Hi !

modirum

WHY AM I HERE?

- Used FreeBSD since ~2000
- Love open source
- Been working in the payment industry since 2003
- Rare to hear the victims' side of a DDoS story
  - ..except if it's a "success story" (ours is not.. this story has no winners)

modirum

WHY ARE _WE_ HERE?

- Hosting in-house-developed SW on FreeBSD since 2003
- Authenticating users during on-line card payments
- SW for card issuers (your bank), merchants and processors (Amazon, PayPal) and card companies (that other logo on your card)
- Recently had a huge target painted on our backs

PREVIOUSLY ON

THE BLAME GAME

THE WORLD WE LIVE IN

- Three players
  - Those writing the requirements
  - Those covering their asses
  - Those who are blamed in the end
- Several sets of requirements
  - PCI DSS and 3DS
  - Payment systems (Visa, MC, etc.)
  - Legal (PSD2, GDPR, FSAs, local law, etc.)
  - Customer specific

modirum

- Photos of password files
- "Please document that grep(1) supports regular expressions"
- Auditor storing evidence collected from clients on Desktop (WinXP)
  - ..also used for adult entertainment..
- "root account must have a strong password under split knowledge and dual control"
  - No, disabling your root account won't do.

THE BLAME GAME

CONTINUES

modirum

WHAT I WANT TO COVER

- The anatomy of a DDoS campaign
- The personal cost - especially in a small organisation
- Defences that work
- Lessons learned

modirum

IN THE BEGINNING WAS PEACE AND QUIET

- Most merchants prefer liability shift - uses 3-D Secure
- Banks buy Access Control Servers to authenticate
- Shady merchants won't use 3-D Secure, and don't care
- Banks quietly reimburse cardholders, no need for drama
- Card companies implement "backup solutions"; if authentication is broken, skip it

modirum

...ENTER THE PAYMENT SERVICES DIRECTIVE (PSD2)

- Requires (strong?) authentication for internet payments
- Remaining merchants have to implement support
  - ..even those who don't care whose money is being spent
- Bad guys pissed: Can't even buy crypto with stolen cards anymore!
  - ..let's kick over this authentication server and see what happens

WHAT DOES THIS BUTTON DO?

TRIAL RUNS

MARCH 2021: THE FIRST ATTACK WAVE

Let's face it: We were woefully unprepared

- Normal inbound traffic levels: ~10-20 Mbps
- War-time inbound traffic: >10Gbps and 1Mpps at ISP, ~200Kpps reaching us on our 1Gbps links
- Run-of-the-mill attacks: UDP, TCP SYN, ICMP
  - Cheap garbage traffic
  - "10 minutes free" DDoS platform sales campaigns

modirum

OK WHAT JUST HAPPENED..??

- "Backup solutions" effectively de-brick stolen cards
- ISP-owned DDoS mitigation platform of symbolic value:
  - Running out-of-box config
  - Three years out-of-date
  - Not enabled for our networks(!)
- Secondary ISP mitigates by null-routing us in self-defence
- Laugh now - might make up for all our crying

halp

Sysadmin pleading to upstream ISP for help

THE REAL COSTS

- Individual humans don't scale well
- Lack of sleep is only part of the problem
  - ..it gets really personal, really quickly
- Anxiety, ruined birthdays, suffering families
- The price of doing a lot with little (and few)
  - (Thanks, FreeBSD)

WELL THAT WAS FUN

LEVEL UP!

AUGUST 2021: ROUND TWO

- July was quiet - vacation time?
- Increased volumes, new attack vectors
  - Full TCP handshakes, some TLS
  - Exploiting BGP bypasses ("forgotten" peers)
  - Passing the 100Gbps, 10Mpps mark
- Time before countermeasures bypassed: 1-3 days

modirum

NOW WE KNOW THAT

- The attacks are clearly profitable
- Professionals at work: They go on holiday!
- No ransom notes: Instant payout

- But also: Attacks cost real money
- Cat-and-mouse: We really need to up the game
  - ..make attacks prohibitively expensive

WEARING US DOWN

GOING ALL IN

modirum

OCTOBER 2021: THINGS GET PERSONAL

- DDoS mitigation works reasonably well
- Attackers pull all the stops:
  - TLS-level exhaustion smoke screen
  - Application stack profiling
  - Custom layer 7 attack tooling
- Time before countermeasures bypassed: 5-15 minutes
- Mentally and physically devastating - for months

modirum

...WITHSTAND DENIAL-OF-SERVICE (DOS) ATTACKS THAT COULD FORCE FALLBACK TO LESS SECURE VERIFICATION METHODS OR PROVIDE COVER FOR OTHER ATTACKS...

PCI guidance (for auditors)

modirum

WHY IS THIS A "BLAME GAME"?

- Requirements extremely vague
  - Will certainly be read differently by the blaming party
- It's hard to work from under the bus
  - Loss from fraud << loss from potential outage
  - Remember the "backup solutions"?
- A DDoS is not an "incident", it's Force Majeure!
  - (Assuming you've done a minimum of planning)

SO...

WHAT WORKS?

modirum

USE THE CLOUD, LUKE!

Customers, the Internet, and the bartender

modirum

BEEFY HARDWARE AND FAT PIPES

- FreeBSD (BSDrp) routers, 10GbE everywhere
- OPNsense firewalls, 10GbE everywhere
- Before: 1Gbps, two ISPs, either-or
- Now: 10Gbps, several ISPs, distributed traffic

SYNCOOKIES AND PF CONFIG

- Imported from OpenBSD, made multi-threaded
  - Massive effect on state-exhaustion attacks
- Most defaults are from the modem era:
  - No need to wait a minute for a TCP handshake to complete
  - No need to keep states around "forever"
- With VIMAGE-jails: Per-jail limits and rules!

NGINX CONFIG NASTIES

- Make sure handshakes are handled by all your CPUs
- Use rate limiting - tailored for each resource type
- Lower your timeouts - more defaults from the modem era
- `return 444;` is incredibly resource-saving
  - Use it wherever you may be seeing 404/403 storms
- Fine-grained `location` statements and rules
- Reserve resources for monitoring - in nginx *and* back-end

CLOUDFLARE MAGIC TRANSIT

- They do BGP route announcement of our networks
- GRE tunnels from CF sites around the world
- Ingests, filters and shapes most traffic coming our way
- Our transit providers see only GRE traffic (we don't actually announce to them unless Cloudflare falls over)
- Best of all: No keys outside the kingdom!

(but it's no "god mode")

modirum

L3/4 DDOS MITIGATION

- Kinda like e-mail greylisting:
  - Deliberately sabotage ("challenge") TCP handshakes
    - Well-behaved TCP implementations will retransmit
- Assumes attackers are poor and bots are stupid
  - ..neither is true!
- Only works when you're the only game in town
  - Remember, ricochet is a killer

AH WHAT THE HELL...

SIGH

modirum

2022-NOW

- Attacks in the Tbps and >>100Mpps-scale
- Surgical strikes:
  - Attackers know where it hurts
  - ..and what we can't mitigate
- DDoS mitigation causes devastating side-effect
  - Bots live behind end-users' ISPs, self defence kicks in
  - Systems are fine, but users suffer

WHEN DO YOU PLAN TO SOLVE THIS?

Those who can solve this

modirum

WHY NOT LAYER 7 IN THE CLOUD?

- The obvious: GDPR, decrypting sensitive data in the cloud
- Less obvious: Technical constraints from 3rd parties
- Would it help? Not much:
  - Even an L7 proxy in the cloud is protected
  - The same interference will apply there
- May protect a single system from direct attack
  - ..but won't prevent the side-effects

OTHER HELPFUL SUGGESTIONS

- Get more hardware!
  - We do ~100k TLS handshakes/s. CF drops >1M/s.
    - Do the math. Who's going to pay for that?
- Accelerate your TLS!
  - Most EC/RSA hardware is for low core count servers
- Use proprietary TLS - open source is vulnerable!
  - Don't get me started. I mean it. Don't.

LESSONS LEARNED

- The Internet is no longer a friendly place (duh!)
- Sustained DDoS attacks expose the "bus factor" in spectacular new ways
- This kind of fight can not be fought alone
- Peace is temporary
- The best defence is making attacks expensive
- Don't be afraid of violating (some) standards - teapots cost a lot more than 444s

modirum

THANK YOU ALL!

- Contributors of all kinds
- Organisers of this event
- Everyone working to make the community tick

Until next time..

NGINX CONFIGS

modirum

NGINX CONFIG HACKS – GLOBAL

```
# Adapt to your CPU count, leave (some) room for other stuff
worker_processes 8;
# Each worker has at least two sockets; leave some headroom here
worker_rlimit_nofile 65536;

# Works for us..
worker_cpu_affinity auto;
events {
    # We want round-robin-like handling of inbound requests
    accept_mutex on;
    # During war-time we want plenty of these; each one does very little work
    worker_connections 2048;
}

http {
    # Define rate limiting zones - totally application dependent!
    limit_req_zone $binary_remote_addr$request zone=request_5:100m rate=5r/s;
    ....
}
```

modirum

NGINX CONFIG HACKS – SERVER

```
# This is actually a kinda high value, but some people are slow and far away..
client_body_timeout 10s;

# Attackers won't wait for data anyway, no reason to keep connections open
send_timeout        2s;

# Don't sit on dead connections
reset_timedout_connection on;


# And for the love of $deity: DO NOT use 'listen ... reuseport'!
# This effectively limits handshakes to a single CPU core.
```

modirum

NGINX CONFIG HACKS – LOCATIONS

```
# Catch 403s from limit_except, pass to first location
error_page 403 /444.html;
location /444.html {
    return 444;
}

# Be specific in your location blocks (better: have different ones for different
request types!)
location ~ ^/application/(known|endpoints)[^/.]* {
    # Refuse any other request type
    limit_except GET POST {
        deny all;
    }
    # If someone tries a GET on your POST-only endpoint, drop connection
    if ($request ~ "(GET|HEAD) +/application/(post-only|endpoints)/? +HTTP/") {
       return 444;
    }
    # Use rate limiting..
    limit_req zone=request_5 burst=7 delay=5;

    # Pass to a named backend
    proxy_pass http://applications;
}
```

modirum

NGINX CONFIG HACKS – BACKENDS

```
# First we define a backend for our applications:
upstream applications {
  server localhost:8180 max_conns=10000;
  zone applications 128k;
  keepalive 20;
}
```

```
# Then we define another for monitoring:
upstream monitoring {
  server localhost:8182 max_conns=100;
  zone monitoring 32k;
  keepalive 2;
}

# Finally make sure our monitoring endpoints use the dedicated back-end:
location ~ ^(/status/.*) {
    allow ....;
    deny all;
    proxy_pass http://monitoring;
}
```