Computer Security (COM-301)
Monday Live Exercises
TCP, TLS, DoS, and others

Carmela Troncoso
SPRING Lab
carmela.troncoso@epfl.ch

Firewall as a solution

a) A properly configured firewall can prevent any DDoS attack from disrupting the ability of remote users to access your network.

a) A firewall protects your communication from eavesdroppers on the intranet

a) If you have a VPN, you do not need a firewall to protect your web server

Firewall as a solution

a) A properly configured firewall can prevent any DDoS attack from disrupting the ability of remote users to access your network.

a) A firewall protects your communication from eavesdroppers on the intranet

a) If you have a VPN, you do not need a firewall to protect your web server

Possible answer

A) **False**. A firewall cannot prevent **any** DDoS attack. It can only prevent attacks in so far they break the filtering policy.
B) **False**. the firewall is at the exit/entry of the intranet, so it does not prevent anything inside the intranet
C) **False**. VPNs and Firewalls protect from different attacks. Ideally you need both.

The great cannon

In 2015, Github experienced a DoS attack orchestrated by China using the so-called “Great Cannon” (GC). It worked as follows. (Some details of the attack have been simplified or modified for this problem.)

Many websites include a fetch for a script for analytics from Baidu, a large Internet service in China somewhat similar to Google. The script would be retrieved via `http://hm.baidu.com/h.js`. The GC operated in-path at the border between China and the rest of the Internet. Upon seeing a request for this script, the GC would prevent the original HTTP request from being forwarded, and would instead return a different script, which instructed clients to repeatedly load `http://github.com/cn-nytimes`. You can assume that Baidu served its traffic using servers in China; Github did so from servers in the USA; and websites using the analytics script were hosted all over the world.

Whose traffic contributed to the DDOS attack?
a) Web browsers inside China
b) Web browsers outside China
c) Both of these
d) Neither of these

The great cannon

In 2015, Github experienced a DoS attack orchestrated by China using the so-called “Great Cannon” (GC). It worked as follows. (Some details of the attack have been simplified or modified for this problem.)

Many websites include a fetch for a script for analytics from Baidu, a large Internet service in China somewhat similar to Google. The script would be retrieved via `http://hm.baidu.com/h.js`. The GC operated in-path at the border between China and the rest of the Internet. Upon seeing a request for this script, the GC would prevent the original HTTP request from being forwarded, and would instead return a different script, which instructed clients to repeatedly load `http://github.com/cn-nytimes`. You can assume that Baidu served its traffic using servers in China; Github did so from servers in the USA; and websites using the analytics script were hosted all over the world.

Whose traffic contributed to the DDOS attack?
a) Web browsers inside China
b) Web browsers outside China
c) Both of these
d) Neither of these

Possible answer

a) **False.** Browsers inside China will go directly to Baidu and not pass by the GC
b) **True.** Browser outside China are the ones receiving the intervened script and connect to GitHub
c) **False**
d) **False**

The great cannon

In 2015, Github experienced a DoS attack orchestrated by China using the so-called “Great Cannon” (GC). It worked as follows. (Some details of the attack have been simplified or modified for this problem.)

Many websites include a fetch for a script for analytics from Baidu, a large Internet service in China somewhat similar to Google. The script would be retrieved via `http://hm.baidu.com/h.js`. The GC operated in-path at the border between China and the rest of the Internet. Upon seeing a request for this script, the GC would prevent the original HTTP request from being forwarded, and would instead return a different script, which instructed clients to repeatedly load `http://github.com/cn-nytimes`. You can assume that Baidu served its traffic using servers in China; Github did so from servers in the USA; and websites using the analytics script were hosted all over the world.

Which packets would the implementers of this attack need to inspect?
a) Outgoing packets from China
b) Incoming packets to China
c) Both of these
d) Neither of these

The great cannon

In 2015, Github experienced a DoS attack orchestrated by China using the so-called “Great Cannon” (GC). It worked as follows. (Some details of the attack have been simplified or modified for this problem.)

Many websites include a fetch for a script for analytics from Baidu, a large Internet service in China somewhat similar to Google. The script would be retrieved via `http://hm.baidu.com/h.js`. The GC operated in-path at the border between China and the rest of the Internet. Upon seeing a request for this script, the GC would prevent the original HTTP request from being forwarded, and would instead return a different script, which instructed clients to repeatedly load `http://github.com/cn-nytimes`. You can assume that Baidu served its traffic using servers in China; Github did so from servers in the USA; and websites using the analytics script were hosted all over the world.

Which packets would the implementers of this attack need to inspect?

a) Outgoing packets from China
b) Incoming packets to China
c) Both of these
d) Neither of these

Possible answer

a) **False.** Packets leaving China will not be visiting baidu.com
b) **True.** Those are the packets in which the GC can find requests to Baidu scripts and thus to whom send the malicious javascript to connect to GitHub
c) **False**
d) **False**

The great cannon

In 2015, Github experienced a DoS attack orchestrated by China using the so-called “Great Cannon” (GC). It worked as follows. (Some details of the attack have been simplified or modified for this problem.)

Many websites include a fetch for a script for analytics from Baidu, a large Internet service in China somewhat similar to Google. The script would be retrieved via `http://hm.baidu.com/h.js`. The GC operated in-path at the border between China and the rest of the Internet. Upon seeing a request for this script, the GC would prevent the original HTTP request from being forwarded, and would instead return a different script, which instructed clients to repeatedly load `http://github.com/cn-nytimes`. You can assume that Baidu served its traffic using servers in China; Github did so from servers in the USA; and websites using the analytics script were hosted all over the world.

This attack occurred for sets of HTTP requests. Which of the following changes would have prevented the attack? For each choice, assume that the content that the site serves remains the same.

a) Every website that uses Baidu’s analytics switches to serve its content using HTTPS instead of HTTP
b) Github’s server redirects any incoming HTTP connection to a corresponding HTTPS URL
c) Baidu switches its analytics server over to only be accessible using an HTTPS URL
d) None of these

The great cannon

In 2015, Github experienced a DoS attack orchestrated by China using the so-called “Great Cannon” (GC). It worked as follows. (Some details of the attack have been simplified or modified for this problem.)

Many websites include a fetch for a script for analytics from Baidu, a large Internet service in China somewhat similar to Google. The script would be retrieved via `http://hm.baidu.com/h.js`. The GC operated in-path at the border between China and the rest of the Internet. Upon seeing a request for this script, the GC would

Possible answer

a) **False.** The connection between client’s browsers and the web server that servers the page that has the Baidu’s script is not inspected/used by the GC

b) **False.** Redirecting connections does not eliminate the connections. It still consumes resources

c) **True.** This would help, as the GC cannot inspect the connection not inject content

d) **False.**

This attack occurred for sets of HTTP requests. Which of the following changes would have prevented the attack? For each choice, assume that the content that the site serves remains the same.

a) Every website that uses Baidu’s analytics switches to serve its content using HTTPS instead of HTTP
b) Github’s server redirects any incoming HTTP connection to a corresponding HTTPS URL
c) Baidu switches its analytics server over to only be accessible using an HTTPS URL
d) None of these

The great cannon

In 2015, Github experienced a DoS attack orchestrated by China using the so-called “Great Cannon” (GC). It worked as follows. (Some details of the attack have been simplified or modified for this problem.)

Many websites include a fetch for a script for analytics from Baidu, a large Internet service in China somewhat similar to Google. The script would be retrieved via `http://hm.baidu.com/h.js`. The GC operated in-path at the border between China and the rest of the Internet. Upon seeing a request for this script, the GC would prevent the original HTTP request from being forwarded, and would instead return a different script, which instructed clients to repeatedly load `http://github.com/cn-nytimes`. You can assume that Baidu served its traffic using servers in China; Github did so from servers in the USA; and websites using the analytics script were hosted all over the world.

Which of the following techniques could Github have used to make the attack ineffective?

a) Blacklist any packets from Chinese IP addresses
b) Use SYN cookies for all new connections
c) Move the affected Github server to a new IP address
d) Remove all use of Baidu analytics from Github web pages
e) None of these

The great cannon

In 2015, Github experienced a DoS attack orchestrated by China using the so-called “Great Cannon” (GC). It worked as follows. (Some details of the attack have been simplified or modified for this problem.)

Possible answer

a) **False.** The IPs used in this attack are not Chinese (see the first question of this batch)
b) **False.** Connections are finished! SYN cookies will not help
c) **False.** The machines tricked into doing the attack would resolve the new IP (note that they receive a domain, not an IP)
d) **False.** It is not github pages the ones that trigger the attack
e) **True.**

Which of the following techniques could Github have used to make the attack ineffective?

a) Blacklist any packets from Chinese IP addresses
b) Use SYN cookies for all new connections
c) Move the affected Github server to a new IP address
d) Remove all use of Baidu analytics from Github web pages
e) None of these

Ricco’s cafe

Part I. Rita has heard the best coffee in town is served at Ricco's Cafe. She gets there and while she waits for her coffee she wants to watch the new TikTok hits. It turns out that the WiFi network at Ricco's Cafe has no encryption. Ricco warns Rita that it is not safe to use this connection, but Rita disagrees.

Rita connects to the WiFi, and tests that she has Internet connectivity by visiting https://cutestkittens.com. It loads without issues. Rita says to Ricco: “See, no problem! That access was totally safe!” If Rita is correct and the access to cutestkittens.com was safe, explain why she is correct. If she is not correct, provide a network attack against Rita.

Ricco’s cafe

Part I. Rita has heard the best coffee in town is served at Ricco's Cafe. She gets there and while she waits for her coffee she wants to watch the new TikTok hits. It turns out that the WiFi network at Ricco's Cafe has no encryption. Ricco warns Rita that it is not safe to use this connection, but Rita disagrees.

Rita connects to the WiFi, and tests that she has Internet connectivity by visiting https://cutestkittens.com. It loads without issues. Rita says to Ricco: “See, no problem! That access was totally safe!” If Rita is correct and the access to cutestkittens.com was safe, explain why she is correct. If she is not correct, provide a network attack against Rita.

Possible answer
Yes Rita is correct. Because she is connecting to cutestkittens.com using TLS (notice the URL is https://), this means that her client checked the certificate of the website and accepted it. This guarantees the connection is with the cutestkittens.com server even if the WiFi is not encrypted [in fact, WiFi encryption can’t provide any guarantees about the destination]

Ricco’s cafe

Part II. Now that she has tested her WiFi access, Rita decides to have the only muffin sold in the cafe. She does not remember whether she has enough money so she tells Ricco: “Let me check if I have enough money in my bank account.” and starts typing https://QuiteSecBan... on her phone's browser. The next client in line, Randy, also wants the muffin, so he decides to stop Rita from buying it and wants to prevent her from checking her bank account.

Describe a network attack that Randy can do to prevent Rita from checking whether she has enough money in her account. Describe clearly (in one or two sentences) how Randy performs this attack and what capabilities he needs to have to perform the attack.

Ricco’s cafe

Part II. Now that she has tested her WiFi access, Rita decides to have the only muffin sold in the cafe. She does not remember whether she has enough money so she tells Ricco: “Let me check if I have enough money in my bank account.” and starts typing https://QuiteSecBan... on her phone's browser. The next client in line, Randy, also wants the muffin, so he decides to stop Rita from buying it and wants to prevent her from checking her bank account.

Describe a network attack that Randy can do to prevent Rita from checking whether she has enough money in her account. Describe clearly (in one or two sentences) how Randy performs this attack and what capabilities he needs to have to perform the attack.

Possible answers
DNS hijacking – to avoid that Rita receives the bank’s IP
TCP RST – to MitM the connection and stop Rita
DoS on the wifi (or Rita).
ARP poisoning : as Rita is already connected, you need to specify the ARP response is unsolicited (Gratuitous ARP)

And…
DNS poisoning would not work: No time for the process