Packet Sniffing & Related Forms of Network Surveillance

Steve Revilak
https://masspirates.org

Boston Desktop Linux Users Group

Feb 4th, 2015

What happens when your web browser asks for a page?

Suppose you'd like to look at `http://www.example.com`

- Your browser resolves the name www.example.com to an IP Address (say, 93.184.216.34)
- Your Browser opens a socket to that IP Address, port 80
- Your Browser send a few lines of text, something like

```
GET / HTTP/1.1
Host: www.example.com
```

- The www.example.com web server sends back a response (i.e., the content of a web page).

Let's dig a little deeper

Where's 93.184.216.34?

- Dunno. Somewhere on the net.
- Our computer uses its routing configuration to send the packet.

```
$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Iface
0.0.0.0         192.168.1.1     0.0.0.0         eth0
127.0.0.0       0.0.0.0         255.0.0.0       lo
192.168.1.0     0.0.0.0         255.255.255.0   eth0
```

- My computer sends the request to a router, which sends it to another router. Eventually we reach the destination.
- We get there by "passing the buck".

Directions to 92.184.216.34

```
$ traceroute 93.184.216.34
traceroute to 93.184.216.34 (93.184.216.34)
30 hops max, 60 byte packets
 1 fw.local (192.168.1.1)
 2 10.16.0.1 (10.16.0.1)
 3 10.65.92.101 (10.65.92.101)
 4 tge0-0-0-0.core1.sbo.ma.rcn.net (207.172.15.131)
 5 tge0-1-0-5.core2.nyw.ny.rcn.net (207.172.19.62)
 6 bdle3.border2.nyw.ny.rcn.net (207.172.15.85)
 7 core1.lga.edgecastcdn.net (198.32.118.202)
 8 192.16.18.89 (192.16.18.89)
   93.184.216.34 (93.184.216.34)
```

Our route involved 8 intermediaries.

Stuff we've learned so far

- TCP/IP *packets* include a destination address and port number (so the packet gets there)
- TCP/IP packets include a source address and port number (so a the receiver can send a response, or retransmission requests)
- TCP/IP packets can pass through many intermedaries (MaxMind claims that 93.184.216.34 is in Norwell, MA. We apparently got there by way of NY)
- Packets contain other data: namely, the content of the request you send, and the content of any response you receive.

Let's play capture the packet!

Anyone who handles a packet has the opportunity to examine and/or store it. So let's do this (I'll capture, store, and examine some of my own packets).

Some options

- `sudo tcpdump -s 65535 -l -A dst port 80`
- `sudo tshark -O http -i eth0 \`
  `-R "http.request || http.response"`
- wireshark

tcpdump is a raw ascii capture. tshark is a decoded ascii capture. wireshark is a graphical interface layered on top of tshark.

Questions for the Audience

- If you're on a wireless network, what's your gateway address?
- How many people in this room have the same gateway IP?
- While in this room, what's your publicly visible IP?
- Suppose you captured all of the traffic from that public IP – what would it look like?
- Could you sort the traffic out into per-user streams? How?
- If you can observe traffic, can you modify it?

MITM = observation++

```
$ telnet smtp.gmail.com 587
Trying 74.125.29.109...
Connected to smtp.gmail.com.
Escape character is '^]'.
220 mx.google.com ESMTP h9sm2623161qaq.48 - gsmtp
EHLO fw.local
250-mx.google.com at your service, [209.6.11.2]
250-SIZE 35882577
250-8BITMIME
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-CHUNKING
250 SMTPUTF8
```

What if we removed the 250-STARTTLS line?

Known (commercial) MITM attacks

- Golden Frog caught one wireless broadband provider 'clobbering' STARTTLS
  `https://www.techdirt.com/blog/netneutrality/articles/20141012/06344928801/`
- Verizon's UIDH Header
  `https://www.verizonwireless.com/support/unique-identifier-header-faqs/`
- Xfinity Wireless
  `http://arstechnica.com/tech-policy/2014/09/why-comcasts-javascript-ad-injections-threaten-security-net-neutrality/`

Stuff that Super H8x0rs (and government intelligence agencies) can do

- DNS spoofing. Resolve www.example.com to some other IP address (e.g., a proxy that they control)
- Routing attacks. Divert packets through a network they control.
- Physical tapping (e.g., Project Tempora, or a Raspberry Pi taped underneath a desk)

This is where encryption comes in

- Encryption can prevent a third party from seeing the content of your internet traffic.
- Encryption can prevent a third-party from modifying the content of your internet traffic.
- Some schemes (e.g., STARTTLS, PGP) provide assurance that the other party is who you think they are.

It's time to do away with the old (insecure) internet.

For communications security, the goal isn't *perfect*; the goal is simply *better*.