

THE VARIOUS WAYS YOUR RTC MAY BE CRUSHED

Sandro Gauci, Enable Security

May 22, 2019

INTRODUCTION

WHOAMI, WHY ETC

OUR JOB AS PENTESTERS

- we test systems for security issues
- special focus on VoIP and WebRTC
- especially interested security vulnerabilities specific to RTC

**HOW HARD WOULD IT BE TO TAKE AWAY
THE *RT* FROM YOUR RTC SYSTEM?**

WHAT THIS PRESENTATION IS NOT ABOUT

- volumetric DoS attacks
- DDoS that relies on saturating bandwidth

THEN WHAT IS IT ABOUT?

- Compilation of DoS vulnerabilities affecting RTC systems
- Finding the common denominator
- Removing any identifying information (we do sign NDAs)
- Describing some of them in a bit more detail

AGENDA

- DoS on:
 - Signalling
 - Media
 - Monitoring tools
 - Callbacks
 - Security protection
- Evading protection
- What can be done

WHO IS THE TARGET AUDIENCE FOR OUR PRESENTATION?

- VoIP engineers
- WebRTC infrastructure engineers
- Vendors
- Service providers
- Security researchers interested in RTC

DEMO MACHINE SPECS


























DROPLETS (6)

ddos-target	Add tags		
Image	Ubuntu ddos-target-20190501	Region	FRA1
Size	8 vCPUs 32GB / 200GB Disk (\$260/mo) Resize	IPv4	157.230.126.255
		IPv6	Enable
		Private IP	Enable
test-test-4		157.230.19.239	
test-test-3		104.248.143.55	
test-test-2		157.230.125.1	
test-test-1		157.230.117.213	
test-test-0		157.230.98.81	

DDoS target specs

DEMO MACHINE SPECS

DROPLETS (6)

  ddos-target	157.230.126.255	 	
  test-test-4	Add tags	 	
Image	 Ubuntu test-base-snapshot	Region	FRA1
Size	1 vCPUs 0.5GB / 20GB Disk (\$5/mo) Resize	IPv4	157.230.19.239
		IPv6	Enable
		Private IP	Enable
  test-test-3	104.248.143.55	 	
  test-test-2	157.230.125.1	 	
  test-test-1	157.230.117.213	 	
  test-test-0	157.230.98.81	 	

Attack node specs

DISCLAIMER SLIDE

- yes, there are better ways of setting stuff up (clustering, load balancing, separate systems etc)
- trying to replicate what we see in real pentests
- monitoring for bandwidth problems was done using `iperf3`
- the demos are not meant to show that a particular software package is vulnerable

SIGNALLING

SIP FLOODING

- Continuous sending of SIP messages at high rates
- **INVITE** flooding can be extremely effective
- **REGISTER** flooding, or just authentication attempts using other methods, too

SIP FLOODING: WHY IS IT SO EFFECTIVE?

The obvious: SIP parsing may be CPU intensive

SIP FLOODING: WHY IS IT SO EFFECTIVE?

REGISTER FLOOD

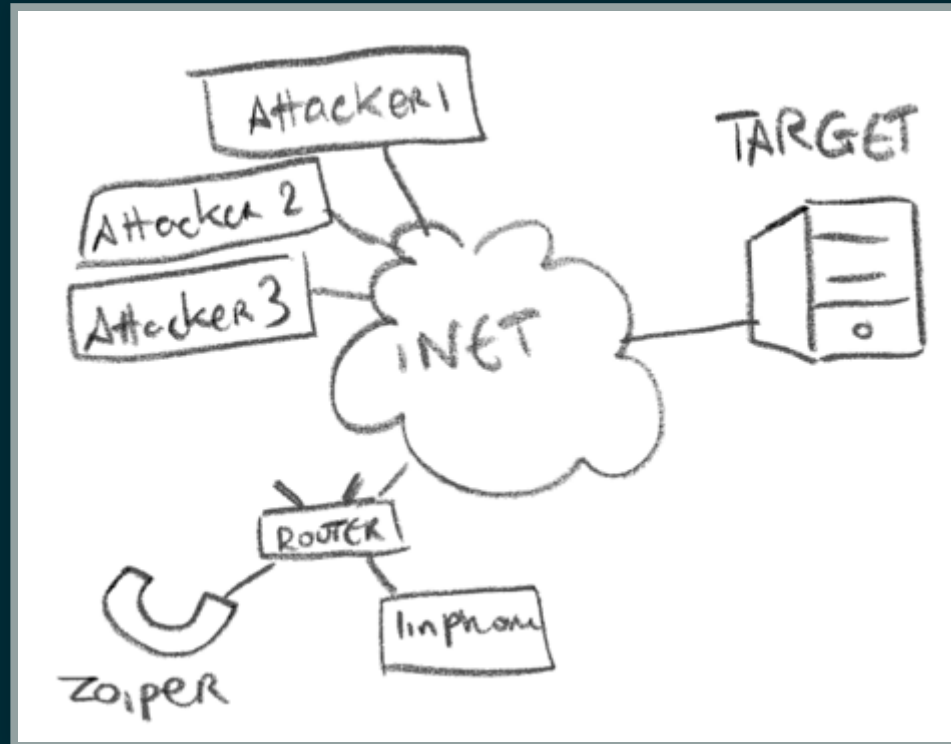
- In the case of authentication: database lookups may be expensive
- Address of Record (AOR) storage and lookup
- The custom stuff (extra logging)

SIP FLOODING: WHY IS IT SO EFFECTIVE?

INVITE FLOOD

- New UDP ports are allocated for each call
- Each dialog allocates memory for state
- The custom stuff (e.g. voicemail, call recording)

SIP FLOODING: DEMO SETUP



Demo setup

CHANGES DONE TO TARGET

ASTERISK PJSIP CONFIGURATION

```
threadpool_initial_size=20  
threadpool_auto_increment=5  
threadpool_max_size=100
```

KAMAILIO CONFIGURATION

```
#!/define WITH_AUTH  
#!/define WITH_MYSQL  
#!/define WITH_TLS  
#!/define WITH_NAT  
mlock_pages=yes  
children=8
```

SIP FLOODING: DEMO (KAMAILIO)



SIP FLOODING: DEMO (ASTERISK PJSIP)



SIP FLOODING: DEMO (ASTERISK CHAN_SIP)



TCP AND TLS FLOODING

- Keeping connections open is easy for an attacker (i.e. SYN-ACK flood)
- Running out of File Descriptors is a problem
- Asterisk advisory: [AST-2018-005](#)

TLS CERTIFICATE FLOODING

- A concern when TLS client certificates are required
- The attack involved creating client certificates with a large number of FQDNs
- Flooding the server which was checking each FQDN against a whitelist

WEBSOCKET PROPRIETARY PROTOCOL MALFORMED MESSAGE

WebRTC system that made use of a JSON based
custom protocol

```
{"magic": "magic string", "length": 32, "type": "message"}  
{"type": "init", "token": "JWT"}
```

WEBSOCKET PROPRIETARY PROTOCOL MALFORMED MESSAGE

- Blackbox fuzzing (*hint: radamsa)
- the system starts returning 5XX errors with the following message:

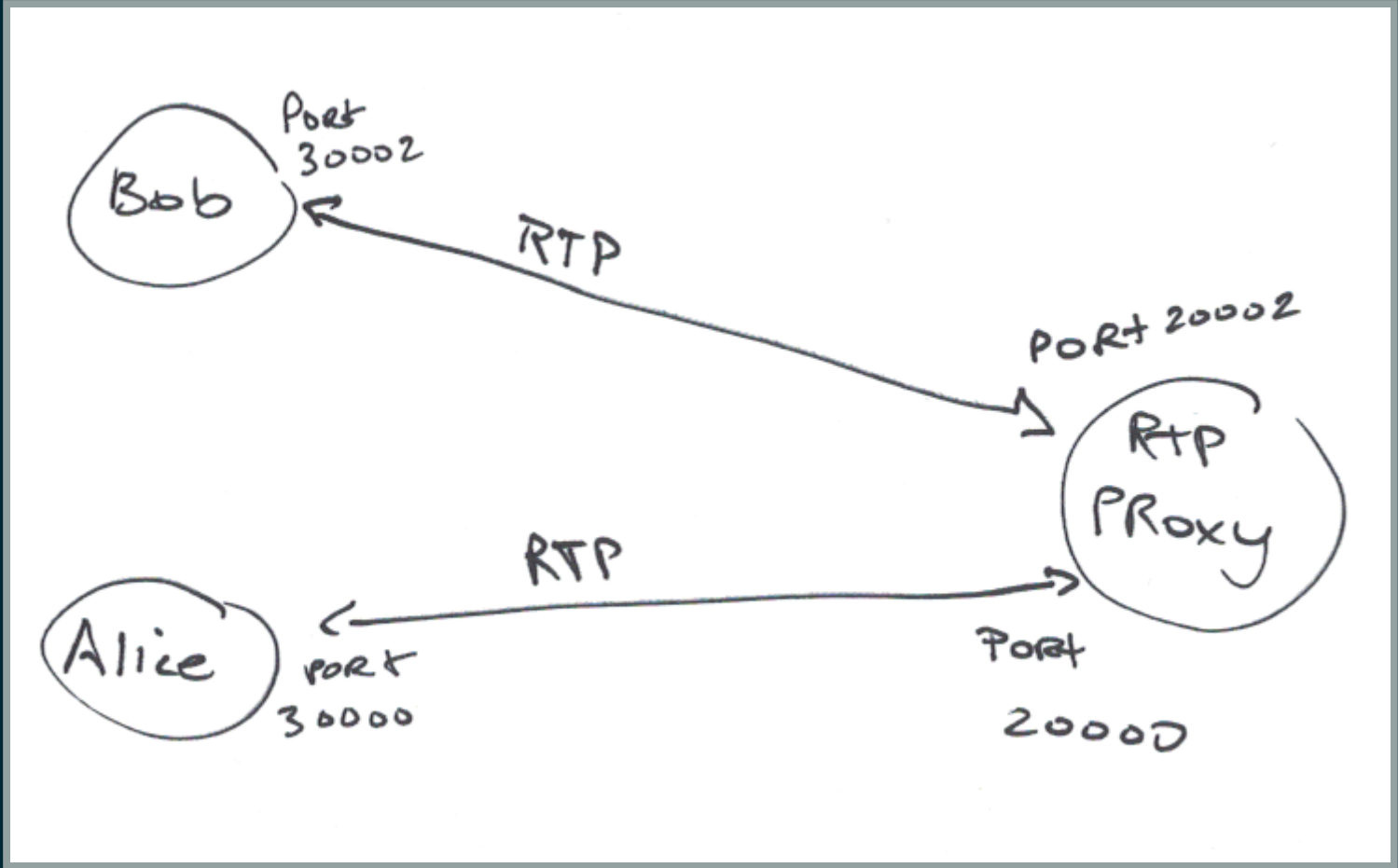
```
{"magic": "magic string", "length": 320000, "type": "message"}  
{"type": "init", "token": "JWT"}
```

- Repeated sending of this message == the system became unavailable

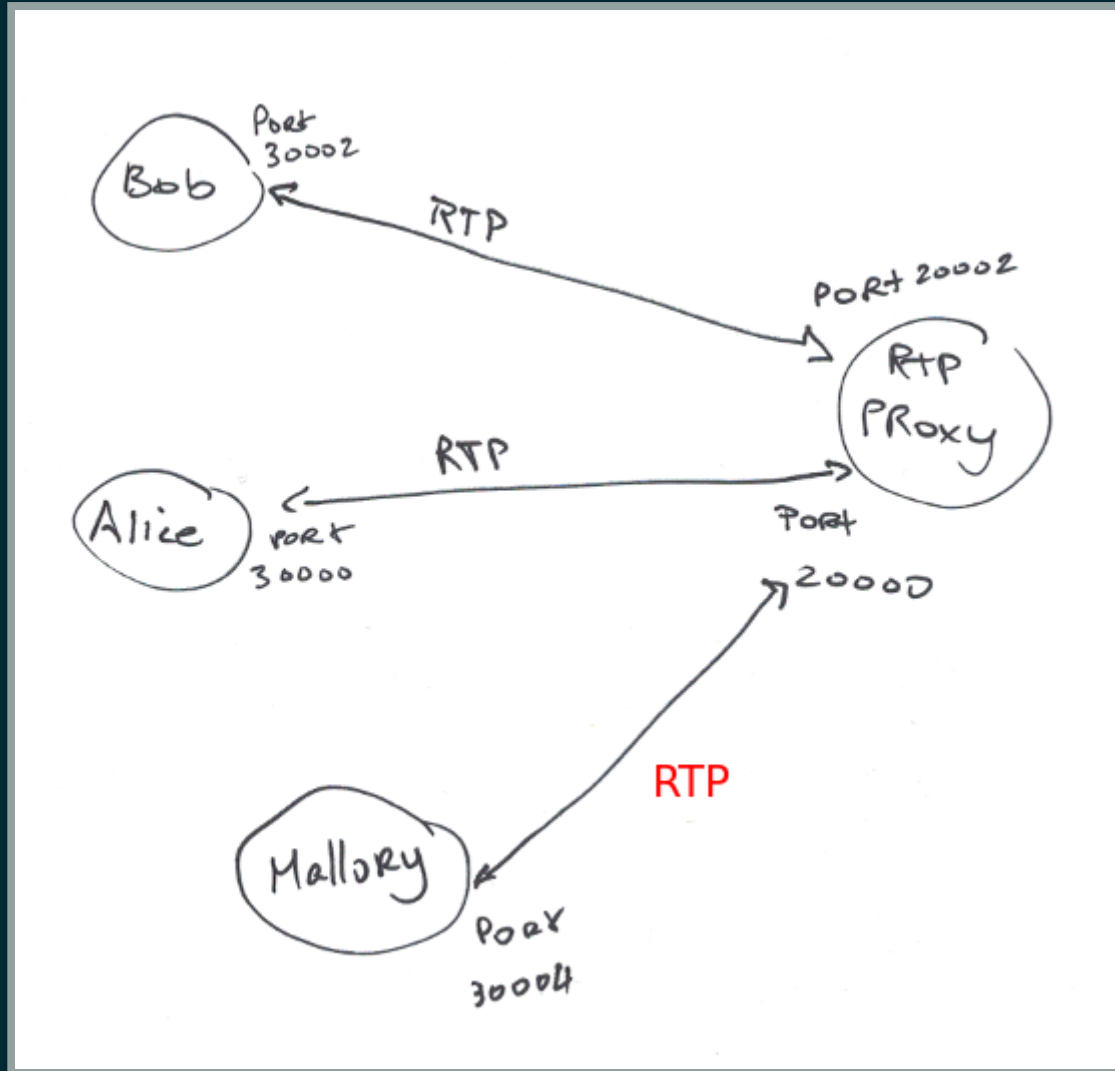
MEDIA

RTP BLEED, AGAIN?

Reminder: affects RTP proxies when they attempt to solve NAT



RTP Proxy normal functionality

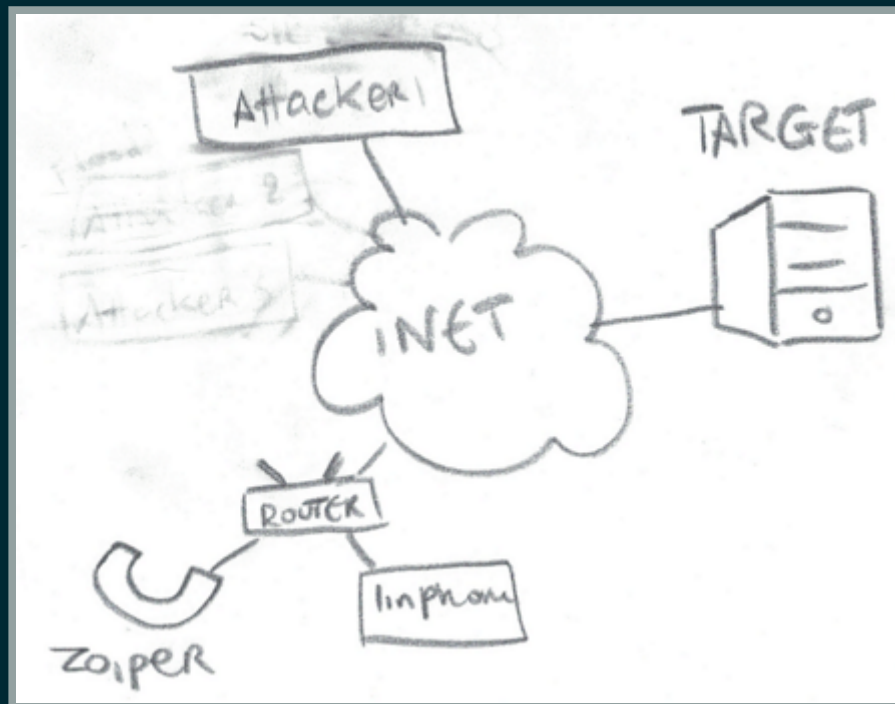


RTP Proxy under RTP Bleed attack

RTP BLEED = DOS?

- the RTP packets end up going to the attacker
- attacker can be very fast, winning the race condition every time
- rtp proxy locks to the attacker's IP
- the affect: call goes mute

SITEWIDE RTP BLEED: DEMO SETUP



SITEWIDE RTP BLEED: DEMO

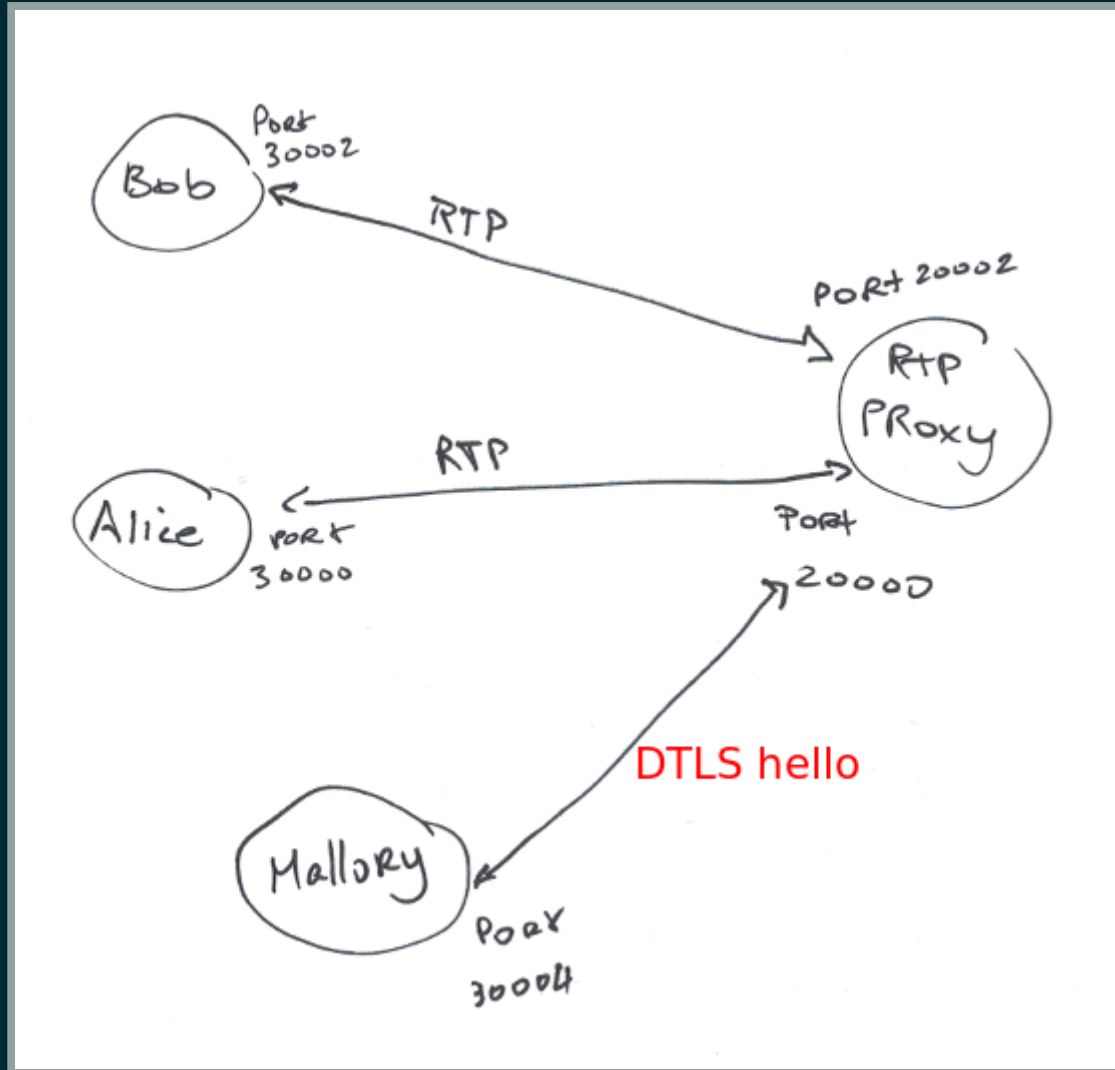


INVALID DTLS CERTIFICATE

- Proprietary RTP proxy terminating DTLS
- Logic: if wrong DTLS certificate is presented, that is a fatal error

INVALID DTLS CERTIFICATE

- RTP proxy was not checking the state of the DTLS session
- An attacker could spray DTLS hello packets on the RTP proxy and disconnect all ongoing calls



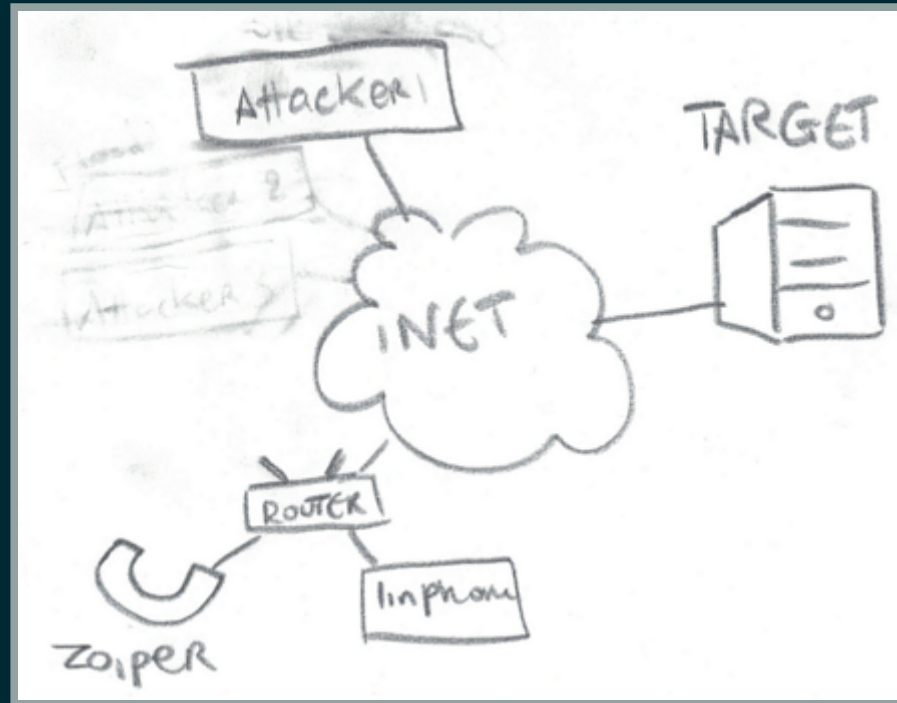
Mallory sends a DTLS hello

MONITORING TOOLS

RECORDING SYSTEMS

- recording phone calls seems harmless from a DoS perspective
- sending large amounts of RTP packets during a call in a short time

RECORDING SYSTEMS: DEMO SETUP



RECORDING SYSTEMS: DEMO



PCAP MONITORING

- Similar to the calling
- Trickier to exploit
- Often filtering only SIP traffic
- DoS by sending large SIP messages or large amounts
- SIP flooding may reproduce this issue

FLOODING THE FIREWALL

The firewall was configured to log all dropped packets..
something like the following:

```
iptables -A INPUT -j LOG --log-level info
```

Easily attacked as follows:

```
cat /dev/urandom | nc -u target 20000
```

CALLBACKS

SCENARIO

- Platforms that allow users (developers) feedback via callbacks
- e.g. An HTTP callback is triggered when a call is completed

RETURNING GARBAGE

```
nc -l 9999 < /dev/random
```

SLOWLORIS

SLOWLORIS

- HTTP/1.1 200 OK

SLOWLORIS

- HTTP/1.1 200 OK
- Slow: loris

SLOWLORIS

- HTTP/1.1 200 OK
- Slow: loris
- Slow: loris

SLOWLORIS

- HTTP/1.1 200 OK
- Slow: loris
- Slow: loris
- Slow: loris

SLOWLORIS

- HTTP/1.1 200 OK
- Slow: loris
- Slow: loris
- Slow: loris
- Slow: loris

SLOWLORIS

- HTTP/1.1 200 OK
- Slow: loris
- Slow: loris
- Slow: loris
- Slow: loris
- Slow: loris

SLOWLORIS

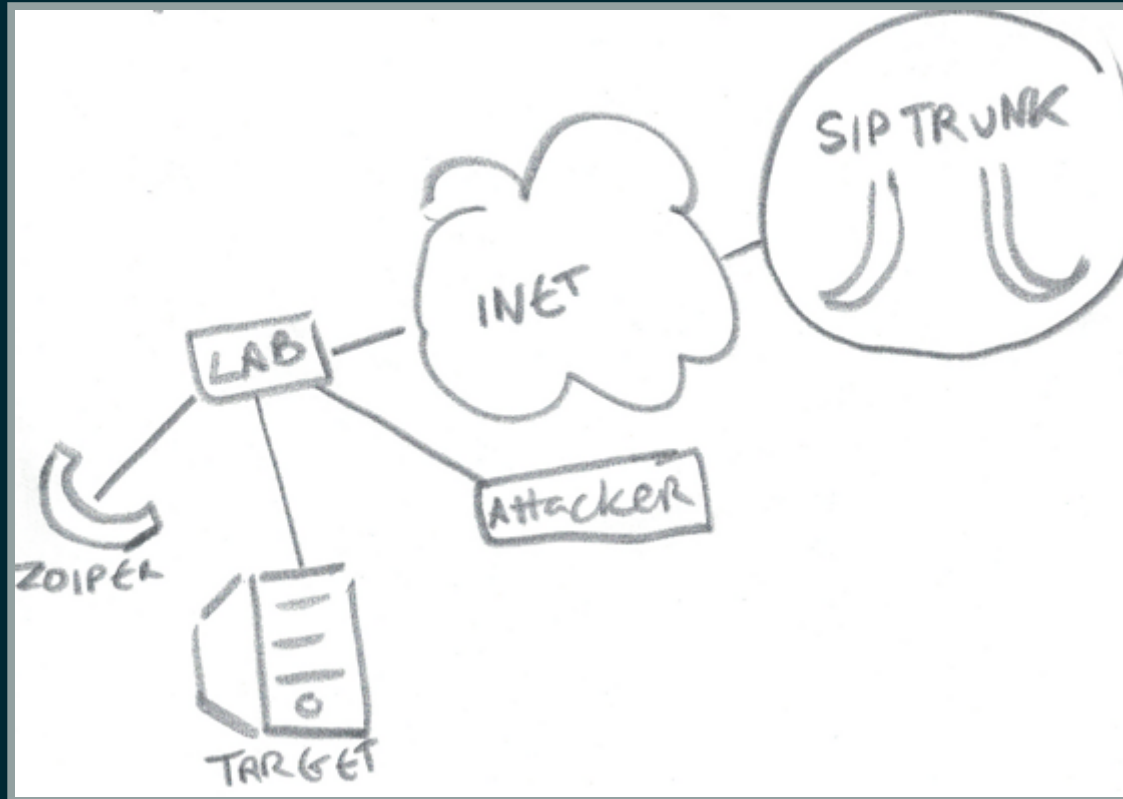
- HTTP/1.1 200 OK
- Slow: loris
- Slow: loris
- Slow: loris
- Slow: loris
- Slow: loris
- Slow: loris

SECURITY PROTECTION

IP SPOOFING

- an IPS (such as fail2ban) that simply blocks IP addresses indiscriminately
- attackers may spoof the source IP of a trusted peer, e.g. a SIP trunk

IP SPOOFING: DEMO SETUP



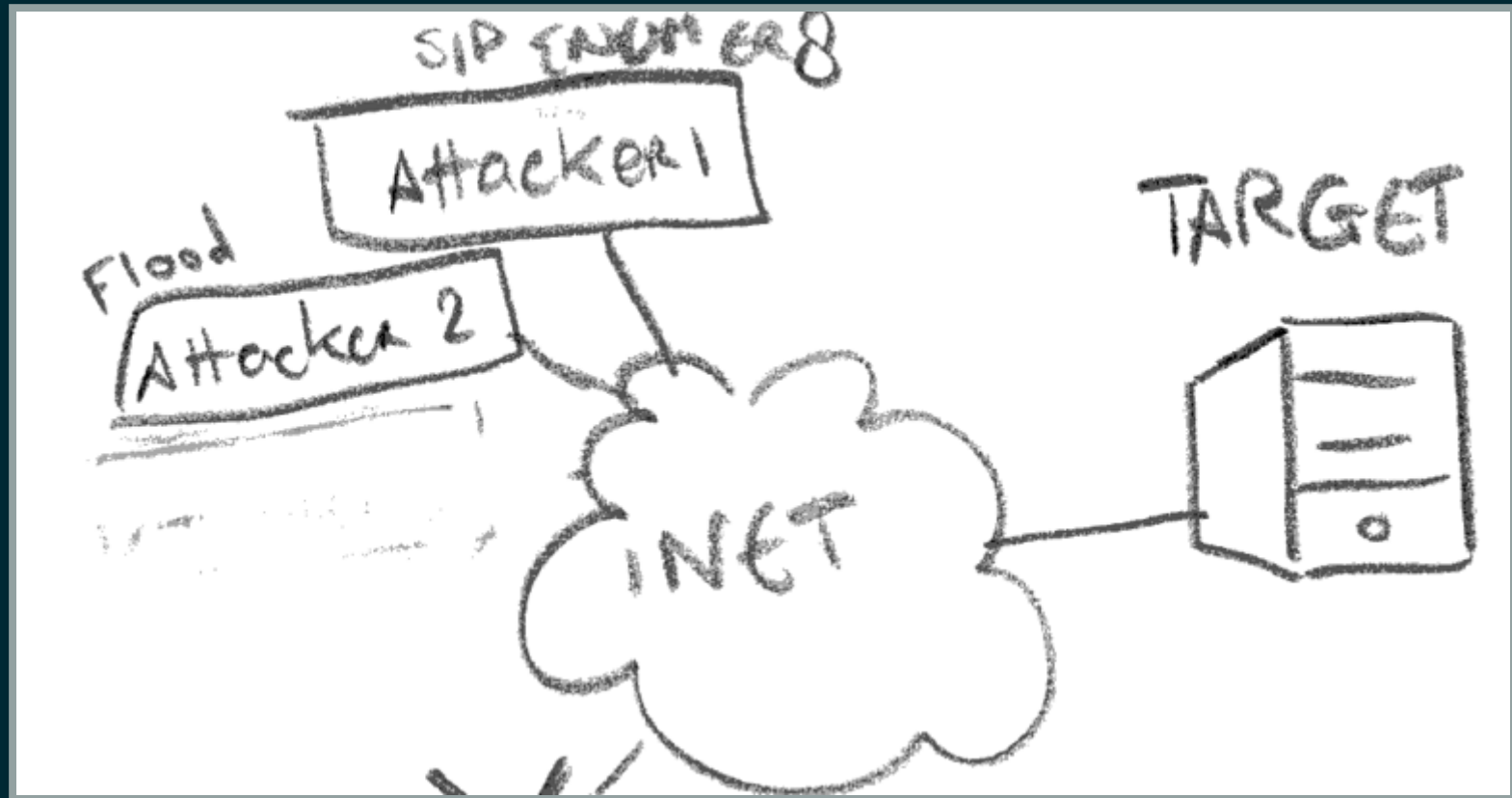
IP SPOOFING: DEMO



FLOODING THE IPS

- during an attack some systems generate a massive amount of logs
- IPS systems that rely on logs will try to keep up
- in the meantime, a real attack may be happening

FLOODING THE IPS: DEMO SETUP



FLOODING THE IPS: DEMO



EVASION

HOW DISTRIBUTED ATTACKS HELP

- Finding the right rate to avoid triggering rate limiting
- Distributing the attack, adding attacker nodes until the system is overwhelmed

SLOWING DOWN THE ATTACK

- Some attacks do not require speed
- Examples:
 - some memory leak issues
 - infinite loops

**SO HOW DO WE FIX THIS
STUFF?**

RATE LIMITING

- definitely a useful tool in your arsenal
- need to find the sweet spot
- keep in mind that distributed attacks are a thing

OVERZEALOUS MONITORING

don't

FUZZING (AGAIN?)

- Yes fuzzing is important
- Use ASAN / memory sanitizers
- weed out memory leaks/infinite loops/crashes

BUY MORE! (RESOURCES)

Can work but it is a cat and mouse game

SECURITY TESTING

- that's what we do :-)
- identify your bottlenecks
- DIY is useful

CONCLUSION

WE HAVE SEEN ...

- Flooding of SIP servers
- TCP / TLS connection flooding
- (SIP) TLS certificate authentication flooding
- Websocket custom protocol breakage
- RTP Bleed as DoS
- Invalid DTLS certificate hello packet as DoS
- Flooding the recording system with RTP

AND ALSO ...

- Flooding packet capturing
- Flooding the firewall's logging system
- Attacking callbacks
- Subverting the IPS to block a SIP trunk
- Flooding the IPS to run an attack in the meantime
- Tips on evasion
- A few thoughts on solutions

THANKS

- Alfred Farrugia for his help



SIPVicious PRO

SOME WAYS TO GET IN TOUCH

- sandro@enablesecurity.com
- Enable Security:
<https://enablesecurity.com/#contact-us>
- Subscribe for the private beta:
<https://sipvicious.pro>