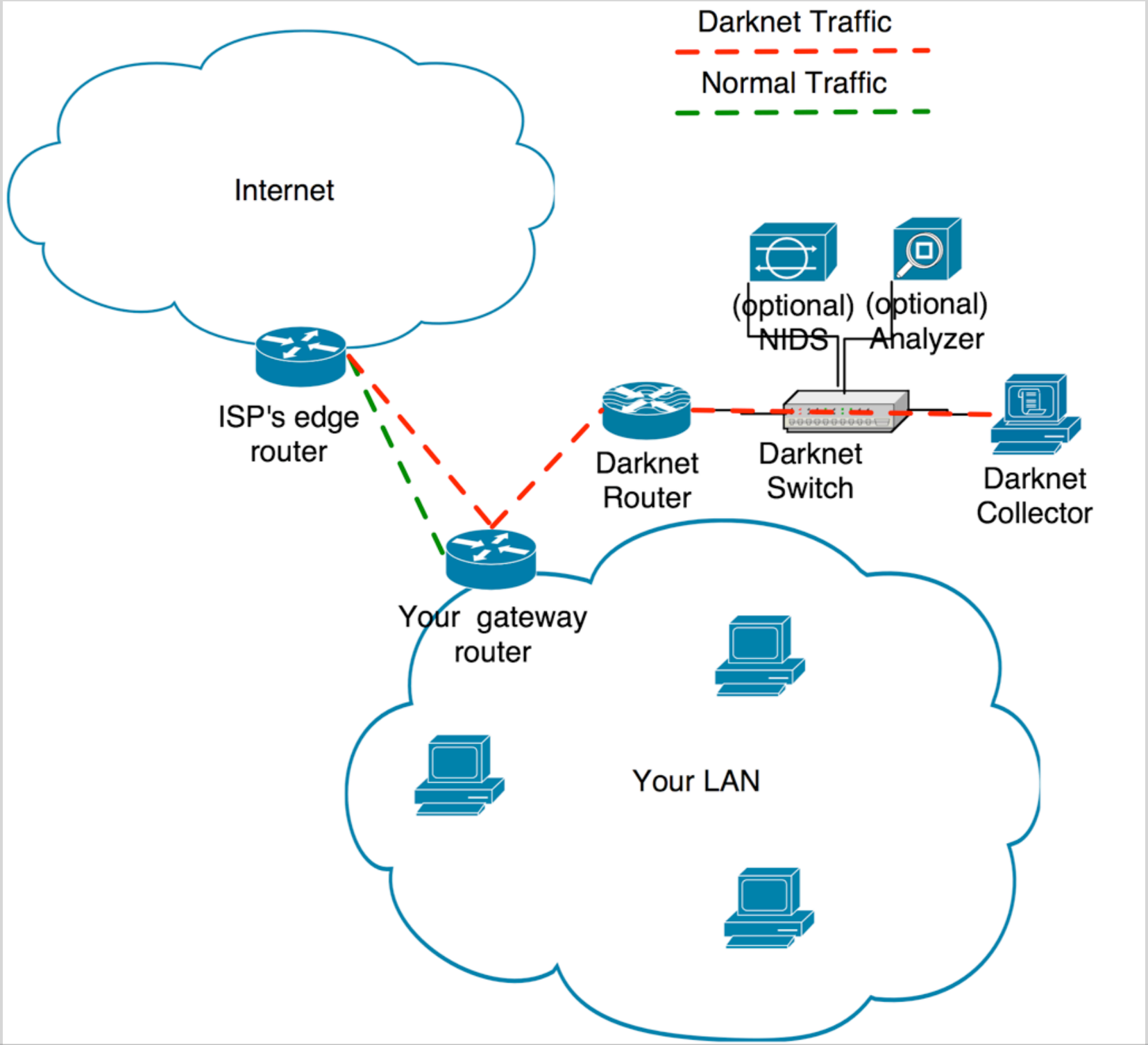
Darknet Deployment: 5 easy steps

- Router can actually be any layer-3 device that internal or external traffic is flowing through, including most firewalls -- in most networks, this is your existing gateway router

- Configure your server to have one “darknet” interface with a made-up IP address (a p2p /30 address whose other side is an interface on the darknet router) and one management interface you can access (ssh, whatever)

- While a darknet can be located within a DMZ, fight the urge to use the DMZ switch as the darknet switch unless you properly VLAN it (we don’t want legitimate broadcast traffic flowing into the darknet)

VOSTROM



Darknet Deployment: 5 easy steps

In our example, we're using a Cisco router running IOS, a FreeBSD-based server, and a commodity layer-2 darknet switch

Darknet Deployment: 5 easy steps

- We don’t want the darknet server to ARP for each darknet IP (there could be thousands), so we need to configure the router to forward the darknet-destined packets to the server’s darknet-interface IP address

- Use a /30 network to create a *point-to-point* between your router and the darknet interface of your server/collector, such as 192.0.2.0/30.

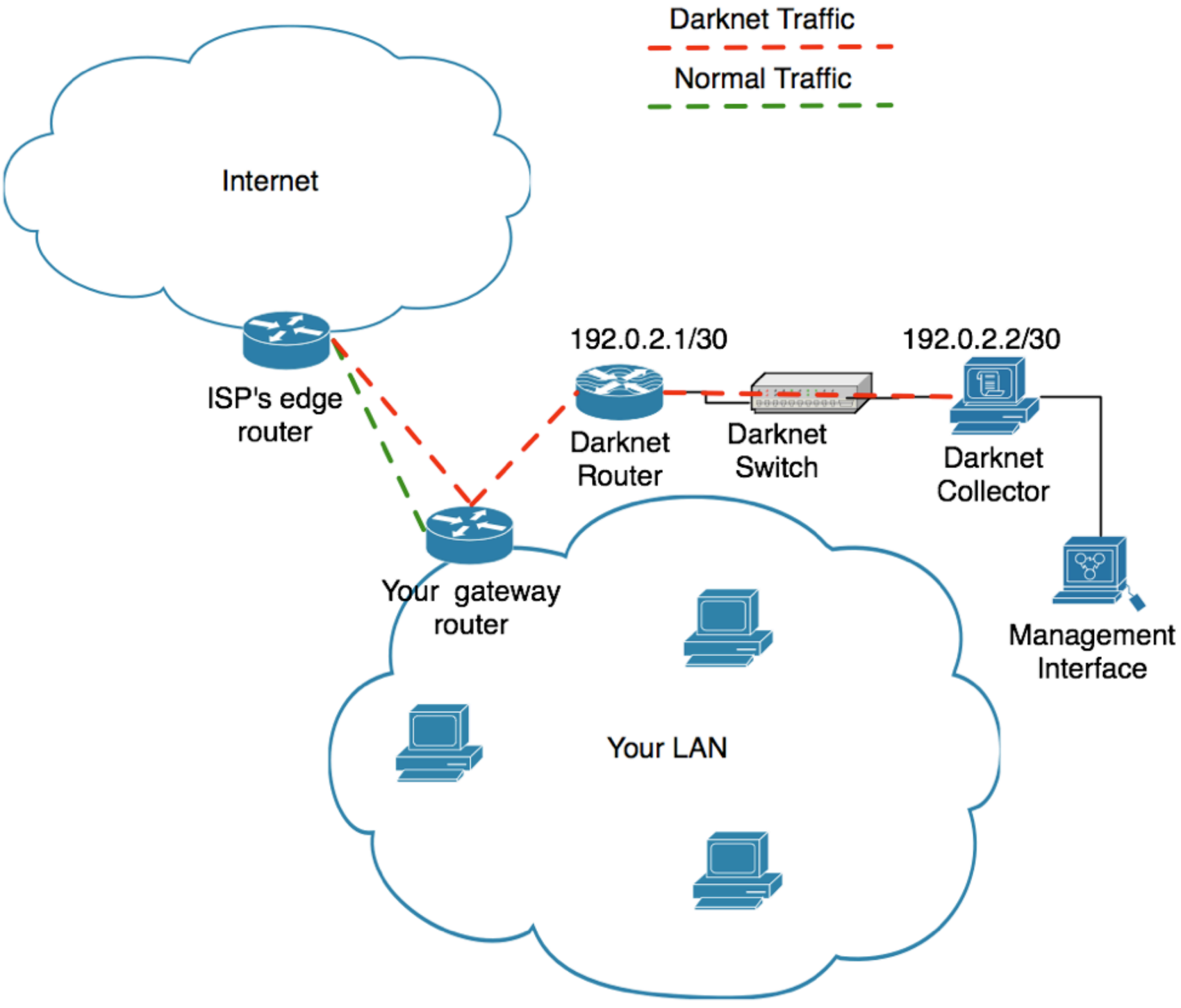
- This would make your router’s Ethernet interface 192.0.2.1/30 and the collector server’s darknet interface would be 192.0.2.2/30

VOSTROM

Darknet Deployment: 5 easy steps

- Once your *point-to-point* network interfaces are set up on the router end and on the server end, enter routing statements into your darknet router to redirect the "darknet IPs" you've chosen from the router toward the darknet server using the /30 *point-to-point*

```
router#conf t
router(config)# ip route 10.0.0.0 255.0.0.0 192.0.2.2
router(config)# ^Z
router# wr
```



Darknet Deployment: 5 easy steps

Your collector or packet vacuum should now be receiving traffic

Darknet Deployment: 5 easy steps

- **Special considerations for the collector server**
  - Never respond on the darknet interface--use a firewall to “deny all”
  - Only ARP for 192.0.2.2/30 (server’s darknet IP)
  - Default gateway should be via the management Ethernet interface, NOT THE DARKNET INTERFACE
  - The firewall you use is up to you... you may get value from the firewall log directly

Darknet Deployment: 5 easy steps

- All firewalls perform differently, use what you know. Some perform well with logging, others don't.

- Packet filter logs will be the easiest way to immediately get value from the darknet as you should almost instantly see packets falling into the darknet and getting filtered by the "deny all" rule of your firewall

- As a safety precaution, null-route the darknet traffic just in case the firewall fails or gets turned off:

```
route add –net 10.0.0.0/8 127.0.0.1 –blackhole
```

Beyond firewall logs, how do you store and analyze the traffic?

Darknet Deployment: 5 easy steps

- **I recommend you store the data so it can be used later with lots of different tools**

- PCAP format is best -- almost every tool operates on it

- **tcpdump/windump** is the most popular way to do that and easily writes pcap-format binary files

```
tcpdump -i en0 -n -w darknet_dump –C125
```

  - Interface selection, no DNS, write files named darknet_dumpN where N makes a unique name for each file containing 125 million bytes, cycled automatically

Darknet Deployment: 5 easy steps

- Once you have PCAPs recorded, you can use your favorite analyzer program to open and analyze them

- I also recommend:

  - ARGUS: **A**udit **R**ecord **G**eneration and **U**tilization **S**ystem developed by QoSient. It provides a keen flow-based summary

  - MRTG (monitor the amount of data flowing through the darknet) -- sometimes it's just good to know there's more darknet activity today than yesterday

VOSTROM

Darknet Deployment: 5 easy steps

- **Items I customarily place in darknets**
  - IDS sensor (Bro, Snort, et al..)
  - A packet sniffer/recorder (Ethereal, tcpdump)
  - A flow analyzer (netflow from darknet router, argus, SiLK)
  - MRTG (generate RRD graphs)
  - p0f to categorize platforms

VOSTROM

What we've already learned

- We know **everything** that enters a darknet is either misconfigured or malicious
- We know we should investigate what falls into the darknet because there is no such thing as a false positive
- Most of us know what regions of our network we're using and what we're not, so we can build darknets easily without disrupting our normal network activity

VOSTROM

Now, how do you get data into darknets from multiple locations without deploying one of them everywhere like NIDS?

Use IGP to distribute it to every router in our network

- Using any interior gateway protocol (EIGRP, OSPF, iBGP, RIP?!?), you can easily “announce” regions of darknet to all of your internal segments, even across backbone links.

***WARNING**: pay special attention to the size of your darknets and the impact on your WAN traffic in the case of geographically distributed implementations*

VOSTROM

Ok, I have my 10/8 darknet.

Are there other interesting networks to darknet?

What about **bogons** and **martians**?

Bogons: Definition

- IP subnets that have not yet been allocated by registration authorities such as ARIN, APNIC, UNLESS THEY ARE BEING ROUTED
- Large networks that are allocated, but are not announced/ routed publicly
- RFC-1918 (private Internets) internal-use-only networks (192.168.0.0/16, 10.0.0.0/8)
- **Any packet destined for a Bogon will not reach its destination, so why is anyone talking to it?**

VOSTROM

Martians: Definition

- Non-unicast IP addresses that shouldn't appear on the Internet under any circumstances
- An address that should never be seen on the Internet, like DHCP auto-configuration addresses or zeroconf or link-local addresses
  - Ex. 169.254.1.0 to 169.254.254.255
- See RFC 1812 (Internet Standard)

VOSTROM

Bogons and Martians belong in your darknet

Where to get lists of Bogons and updates

- http://www.cymru.com/Bogons/
- http://www.completewhois.org/
- Mailing lists
  - http://puck.nether.net/mailman/listinfo/bogon-announce

VOSTROM

Another source for Bogon info

Filter objects from RIPE NCC

fltr-unallocated
The unallocated (by IANA) IPv4 prefixes.
fltr-martian
The reserved and special use IPv4 prefixes.
fltr-bogons
The combination of fltr-unallocated + fltr-martian.

Get them using any whois tool:

whois -h whois.ripe.net <filter-set-name>

They’re also available via DNS zone transfers, etc.

VOSTROM

For example...

- Let's say you have a host with anti-virus, anti-spyware, etc on it. Everything checks out clean.
- You see it in your darknet one day scanning a Bogon or two and/or your internal darknet IP space...
- You nailed it, it's evaded anti-virus. Nuke it.

VOSTROM

Remember... Unlike some IDS implementations,

EVERYTHING that lands in a darknet merits investigation

Now, onto honeynets...

Honeynet: Definition

- Like a darknet, a honeynet is routed IP space, physically segregated from the “real” network
- Instead of providing a destination where packets go to die, a honeynet mimics one or several services in order to establish a two-way dialogue with incoming traffic
- Tightly held, constantly monitored
- Many kinds, all with same goal: learn tactics, extrapolate attack signatures, observe intruders

Types of honeypots

- **Physical**
  - whole machines inside the honeynet with their own IP address, operating system, and service-mimicking tools
- **Virtual**
  - “software-simulated” systems within the honeynet that mimic environmental conditions such as the operating system, network stack, and services provided as decoys. One physical server may provide a network of thousands of virtual honeypots.

How interactive are honeypots?

- **Low-interaction honeypots**
  - Most widely used, only a few packets back and forth
  - Used to extrapolate signatures, model activity, etc.
  - Tar pits (think “LaBrea”) are low-interaction honeypots
- **High-interaction honeypots**
  - Less used, allow complete intrusion

Recommendations for using Honeynets

- They’re a type of sinkhole, so I discuss them
- Significant monitoring responsibility
- Most organizations don’t like to interact with attackers on any level, in this case you’re inviting them
- Extremely valuable for conducting malware research

VOSTROM

How can we use sinkholes to survive DDoS attacks?

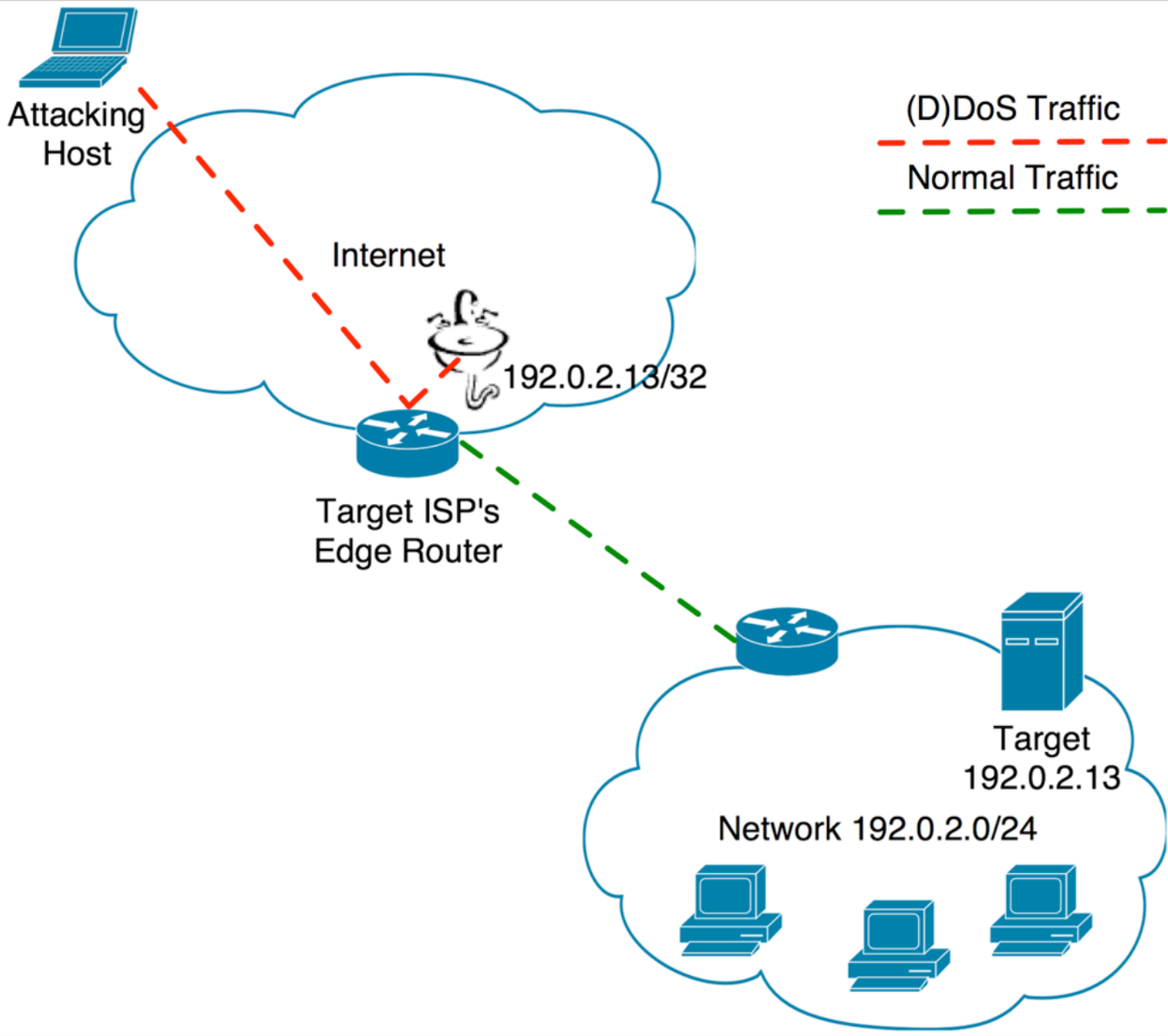
Sinkholes for DDoS Mitigation

- Recall the example from earlier

- ISP implements a sinkhole to redirect traffic destined for the host being attacked into a blackhole in their network

- High-end ISP will allow you to initiate your own “customer-triggered” blackholes to protect yourself.

- You are able to modify their network real-time to mitigate an attack

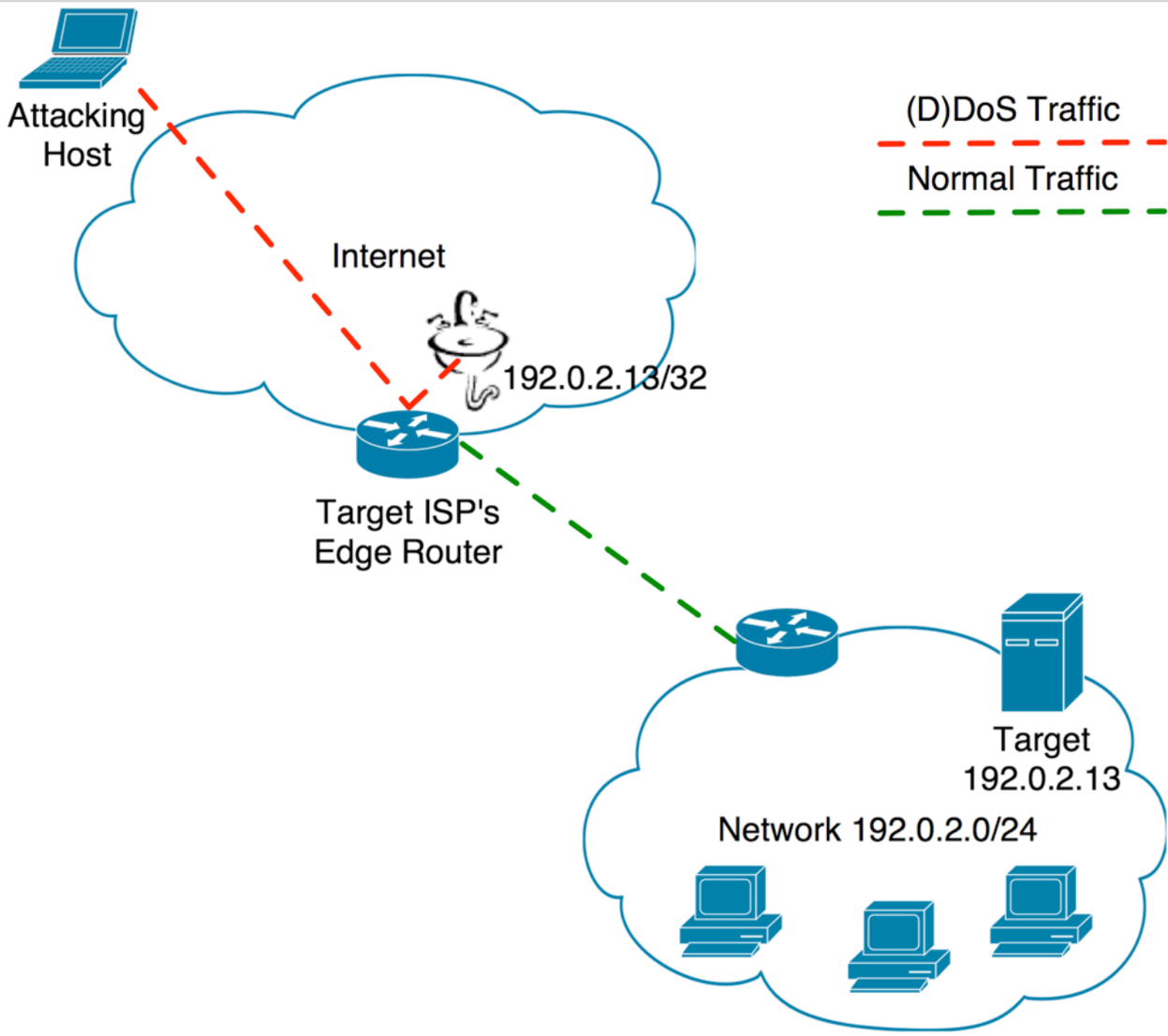


Sinkholes for DDoS Mitigation

- The only difference is that this method doesn't actually protect adjacent ISP customers

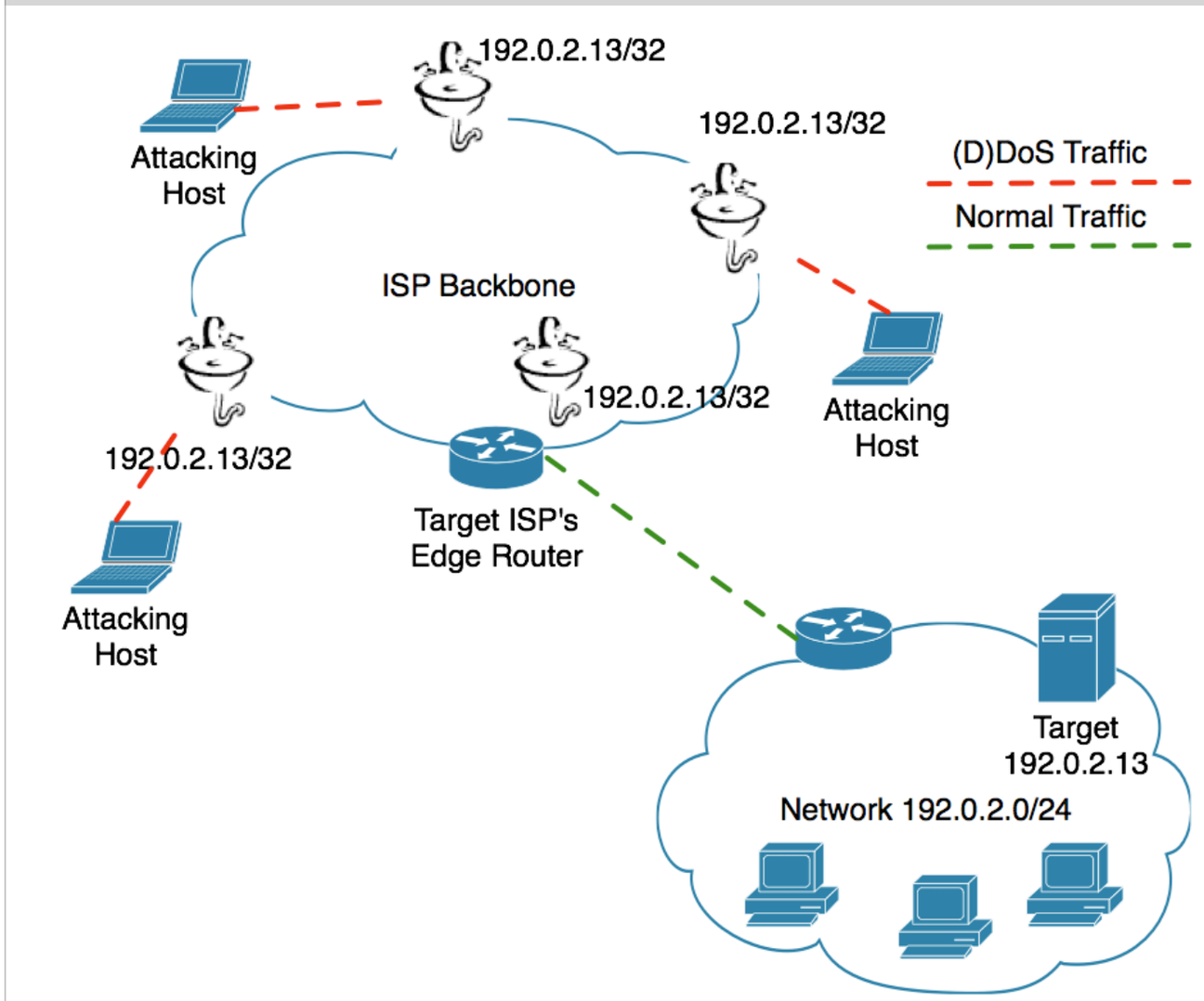
- If attack enters the ISP backbone from multiple sites, the traffic should be dropped at the edge (each site) instead of being dropped in one blackhole before it reaches your circuit

- The "traceback" technique can be used to discover where the attack is entering, but ISPs generally just block it at every edge router for speed of implementation



Sinkholes for DDoS Mitigation

- This is what the modern implementation looks like...



How does the ISP implement this solution?

Triggered Blackhole Routing: ISP Setup

1. Select a non-globally routed prefix, such as the Test-Net (RFC 3330) 192.0.2.0/24, to use as the next hop of any attacked prefix to be blackholed. Using a prefix of length 24 allows you to use many different IP addresses for specific types of blackhole routing. You may wish to differentiate between customer, internal, and external blackhole routes.

2. Configure a static route on each ingress/peering router for 192.0.2.0/24, pointing to the discard interface. For example:

```
ip route 192.0.2.0 255.255.255.0 Null0
```

Triggered Blackhole Routing: ISP Setup

3. Configure BGP and policy route-maps to announce a prefix to be blackholed:

```
router bgp XXX

redistribute static route-map static-to-bgp

# Route-map is a policy mechanism to allow modification of prefix
attributes, or special
# filtering policies
route-map static-to-bgp permit 10
match tag 199
set ip next-hop 192.0.2.1
set local-preference 50
set community no-export
set origin igp
```


Triggered Blackhole Routing: ISP Setup

- In the example configuration, we are redistributing static routes into BGP that match “tag 199” (see below), setting the next hop to an IP address that is routed to the discard interface, setting the local preference to 50 (less preferred), and ensuring we do not leak these routes to any of our external peers (no-export)

- Once this basic configuration is done, the trigger can be initiated by the ISP entering a static route for the attacked prefix (or host) to be blackholed

VOSTROM

Triggered Blackhole Routing: ISP Setup

```
ip route 172.16.0.1 255.255.255.255 192.0.2.1 Null0 tag 199
```

- The static route above is the “trigger” that kicks off the blackhole routing process. The router that this route is configured on will announce the route through iBGP to all internal routers, including edge routers. Any router with a static route to the discard interface for 172.16.0.1/32 will immediately blackhole traffic locally

Now, to allow customers to trigger using eBGP...

Triggered Blackhole Routing: ISP Setup

```
router bgp XXX
# Route-map is simply a policy mechanism to massage routing information such
# as setting the next hop
neighbor < customer-ip > route-map customer-in in
# prefix-list is a static list of customer prefixes and mask length that
# are allowed.  Customer should be allowed to announce down to a single host
# in their prefix(es) such as 172.16.0.1/32
neighbor < customer-ip > prefix-list 10 in
# ebgp-multihop is necessary to prevent continuous prefix announcement and
# withdrawal
neighbor < customer-ip > ebgp-multihop 2
# Now we define the route-map for policy match and setting the blackhole
# next hop
route-map in-customer permit 5
# the customer sets this community on their side, and the ISP matches on its
# side.  XXXX  would likely be the customer ASN, and NNNN is an arbitrary number agreed
# on by the ISP and the customer
match ip community XXXX:NNNN
set ip next-hop < blackhole-ip>
set community additive no-export
```


Triggered Blackhole Routing: ISP Setup

- The ISP already has the < blackhole-ip > statically routed to discard interfaces throughout the network, so as soon as the customer announces the prefix to blackhole, the ISP redistributes that internally and traffic to this prefix is blackholed at the edge of the ISP network

Triggered Blackhole Routing: Customer Setup

- Simple customer router configuration:

```
router bgp XXXX (customer's ASN)
# the customer will install a static route, which is redistributed into BGP
# hereredistribute static route-map static-to-bgp
# just like the ISP, use a route-map to set and match specific prefix
# attributes
route-map static-to-bgp permit 5
# match the arbitrary tag, agreed on by the customer and the ISP
match tag NNNN
set community additive XXX:NNNN
```

Triggered Blackhole Routing: Customer Setup

- Once the BGP configuration is in place, the customer need only install a static route for the prefix being attacked:

```
# NNNN is the tag, agreed on by the customer and the ISP
ip route 192.168.0.1 255.255.255.255 Null0 tag NNNN
```

Triggered Blackhole Routing: Customer Setup

- Remember to negotiate settings with your ISP, all the major ISPs do this a little differently

- With a little up-front configuration, you now have a super-fast mitigation strategy for dealing with DDoS attacks

VOSTROM

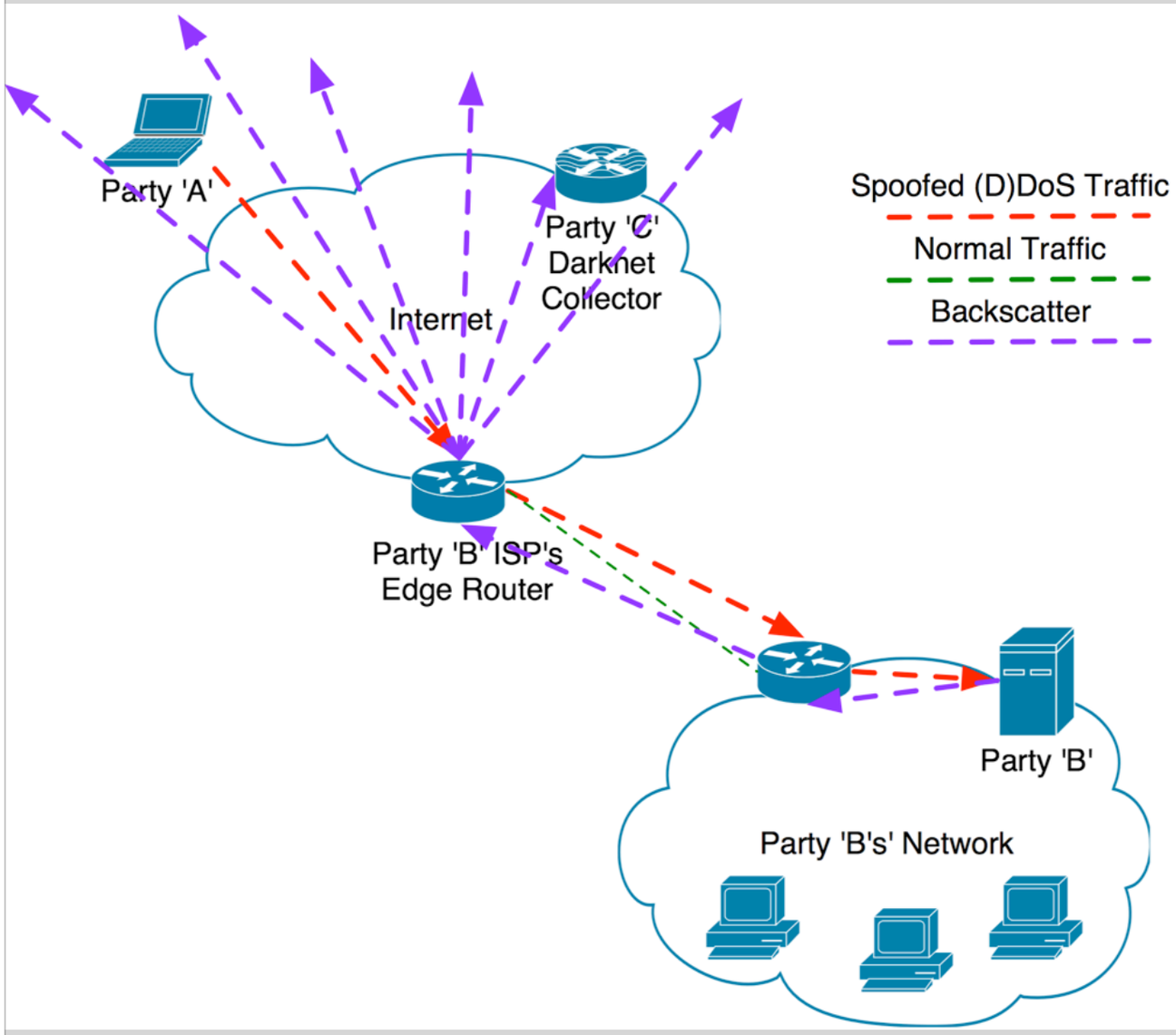
Now, onto analyzing **backscatter**

Backscatter: Definition

- Party **A** attacks party **B** using some kind of DoS flood
- To conceal his identity, party **A** forges his packets to look like he's sourced from **A-Z**
- During the attack, gateways between **A** and **B** inevitably generate messages like "reset" "unreachable" "source quench" and other errors, upset by the DoS attack
- These "error" messages are "returned to sender"

Backscatter: Definition

- Since the sender address is forged, parties **A-Z** all receive the error messages
- Thus, all parties **A-Z** gain knowledge of the attack on **B**
- Most evidence is discarded silently by firewalls since the firewall can tell they are not responses to packets sent by **A-Z**



Backscatter: Usage

- With an external darknet in place, when our IPs are spoofed, we'll receive backscatter
- We'll know when we're being implicated as the source of an attack and who is the target of the attack
- What kinds of packets should we look for?

VOSTROM

Packets to be considered backscatter

Packet	Description
ICMP 3.0	Network Unreachable
ICMP 3.1	Host unreachable
ICMP 3.3	Port unreachable
ICMP 3.4	Fragmentation required
ICMP 3.5	Source route failed
ICMP 3.6	Destination network unknown
ICMP 3.7	Destination host unknown
ICMP 3.10	Host administratively prohibited
ICMP 3.11	Type of service net unreachable
ICMP 3.12	Type of service host unreachable
ICMP 3.13	Comm. admin. prohibited
ICMP 11.0	TTL expired during transit
ICMP 11.1	Fragment reassembly timeout
TCP w/ reset bit set	TCP Reset

Backscatter and Tracebacks

- In large, multi-homed networks, backscatter can be used to perform tracebacks

- When you're suffering a DDoS with spoofed packets, tracebacks will help you determine from which ingress point the attack is sourced

- Not worth explaining since it is mitigated with edge-based triggered blackholes

VOSTROM

This concludes the presentation ... Thank you!

Shout-Out for my Books