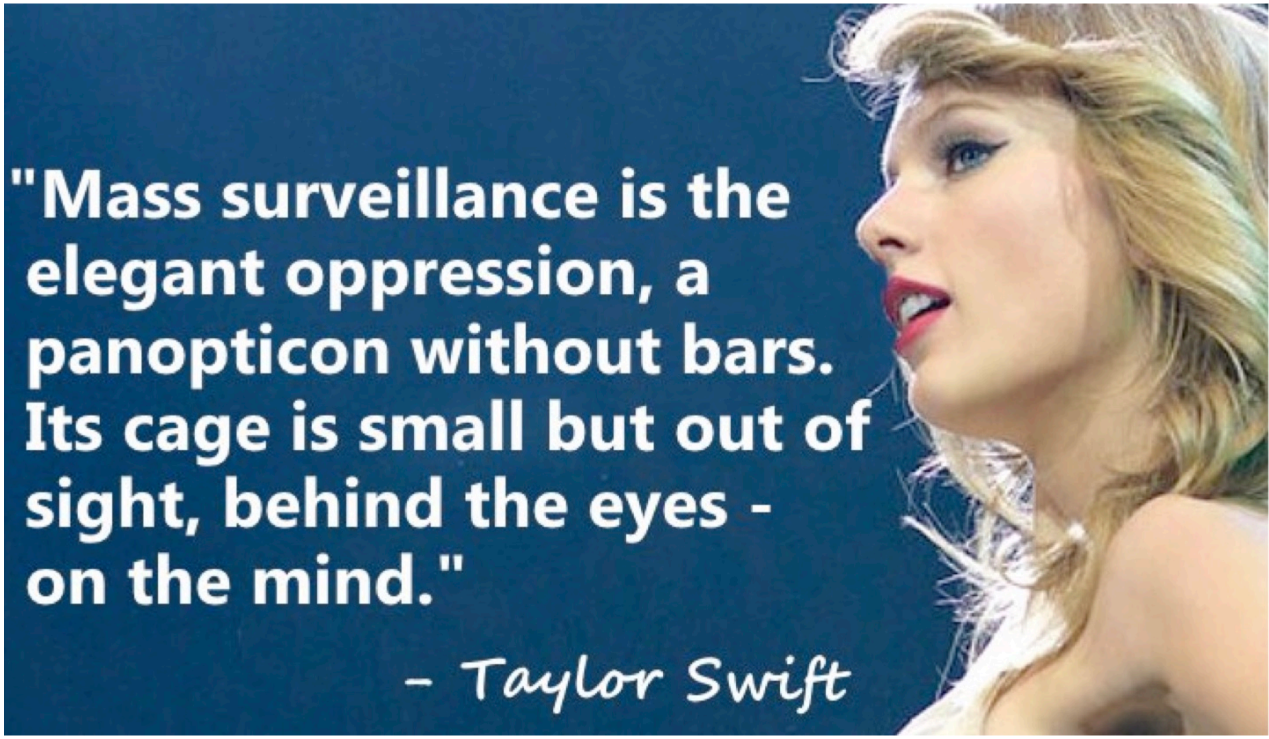
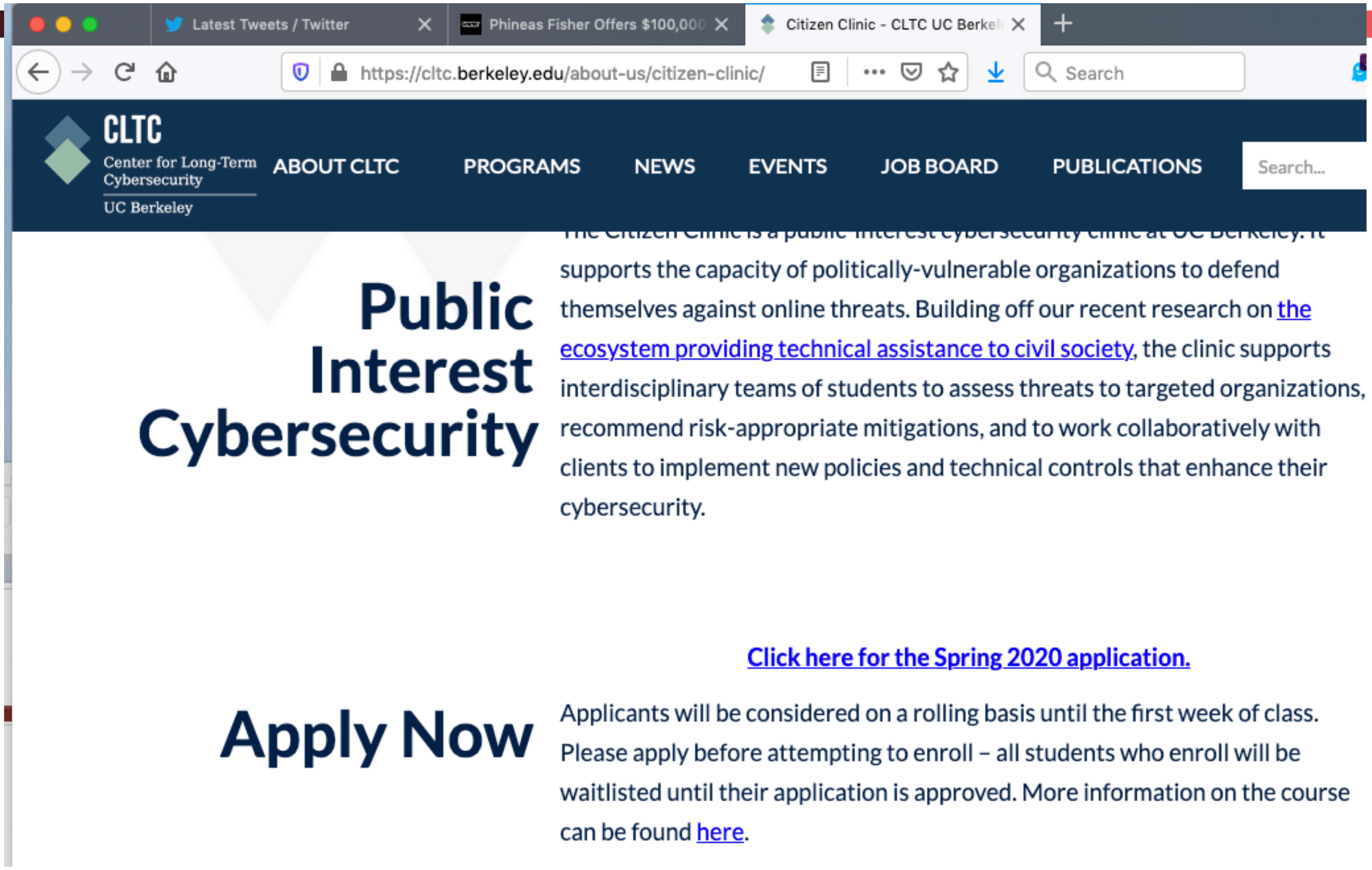
Network Censorship



"Mass surveillance is the elegant oppression, a panopticon without bars. Its cage is small but out of sight, behind the eyes - on the mind."

- Taylor Swift

UC Berkeley Citizen Clinic


Latest Tweets / Twitter
Phineas Fisher Offers $100,000
Citizen Clinic - CLTC UC Berkel
https://cltc.berkeley.edu/about-us/citizen-clinic/
Search
CLTC
Center for Long-Term Cybersecurity
UC Berkeley
ABOUT CLTC
PROGRAMS
NEWS
EVENTS
JOB BOARD
PUBLICATIONS
Search...

Public Interest Cybersecurity

The Citizen Clinic is a public-interest cybersecurity clinic at UC Berkeley. It supports the capacity of politically-vulnerable organizations to defend themselves against online threats. Building off our recent research on the ecosystem providing technical assistance to civil society, the clinic supports interdisciplinary teams of students to assess threats to targeted organizations, recommend risk-appropriate mitigations, and to work collaboratively with clients to implement new policies and technical controls that enhance their cybersecurity.

Click here for the Spring 2020 application.

Apply Now

Applicants will be considered on a rolling basis until the first week of class. Please apply before attempting to enroll – all students who enroll will be waitlisted until their application is approved. More information on the course can be found here.

We Saw Surveillance... Now Lets See Censorship

- Who wants to censor?
- Businesses: Don't want users browsing PornHub at work
  - There is huge potential legal liability if you don't!
- Many countries: Child Exploitation Material
  - Notably the UK requires this of ISPs:
    Block known Child Exploitation sites
- Many countries: Porn
  - Again, notably the UK requires on-by-default porn filters
- Many countries: Politics
  - Russia, China, Iran, etc...
  - China was the pioneer here, but everyone else has followed suit

Mechanisms...

- DNS Interdiction/Mandates
  - China's Great Firewall
  - Turkey v Twitter
- IP Blocking
- On-path attack
  - China's Great Firewall
- In-path proxies
  - Selective: UK
  - Mandatory: Russia
- Serious Voodoo:
  - China's Tor Blocking
  - China's Great Cannon

Evasion...

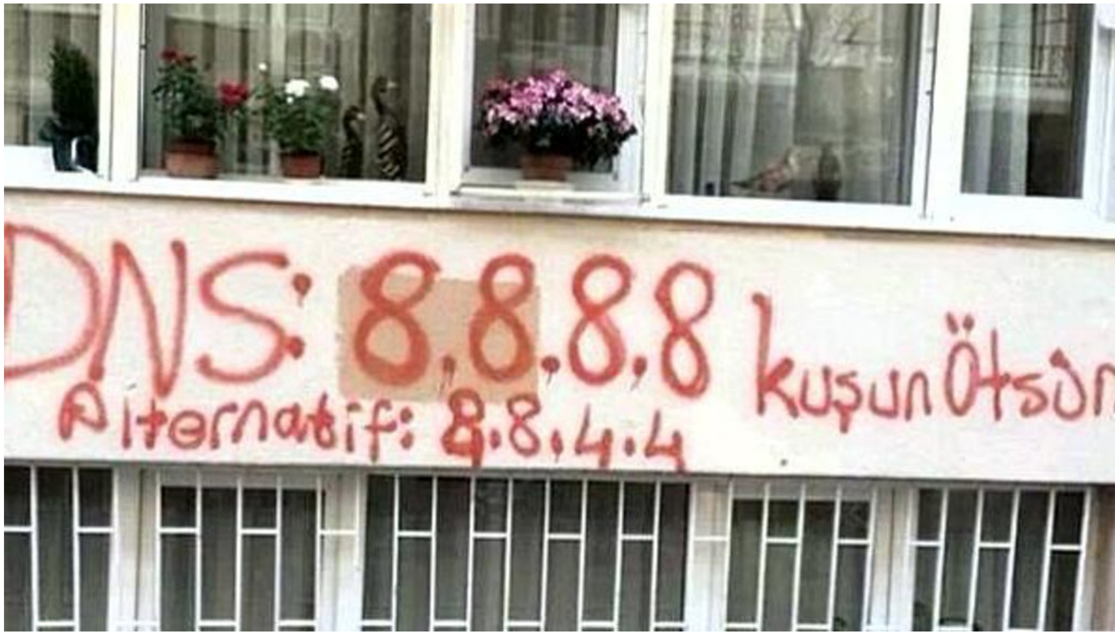
- TLS:
  - Forces a censor into an "all or nothing" decision:
    Can either block the whole site or allow the whole site
- But the censor ***can*** always identify the site
  - TLS Server Name Identification and/or the DNS request
- Well, now they can:
  - For a while, you could say in TLS you want to talk to site A...
    But on HTTP in TLS say you want to talk to site B
  - And if the server supported both sites:
    A Content Delivery Network (CDN) like CloudFlare or Google's App Engine), 👍
  - "Domain Fronting" no longer supported by the CDNs since it really is a bug, not a feature
    - Plus ~~CrimeFlare~~ CloudFlare wants to do business in China with a local partner

Evasion...
VPNs & Other Software

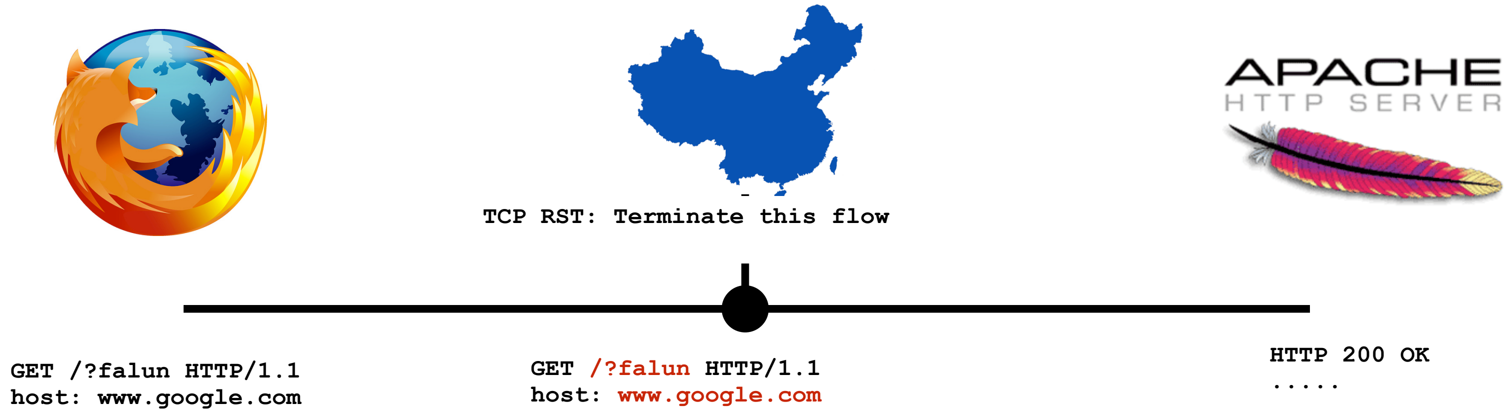
- Create an encrypted link to a non-censored network
  - And through that link direct all your traffic
- Ends up in a cat & mouse game with the censors
  - Censor can't block ***all*** VPNs:
    Business travelers may depend on them so can't just go "terminate"
  - Can block all ***public*** VPNs:
    Buy the services, detect & block them
- So if you are visiting China...
  - Set up your ***own*** VPN or ssh tunnel back here in the US

Blocking DNS...
Force the ISPs to Comply

- Turkey v Twitter in 2014:
  - Turkey got into a spat with Twitter...
  - Twitter was allowing recordings of Turkish government corruption
- Turkey's initial response:
  - ALL ISPs, block Twitter's DNS entry
- People's initial response:
  - Switch DNS servers to 8.8.8.8
- Turkey's Subsequent Response:
  - Block 8.8.8.8...



The Great Firewall:
Packet Injection Censorship Including DNS



```
TCP RST: Terminate this flow
```

```
GET /?falun HTTP/1.1
host: www.google.com
```

```
GET /?falun HTTP/1.1
host: www.google.com
```

```
HTTP 200 OK
.....
```

- Detects that a request meets a target criteria
  - Easiest test: "Looks like a search for 'falun':
    - Falun Gong (法輪功), a banned quasi-religious organization
- Injects a TCP RST (reset) back to the requesting system
  - Then enters a ~1 minute "stateless block": Responds to all further packets with ~~RSTs~~ SYN/ACK PACKETS!!!
- Same system used for DNS censorship:
  - dig www.facebook.com @www.tsinghua.edu.cn

Live Demos of The Great Firewall...

- `dig +short AAAA www.tsinghua.edu.cn`
  - `www.d.tsinghua.edu.cn.`
  - `2402:f000:1:404:166:111:4:100`
- `sudo tcpdump -vvv -i en0 -s 1800 host 2402:f000:1:404:166:111:4:100`
- `dig www.facebook.com @2402:f000:1:404:166:111:4:100`
- `dig www.benign.com @2402:f000:1:404:166:111:4:100`
- `dig TXT www.facebook.com @2402:f000:1:404:166:111:4:100`
- `curl --header "Host: www.google.com" "http://[2402:f000:1:404:166:111:4:100]/?falun"`

Features of the Great Firewall

- The Great Firewall is on-path
  - It can detect and inject additional traffic, but not block the real requests from the server
- It is single-sided
  - Assumes it can see only one side of the flow:
    Can send SYN, ACK, data, and get a response
- It is very stateful
  - Must first see the SYN and ACK, and reassembles out of order traffic
- It is multi-process parallel
  - ~100 independent processes that load-balance traffic
- The injected packets have a distinct side channel
  - Each process increments a counter for the TTL
  - IPIDs are also "odd" but harder to categorize

On Path v In Path

- China went largely with an on-path solution
  - Mostly because they were early, and repurposed network intrusion detection
- Most others use an ***in-path*** solution
  - Generally starting with a web proxy such as ***squid***:
    A MitM tool for intercepting and modifying web traffic
  - Initial use was as a cache for web traffic:
    Designed to speed up web surfing when bandwidth was more expensive and CDNs didn't predominate
  - Now a large market from commercial vendors

Benefits of Both

- On Path:
- Easier deployment:
  Just put into the network backbone
- Fail "safe":
  If device craps out, the net still works
- Easy to scale:
  Load balancer/NIDS approach

- In Path:
- Can't use Layer 3 evasions
- Easy Deployment for ISPs
- Potential to "slow down", not just block
- Can MitM TLS connections with a client-added root cert
- Lots more commercial solutions

Selective Proxy:
Mandatory in the UK

- For some sets of IPs that ***may*** host child exploitation material...
  - ISP redirects just those IPs to a proxy that strips out any known-bad items
  - Allows "fail safe" for the ***rest*** of the Internet
- Of course, for TLS this has to be entirely block-or-not!

The UK "Virgin Killer" Incident

- An album cover for "Virgin Killer" by the Scorpions is on the page about that album
  - And it is borderline at best...
    The record company executive who created it really should have been jailed
- UK's "Internet Watch Foundation" called it CP...
  - So ***all*** Wikipedia traffic got routed through the filtering proxy...
- With very bad effects!
  - No TLS connections allowed
  - Editing attempts w/o TLS triggered the bot detector

Kazakhstan v Browsers

- Kazakhstan uses in-path censorship...
  - But doesn't want to just block sites like Wikipedia that are TLS only but may contain "unfavorable" content
- Their attempt: ***require*** everyone to install another root certificate
  - A feature present for corporate networks which often use in-path monitoring on TLS
- Then just MitM all that traffic to do the fine-grained censorship
- Mozilla and Google said "Hell No!"
  - Alternate roots are only for businesses:
    The browsers modified to reject the Kazakhstan root out of hand
- Kasakhstan backed down...

Advanced Chinese Voodoo:
The Great Cannon and Active Probing...

- China pioneered Internet censorship
  - Partially to advantage local Internet companies
- But manly because the government is a group of seriously repressive A*()holes lead by a guy who looks like Winnie the Pooh
  - Tienamen Square Massacre probably killed >1000
  - The history of the "One Child" policy
  - Ethnic cleansing of Uighurs in Xinjiang
  - And now Hong Kong...
- So two pieces of Advanced Voodoo...
  - Both areas that I was involved in researching

A Chinese Problem: They Can't Block Github!!

- Github is TLS only...
  - So can't selectively censor
- Github can't be blocked since so many Chinese tech businesses are:
  - Pull open source repo from GitHub
  - Put on white box hardware
  - Profit!
- Activists know this:
  The "greatfire.org" activists host instructions on evading the Great Firewall on GitHub

Enter the Chinese Great Cannon

- The Great Cannon is a dedicated Internet attack tool probably operated by the Chinese government
  - An internet-scale selective man-in-the-middle designed to replace traffic with malicious payloads
  - Used to co-opt unwitting foreign visitors to Chinese web sites into participating in DDoS attacks
  - Almost certainly also has the capability to "pwn-by-IP":
    Launch exploits into targets' web surfing
  - "Great Cannon" is our name:
    the actual Chinese name remains unknown
- Structurally related to the Great Firewall, but a separate devices

The DDoS Attack on GreatFire and GitHub

- GreatFire is an anti-censorship group
  - Currently uses "Collateral Freedom": convey information through services they hope are "Too Important to Block"
  - GitHub is one such service:
    You can't block GitHub and work in the global tech economy
- GreatFire's CloudFront instances DDoSed between 3/16/15 and 3/26
- GreatFire's GitHub pages targeted between 3/26 and 4/8
  - GitHub now tracks referer to ignore the DoS traffic

The DDoS used Malicious JavaScript...

- JavaScript in pages would repeatedly fetch the target page with a cache-busting nonce
  - Vaguely reminiscent of Anonymous's "Low Orbit Ion Cannon" DDoS tool
- JavaScript appeared to be served "from the network"
  - Replacing advertising, social widgets, and utility scripts served from Baidu servers
- Several attributed it to the Great Firewall
  - Based on DDoS sources and "odd" TTL on injected packets
  - But it didn't really look quite right to us...

The Baidu Malicious Scripts

```
eval(function(p,a,c,k,e,r){e=function(c){return(c<a  ....
,'|||function|Date|script|new|var|jquery|com|||getTime|url_array|r_send2|responseTime|count|x3c|unixtime|
startime|write|document|https|github|NUM|src|get|http|requestTime|js|r_send|setTimeout|getMonth|getDay|
getMinutes|getSeconds|1E3|baidu|min|2E3|greatfire|cn|nytimes|libs|length|window|jQuery|code|ajax|url|dataType|
timeout|1E4|cache|beforeSend|latest|complete|return|Math|floor|3E5|UTC|getFullYear|getHours'.split('|'),0,{}))
```

- Baidu servers were serving a malicious script...
  - Packet with a standard JavaScript packer
    - Probably http://dean.edwards.name/packer/ with Base62 encoding
  - Payload is "keep grabbing https://github.com/greatfire and https://github.com/cn-nytimes"
    - Github quickly defanged the attack: You first have to visit another page on Github for these pages to load
- Others quickly concluded the Great Firewall was responsible...

But The Malicious Reply For The Baidu Script Seemed "Odd"

```
IP (ttl 64,  id 12345) us > Baidu: [S]   seq 0,                   win 8192
IP (ttl 47,  id 12345) Baidu > us: [S.]  seq 0,         ack 1     win 8192
IP (ttl 64,  id 12346) us > Baidu: [.]   seq 1          ack 1     win 8192
IP (ttl 64,  id 12346) us > Baidu: [P.]  seq 1:119      ack 1     win 8192
IP (ttl 201, id 55896) Baidu > us: [P.]  seq 1:108      ack 119   win 767
IP (ttl 202, id 55741) Baidu > us: [P.]  seq 108:1132   ack 1     win 768
IP (ttl 203, id 55699) Baidu > us: [FP.] seq 1132:1238  ack 1     win 769
```

- The injected packets had incremented TTLs and similar funky IPID sequence
  - The Great Firewall's side channel
- The second and third packets had bad ACK values and incrementing windows too
- But the dog that didn't bark:
  - No legitimate reply from the server?!??

The Eureka Moment: Two Fetches

- Built a custom python script using scapy
  - Connect to server
  - Send request
  - Wait 2 seconds
  - Resend the same request packet
- What happens? The real server replied!?!
  - The first request was attacked by the cannon and ***replaced*** with a malicious payload
  - The second request passed through unmolested to the real server
    - Who's reply indicated it never received the original request!

So Now Its Time To Categorize

- Send "valid target" request split over 3 packets:
  - Ignored
- Send "Naked packets": just a TCP data payload without the initial SYN or ACK
  - May trigger response
- Send "No target than valid target"
  - Ignored
- Retry ignored request
  - Ignored (at least for a while...)
- One over from target IP
  - Ignored

Tells us the basic structure: Flow Cache and Stateless Decider

- Non data packets: Ignore
- Packets to other IPs: Ignore
- Data packet on new flow:
  Examine first packet
  - If matches target criteria AND flip-a-coin (roughly 2% chance): Return exploit and drop requesting packet
- Data packet on existing flow (flow cache): Ignore
  - Even if it decided to inject a packet on this flow

Localizing the Cannon

- Traceroute both for the cannon and for the Great Firewall
  - TTL limited data for the Cannon
  - TTL limited SYN, ACK, DATA for the firewall
- Tracerouted to two intercepted targets on different paths
  - One in China Telecom, the other in China Unacom
  - Both targets intercepted by the Cannon in the same location as the Firewall

Operational History: LBNL Time Machine

- Examine Lawrence Berkeley National Lab's Time Machine for the odd-TTL signature:
  - LBNL does a bulk record start of all connections
- Initial attack: Targeting GreatFire's "collateral freedom" domains
  - Unpacked payload, showed evidence of hand-typing (a 0 vs o typo fixed)
  - Near the end, GreatFire placed a 302 redirect on their domains to www.cac.gov.cn,
    - Makes the DOS target the Cyber Administration of China!
- Second attack: the GitHub targeting
  - Packed payload, but same basic script

Build It Yourself With OpenFlow

- Start with an OpenFlow capable switch or router
- Default rule:
  - Divert all non-empty packets where dst=target and dport=80
- Analysis engine:
  - Examine single packet to make exploitation decision
  - If no-exploit: Forward packet, whitelist flow
  - If exploit: Inject reply, whitelist flow
- Matches observed stateless and flow-cache behavior
  - Other alternative of "BGP-advertise target IP" would probably create a traceroute anomaly (which unfortunately we didn't test for at the time)

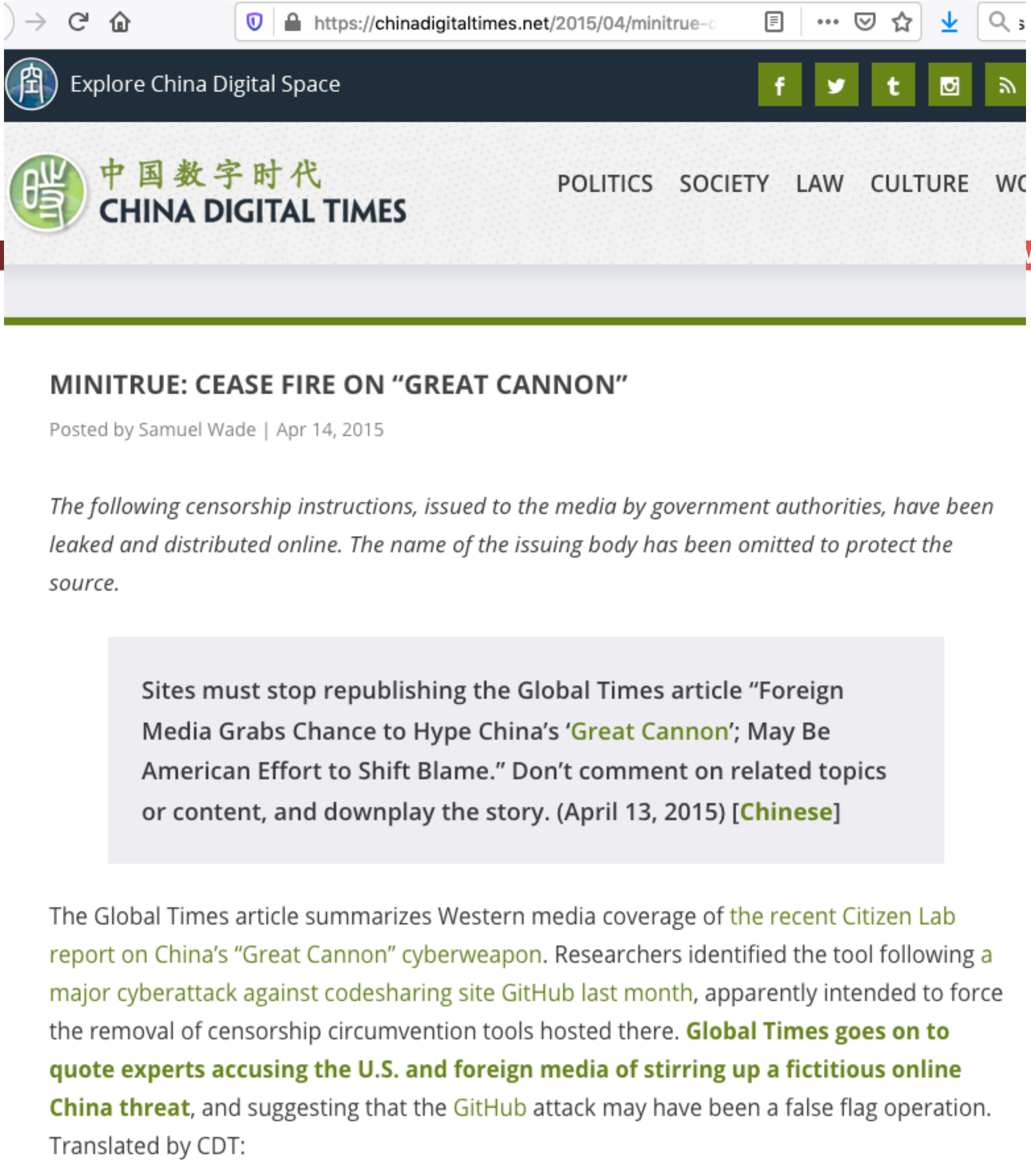
Modifying The Cannon For "Pwn By IP" targeting

- The Cannon is good for a lot more than DDoSing GitHub...
  - A nation-state MitM is a very powerful attack tool...
- Change criteria slightly: select traffic FROM targeted IP rather than to IP
  - Need to identify your target's IP address in some other means
    - Emails from your target, "benign" fishing emails, public data, etc...
- Expand the range of target scripts
  - "Looks like JavaScript" in the fetch
- Reply with "attack the browser" payload
  - Open an iframe pointing to an exploit server with your nice Flash 0-day...
- This change would likely take less than a day to implement!

Modify For "Perfect Phishing" Malicious Email from China

- Identify your target's mail server
  - dig +mx theguyIwanttohack.com
- Intercept all traffic to your target's mail server
  - Redirect to a man-in-the-middle sink server that intercepts the email
    - Able to strip STARTTLS
    - Can't tamper with DKIM, but who validates DKIM?
  - Any word documents to your target? Modify to include malcode
  - Then just send/receive from the cannon to forward the message on to the final server
- Really good for targeting activists and others who communicate with Chinese sources
  - A phishing .doc email is ***indistinguishable*** from a legitimate email to a human!
- I could probably prototype this in a week or two

Oh, and We Know We Struck A Nerve...



https://chinadigitaltimes.net/2015/04/minitrue-c
Explore China Digital Space
中国数字时代
CHINA DIGITAL TIMES
POLITICS
SOCIETY
LAW
CULTURE
WC

MINITRUE: CEASE FIRE ON "GREAT CANNON"

Posted by Samuel Wade | Apr 14, 2015

The following censorship instructions, issued to the media by government authorities, have been leaked and distributed online. The name of the issuing body has been omitted to protect the source.

> Sites must stop republishing the Global Times article "Foreign Media Grabs Chance to Hype China's 'Great Cannon'; May Be American Effort to Shift Blame." Don't comment on related topics or content, and downplay the story. (April 13, 2015) [**Chinese**]

The Global Times article summarizes Western media coverage of the recent Citizen Lab report on China's "Great Cannon" cyberweapon. Researchers identified the tool following a major cyberattack against codesharing site GitHub last month, apparently intended to force the removal of censorship circumvention tools hosted there. **Global Times goes on to quote experts accusing the U.S. and foreign media of stirring up a fictitious online China threat**, and suggesting that the GitHub attack may have been a false flag operation. Translated by CDT:

Serious Policy Implications

- China believes they are justified in attacking those who attack the Great Firewall
  - Both DoS attacks targeted GreatFire's "Collateral Freedom" strategy of hosting counter-censorship material on "too critical to block" encrypted services
- Baidu was probably a ***bigger*** victim than GreatFire
  - GreatFire and Github mitigated the attack
    - GreatFire: Collateral Freedom services now block non-Chinese access, in addition to the DOS-redirection strategy
    - GitHub: Targeted pages won't load unless you visit some other page first
  - But Baidu services (and all unencrypted Chinese webservices) must be considered explicitly hostile to those outside of China
    - It ***can't*** be a global Internet brand
    - Note, we saw at least one injection script on qq.

And Active Probing...

- You see some encrypted goop...
  - No framing, no nothing
- Is it OK to block this IP?
  - It could be someone using a VPN/censorship evasion system
  - It could be something else
- A ***robust*** solution for any public VPN type system...
  - Just handshake it and see!

China Does This Operationally...

- For several different protocols
- See request on the Internet
  - Using yet ANOTHER sensor:
    - It doesn't reassemble (unlike the Great Firewall)
    - It does rely on seeing the SYN (unlike the Great Cannon)
  - Not necessarily at the same location as the Great Firewall's sensor
- Trigger another system to do a handshake
  - Apparently through what appears to be a large proxy network to prevent IP blocking
  - If handshake succeeds, block IP