CMSC 33250: Graduate Computer Security

Lecture 3: Denial of Service & Botnets

Grant Ho, Fall 2023

(Some slides borrowed & adapted from Vern Paxson and Frank Li)

Overview: Denial of Service

Key Security Property: Availability

Denial of Service: prevent users from accessing a service / server

- Typically, by deliberating causing network connection failures

Two common kinds of DoS approaches:

- Logic-based attacks: use misconfiguration/bugs to crash service
- Flooding (state-exhaustion): overwhelm victim's resources (CPU, memory, network bandwidth, etc.)

Successful Flooding Attacks

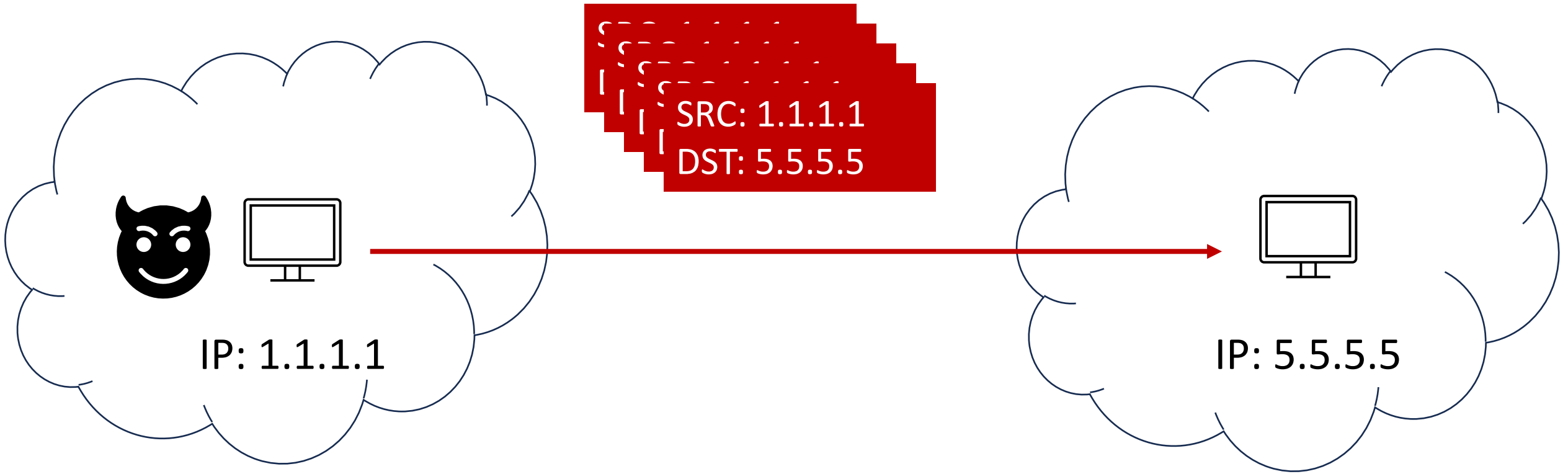
Need to overwhelm the victim's resources, ***without***:

1. Overwhelming yourself as well (DoS your own attack offline)
2. Providing easy mechanism to block/stop the attack

A few common strategies:

1. Distributed attacks via botnets
2. IP address spoofing + state asymmetry
3. Reflection & amplification

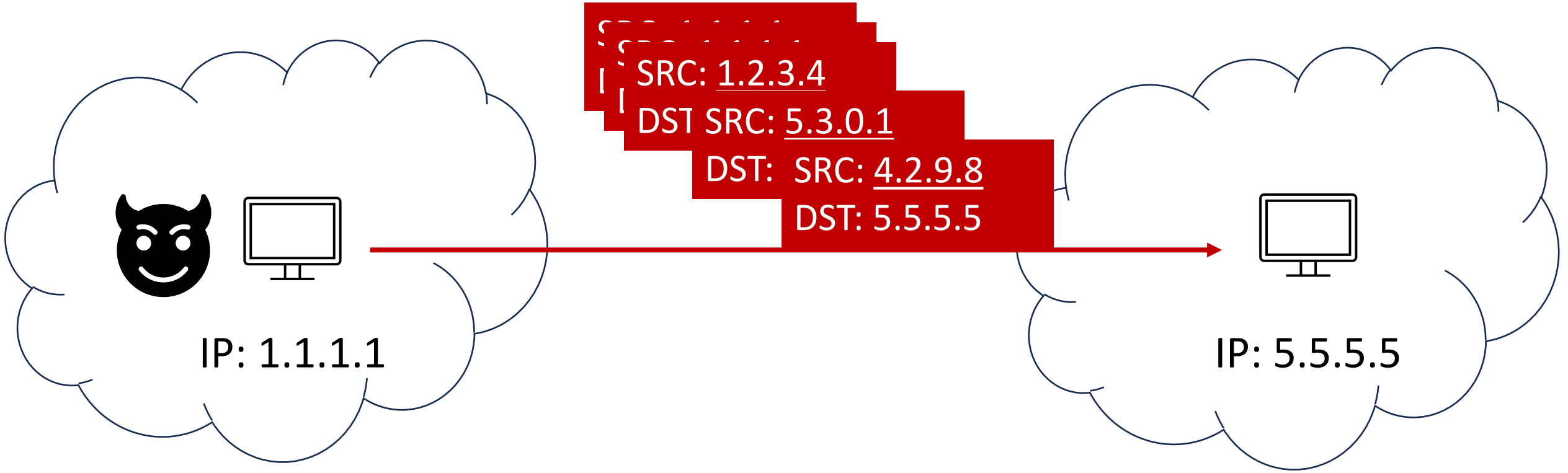
IP Address Spoofing: Benefits of Obfuscation



Trivial to Defend:

Block all packets from src = 1.1.1.1

IP Address Spoofing: Benefits of Obfuscation



Spoof random src IP addresses: hard to block & attack still works (don’t need/want any response in DoS)

Inferring Internet Denial-of-Service Activity

David Moore
CAIDA
San Diego Supercomputer Center
University of California, San Diego
dmoore@caida.org

Geoffrey M. Voelker and Stefan Savage
Department of Computer Science and Engineering
University of California, San Diego
{voelker,savage}@cs.ucsd.edu

Historical & Meta Context

Who are the authors?

Why are they writing this paper?

What style of paper is this?

Inferring Internet Denial-of-Service Activity

David Moore
CAIDA
San Diego Supercomputer Center
University of California, San Diego
dmoore@caida.org

Geoffrey M. Voelker and Stefan Savage
Department of Computer Science and Engineering
University of California, San Diego
{voelker,savage}@cs.ucsd.edu

The Problem & Motivation

The Problem: Measure the prevalence of DoS attacks

Why is this problem unsolved / technically challenging?

- How do you find victims / evidence of attacks?
- Hard to get data even if you know about an attack (e.g., private/sensitive)
- Limited visibility into *global* characteristics

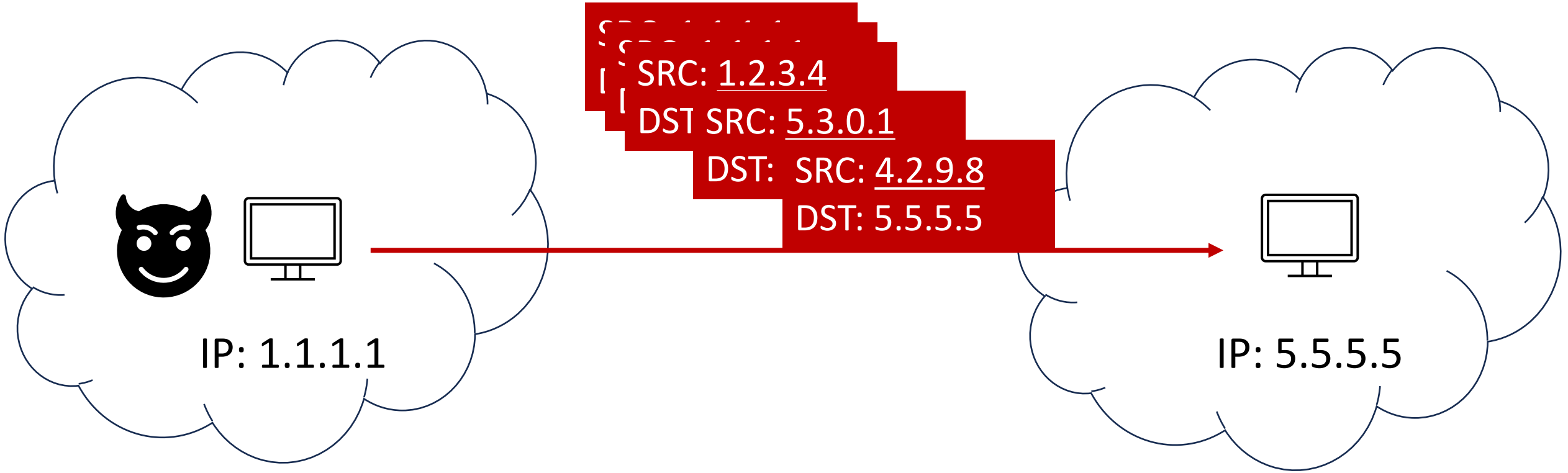
Background: Many Packets Solicit Recipient Response

- Networking protocols dictate how machines should respond when receiving certain packets
- Many kinds of attack packets cause the victim to send response packets
- Only makes sense to receive response packets if you prev sent specific packet (i.e., no unsolicited responses)

Packet sent	Response from victim
TCP SYN (to open port)	TCP SYN/ACK
TCP SYN (to closed port)	TCP RST (ACK)
TCP ACK	TCP RST (ACK)
TCP DATA	TCP RST (ACK)
TCP RST	no response
TCP NULL	TCP RST (ACK)
ICMP ECHO Request	ICMP Echo Reply
ICMP TS Request	ICMP TS Reply
UDP pkt (to open port)	protocol dependent
UDP pkt (to closed port)	ICMP Port Unreach
...	...

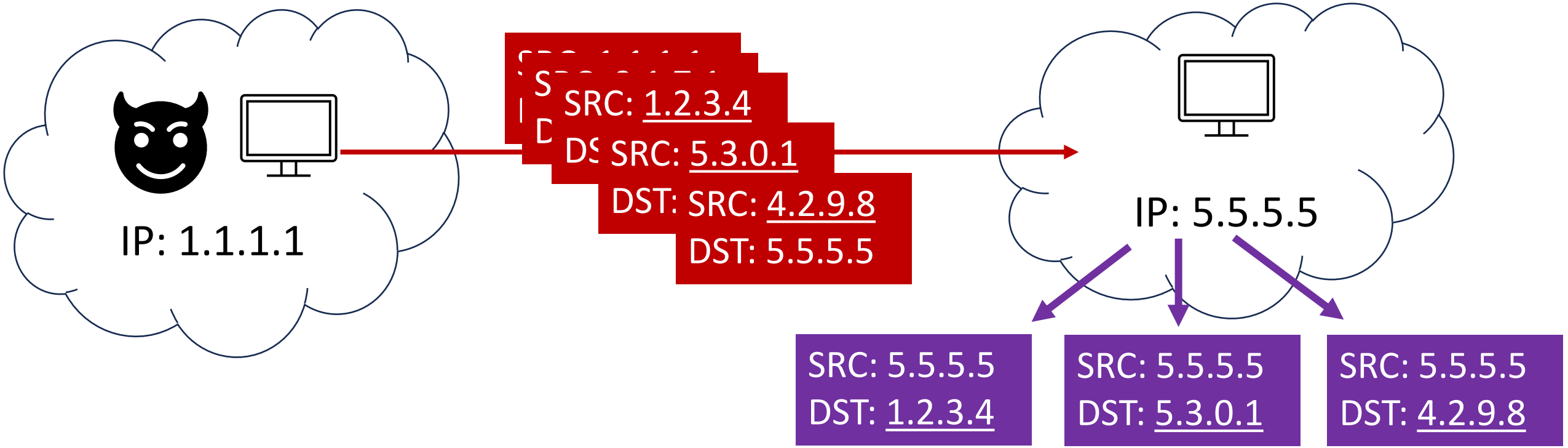
Table 1: A sample of victim responses to typical attacks.

Recall: Attackers Spoof src IP addresses



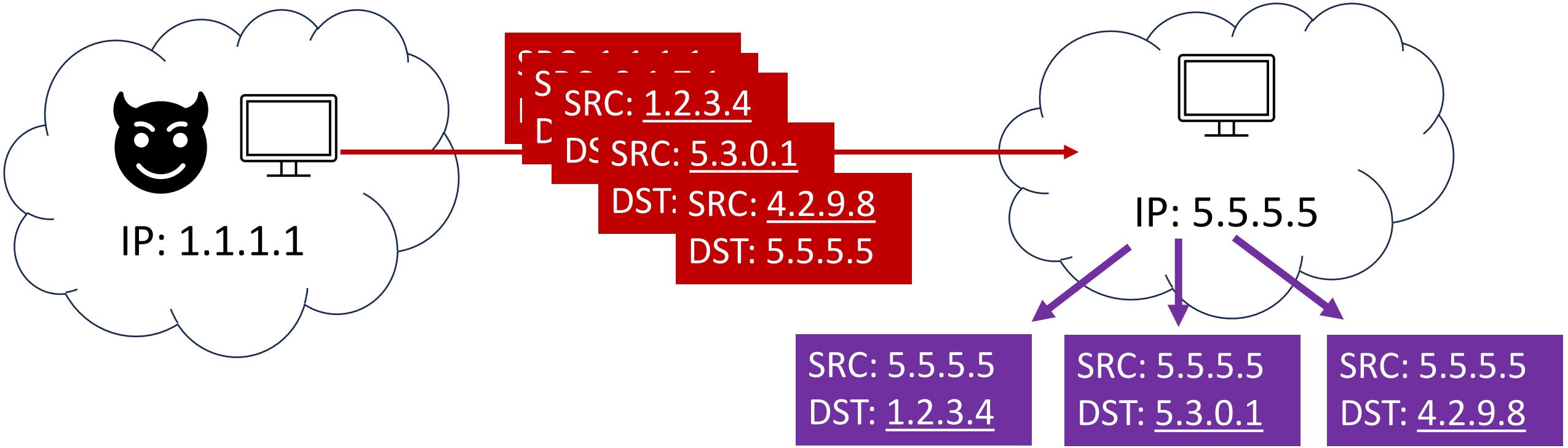
For (non-reflection) flooding attacks, attackers often spoof src IP addresses to prevent easy blocking

Key Idea: Backscatter Inference



Insight #1: Most attack packets trigger a reply packets ->
victim will send replies to the spoofed src addresses (***backscatter***)

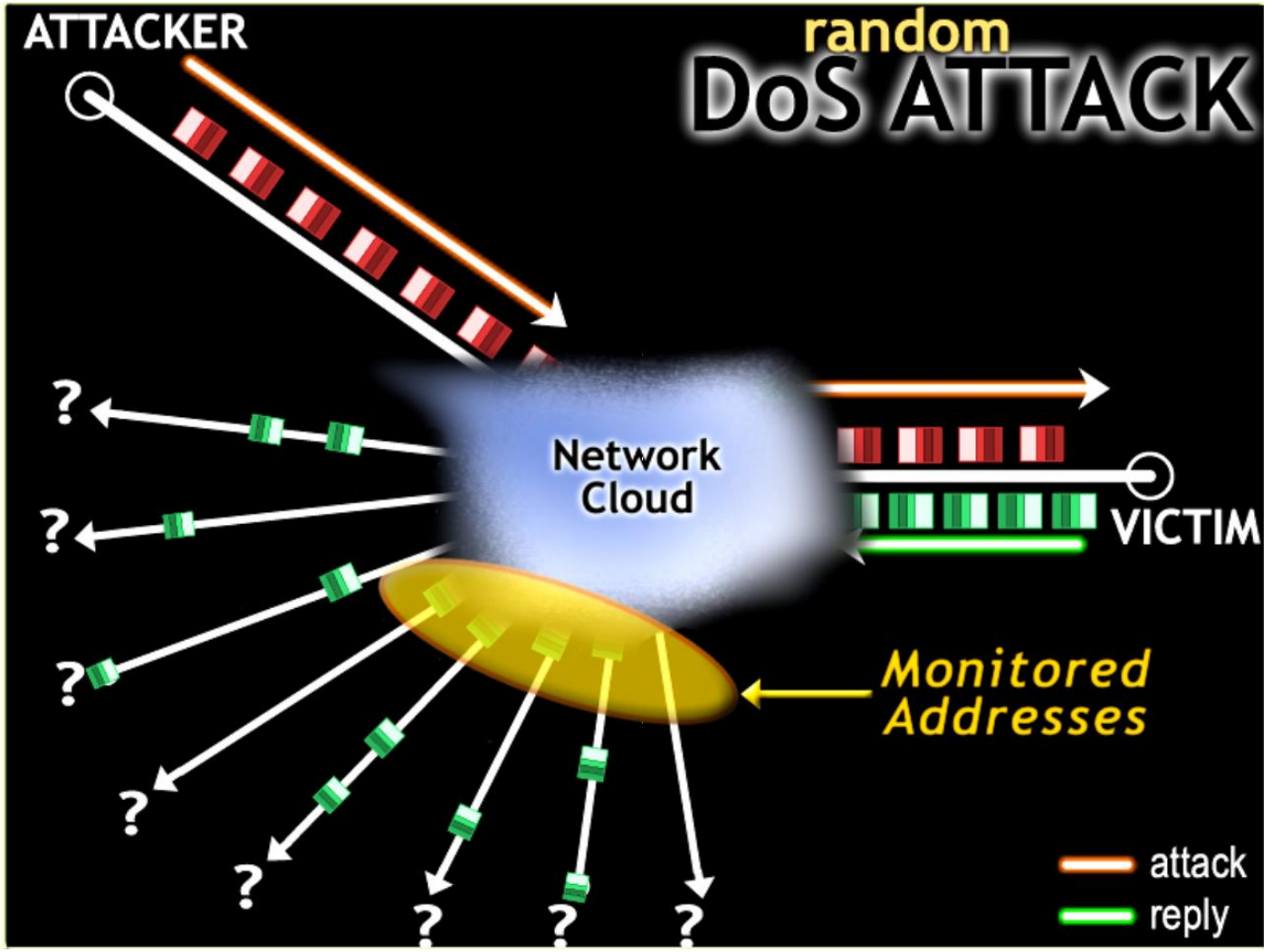
Key Idea: Backscatter Inference



Insight #2: Since most DoS programs ***select src IP addr at random***, any host on the Internet has an equi-probability of getting backscatter packets

Key Idea: Backscatter Inference

The approach: monitor inbound traffic to a ***large set of IP addresses***: will see backscatter packets from real-time DoS attacks w/ high probability



Key Idea: Backscatter Inference

Assuming attackers randomly spoof the src IP addr of attack packets:

Probability of one host seeing a given backscatter packet = $1 / 2^{32}$

- Total # of IPv4 addresses: 2^{32}

Expected # of backscatter packets, for M-packet attack: $M / 2^{32}$

- If we monitor N hosts: $(N * M) / 2^{32}$

Measurement Setup

Monitor all incoming traffic to a "/8" darknet

- i.e., 1 / 256 of entire IPv4 address space
- 3 week data collection
- Darknet = dormant IP address space (no active hosts) : should ***not*** receive any traffic

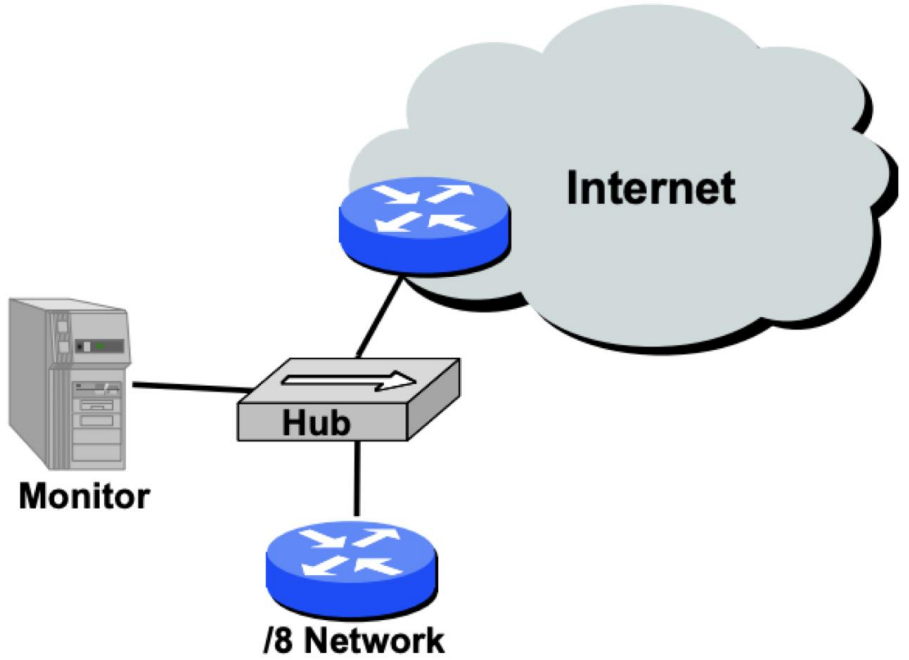


Figure 2: Our experimental backscatter collection platform. We monitor all traffic to our /8 network by passively monitoring data as it is forwarded through a shared hub. This monitoring point represents the only ingress into the network.

Data Analysis Methods

How do we know which backscatter packets belong to a single “attack”?

Two methods

Event-based: per victim IP, fixed time window:

Windows = 1-minute periods

Event occurs if victim emits 10+ backscatter packets/window

Flow-based: per victim IP, continues until 5-minute inactivity

Requires: 100+ packets, 60+ seconds, > 1 monitored addr.

Measurement Results: Attack & Victim Characteristics

- 12,805 attacks (flow-based)
- 200M backscatter packets
- Over 5k distinct victim IP addresses (resolving to over 2k domains)
  - Heuristics from domain names: 10-15% of victims home users

Kind	Trace-1			
	Attacks		Packets (k)	
TCP	3,902	(94)	28,705	(56)
UDP	99	(2.4)	66	(0.13)
ICMP	88	(2.1)	22,020	(43)
Proto 0	65	(1.6)	25	(0.05)
Other	19	(0.46)	12	(0.02)

Primarily TCP-based flooding attacks

Kind	Trace-1			
	Attacks		Packets (k)	
Multiple Ports	2,740	(66)	24,996	(49)
Uniformly Random	655	(16)	1,584	(3.1)
Other	267	(6.4)	994	(2.0)
Port Unknown	91	(2.2)	44	(0.09)
HTTP (80)	94	(2.3)	334	(0.66)
0	78	(1.9)	22,007	(43)
IRC (6667)	114	(2.7)	526	(1.0)
Authd (113)	34	(0.81)	49	(0.10)

No obvious port (service-indicator)

Measurement Results: Attack Duration

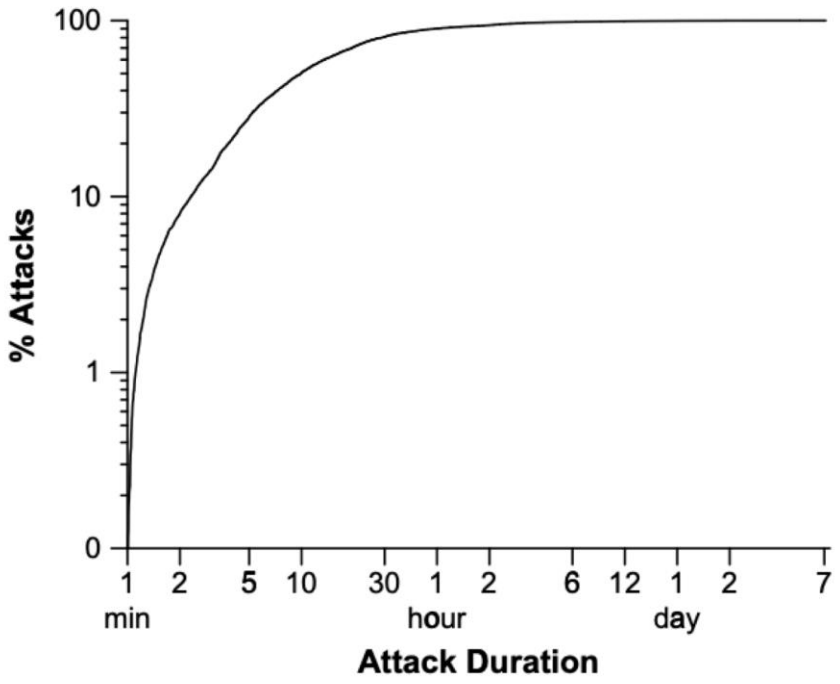


Figure 5: Cumulative distribution of attack durations.

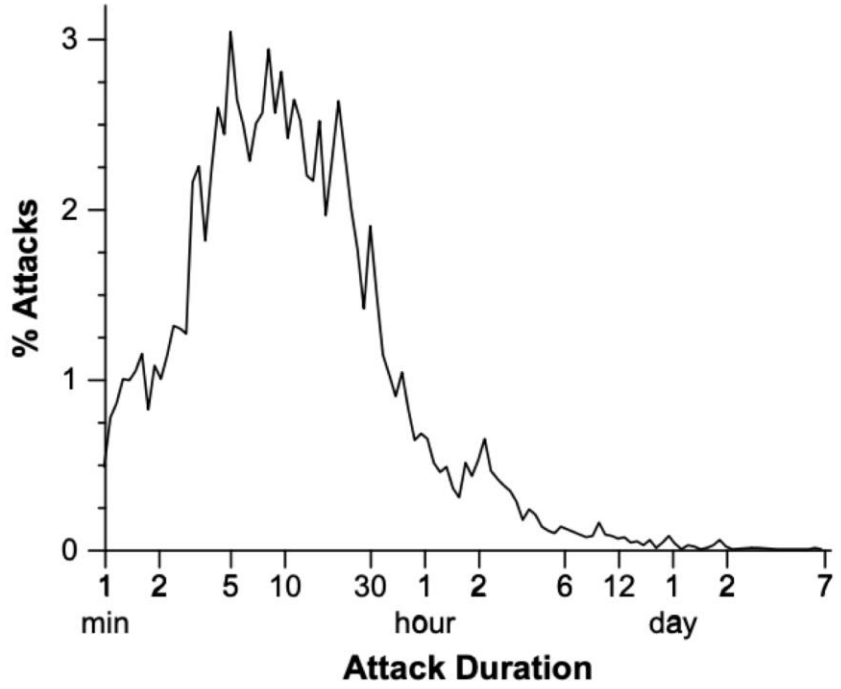


Figure 6: Probability density of attack durations.

Evaluation: Validating Assumptions

What are the key assumptions for the backscatter inference to work?

Evaluation: Validating Assumptions

What are the key assumptions for the backscatter inference to work?

1. Address Uniformity: spoofed src IP's chosen at random
2. Reliable Delivery: packets not dropped/slowed in delivery
3. Backscatter Hypothesis: unsolicited packets received by monitors are backscatter and not something else

Evaluation: Validating Assumptions

How are these assumptions validated?

1. Address Uniformity: spoofed src IP's chosen at random
2. Reliable Delivery: packets not dropped/slowed in delivery
3. Backscatter Hypothesis: unsolicited packets are DoS backscatter

Evaluation: Validating Assumptions

How are these assumptions validated?

1. Address Uniformity: spoofed src IP's chosen at random
   - Looked at DoS software/code, A2 stats testing within own data
2. Reliable Delivery: packets not dropped/slowed in delivery
   - Not validated; instead, logical argument: leads to underestimation
3. Backscatter Hypothesis: unsolicited packets are DoS backscatter
   - 80-90% of backscatter packets do ***not*** elicit reply (not probing/scanning)
   - Validate with external data/IP address space (98% victim IP overlap)

Additional Discussion

- Thoughts on core idea & validity today?
- How do you think DoS attacks / measurement results might differ?
- Unexplored characteristics / measurement results?
- What are some defenses against DoS attacks?

Understanding the Mirai Botnet

Manos Antonakakis[◇] Tim April[‡] Michael Bailey[†] Matthew Bernhard[◁] Elie Bursztein[○]
Jaime Cochran[▷] Zakir Durumeric[◁] J. Alex Halderman[◁] Luca Invernizzi[○]
Michalis Kallitsis[§] Deepak Kumar[†] Chaz Lever[◇] Zane Ma[†*] Joshua Mason[†]
Damian Menscher[○] Chad Seaman[‡] Nick Sullivan[▷] Kurt Thomas[○] Yi Zhou[†]

[‡]*Akamai Technologies* [▷]*Cloudflare* [◇]*Georgia Institute of Technology* [○]*Google*
[§]*Merit Network* [†]*University of Illinois Urbana-Champaign* [◁]*University of Michigan*

Historical & Meta Context

Who are the authors?

Why are they writing this paper?

What style of paper is this?

Understanding the Mirai Botnet

Manos Antonakakis[◇] Tim April[‡] Michael Bailey[†] Matthew Bernhard[◁] Elie Bursztein[○]
Jaime Cochran[▷] Zakir Durumeric[◁] J. Alex Halderman[◁] Luca Invernizzi[○]
Michalis Kallitsis[§] Deepak Kumar[†] Chaz Lever[◇] Zane Ma[†*] Joshua Mason[†]
Damian Menscher[○] Chad Seaman[‡] Nick Sullivan[▷] Kurt Thomas[○] Yi Zhou[†]

[‡]*Akamai Technologies* [▷]*Cloudflare* [◇]*Georgia Institute of Technology* [○]*Google*
[§]*Merit Network* [†]*University of Illinois Urbana-Champaign* [◁]*University of Michigan*

Background

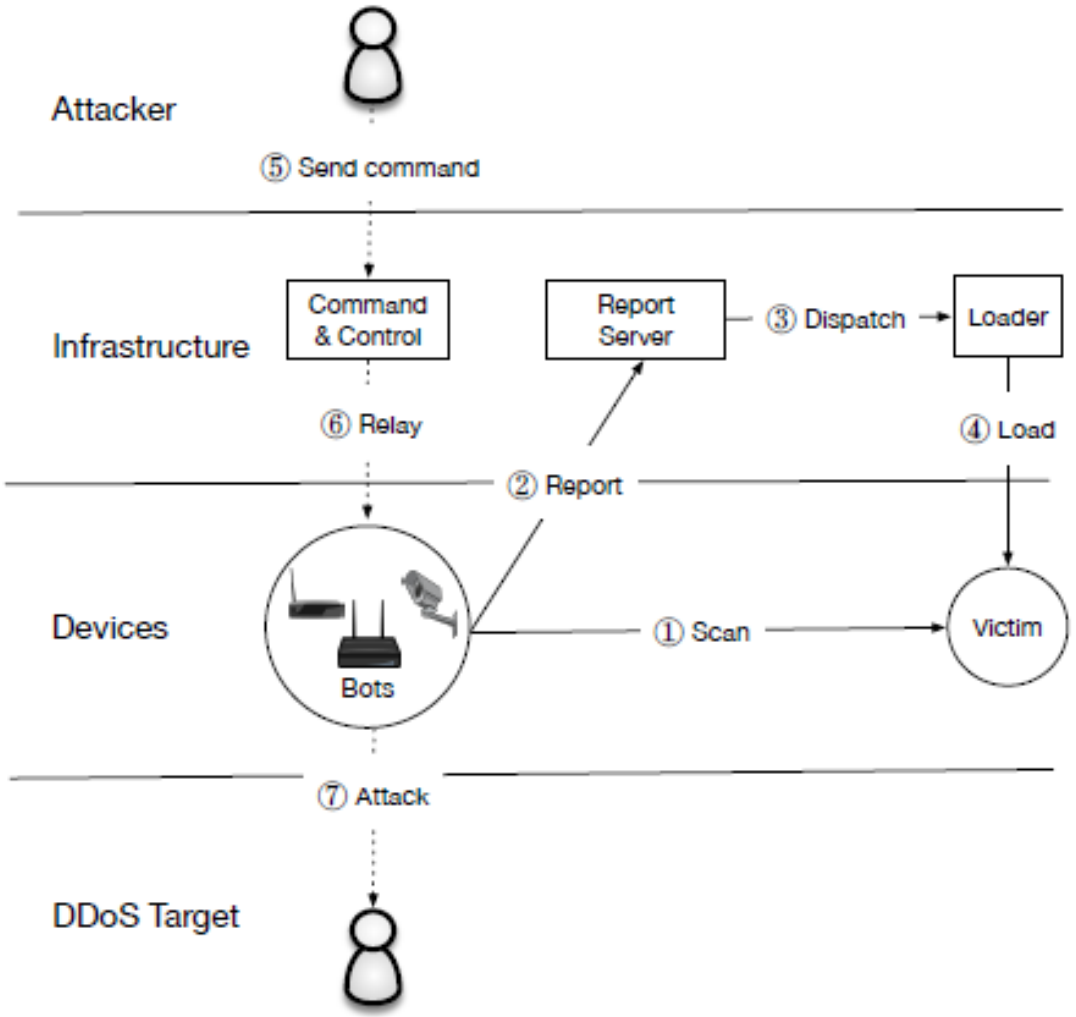
- **Mirai:** Worm-Malware family used to create large botnet that launched massive DDoS attacks

- **Internet of Things (IoT):** sea of cyber-physical objects (sensors/hardware in physical objects) that can connect/transmit data
  - “Smart objects” : TVs, thermostats, fridges, etc.
  - Wearable devices: smart watches, jewelry, clothing, etc.
  - Notoriously poor security practices

- **Goal of paper:** characterize technical aspects & history of Mirai botnet, and its implications for IoT security going forward.

Background: Mirai Lifecycle

- Tries to connect to random IP addresses w/ telnet or SSH & 10 default user/pwd's
- If successful, report victim IP address & login creds
- Infect device with Mirai malware & evasion + persistence
- Listen for remote (C2) commands and execute commands (e.g., DDoS)
- Process repeats on all new/old infected devices

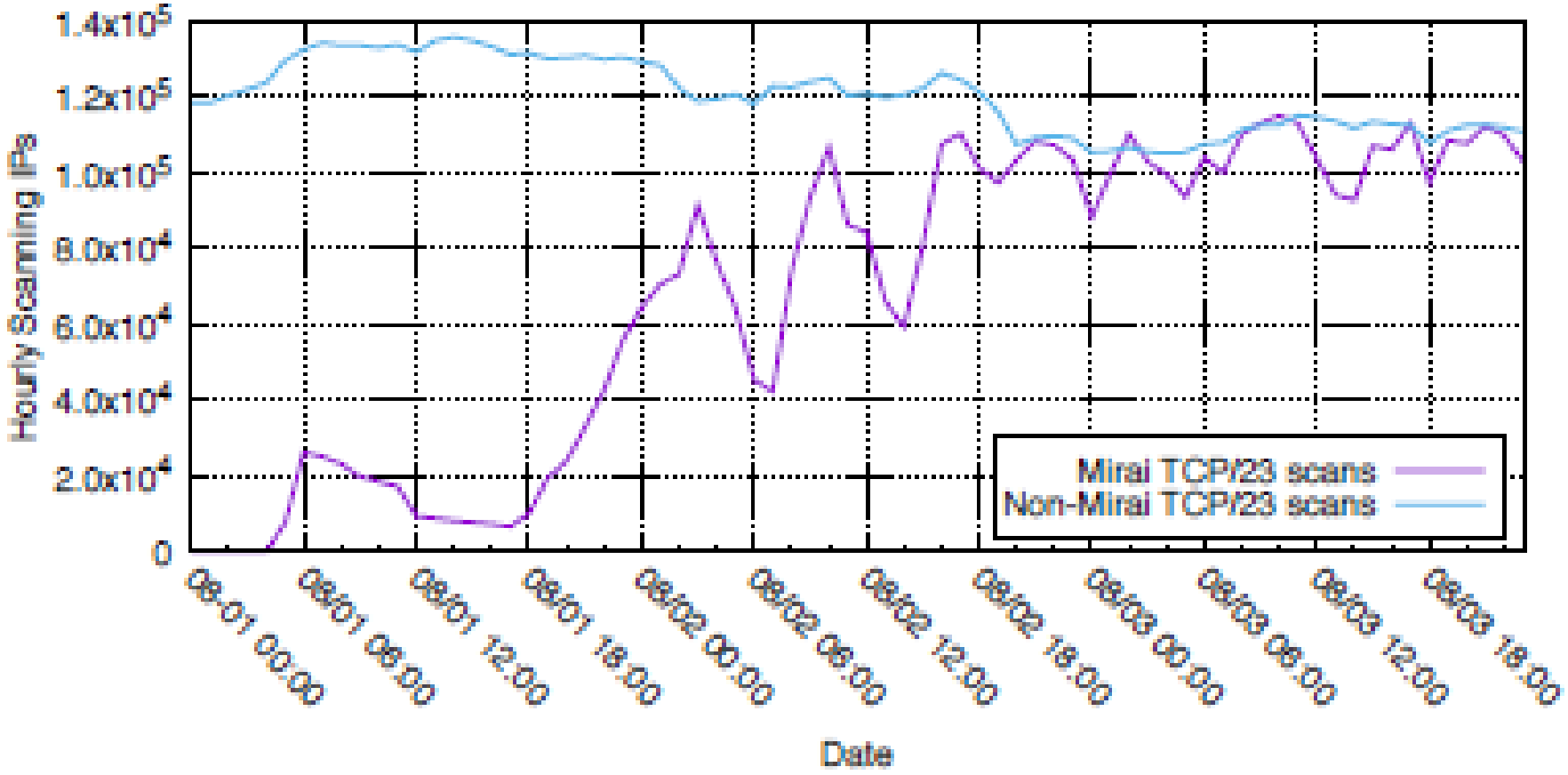


Data & Methodology

Source code of Mirai malware + Many different datasets

Role	Data Source	Collection Site	Collection Period	Data Volume
Growth and size	Network telescope	Merit Network, Inc.	07/18/2016–02/28/2017	370B packets, avg. 269K IPs/min
Device composition	Active scanning	Censys	07/19/2016–02/28/2017	136 IPv4 scans, 5 protocols
Ownership & evolution	Telnet honeypots	AWS EC2	11/02/2016–02/28/2017	141 binaries
	Telnet honeypots	Akamai	11/10/2016–02/13/2017	293 binaries
	Malware repository	VirusTotal	05/24/2016–01/30/2017	594 binaries
	DNS—active	Georgia Tech	08/01/2016–02/28/2017	290M RRs/day
	DNS—passive	Large U.S. ISP	08/01/2016–02/28/2017	209M RRs/day
Attack characterization	C2 milkers	Akamai	09/27/2016–02/28/2017	64.0K attack commands
	DDoS IP addresses	Akamai	09/21/2016	12.3K IP addresses
	DDoS IP addresses	Google Shield	09/25/2016	158.8K IP addresses
	DDoS IP addresses	Dyn	10/21/2016	107.5K IP addresses

Results: Scale & Growth



Rapid growth: 20hrs after first scan & brute forcing, Mirai goes from 1 -> 64,500 infected devices (IP addresses)

Within 1 mon: 200k-300k infected machines

Results: Is anecdotal claim about Mirai = IoT focus true?

Anecdotal claims prior to paper that Mirai is IoT focused botnet

- How would we validate this claim?

Results: Is anecdotal claim about Mirai = IoT focus true?

Password	Device Type	Password	Device Type	Password	Device Type
123456	ACTi IP Camera	klv1234	HiSilicon IP Camera	1111	Xerox Printer
anko	ANKO Products DVR	jvbzd	HiSilicon IP Camera	Zte521	ZTE Router
pass	Axis IP Camera	admin	IPX-DDK Network Camera	1234	Unknown
888888	Dahua DVR	system	IQinVision Cameras	12345	Unknown
666666	Dahua DVR	meinsm	Mobotix Network Camera	admin1234	Unknown
vizxv	Dahua IP Camera	54321	Packet8 VOIP Phone	default	Unknown
7ujMko0vizxv	Dahua IP Camera	00000000	Panasonic Printer	fucker	Unknown
7ujMko0admin	Dahua IP Camera	realtek	RealTek Routers	guest	Unknown
666666	Dahua IP Camera	1111111	Samsung IP Camera	password	Unknown
dreambox	Dreambox TV Receiver	xmhdipc	Shenzhen Anran Camera	root	Unknown
juantech	Guangzhou Juan Optical	smcadmin	SMC Routers	service	Unknown
xc3511	H.264 Chinese DVR	ikwb	Toshiba Network Camera	support	Unknown
OxhlwSG8	HiSilicon IP Camera	ubnt	Ubiquiti AirOS Router	tech	Unknown
cat1029	HiSilicon IP Camera	supervisor	VideoIQ	user	Unknown
hi3518	HiSilicon IP Camera	<none>	Vivotek IP Camera	zlxx.	Unknown
klv123	HiSilicon IP Camera				

Table 5: **Default Passwords**—The 09/30/2016 Mirai source release included 46 unique passwords, some of which were traceable to

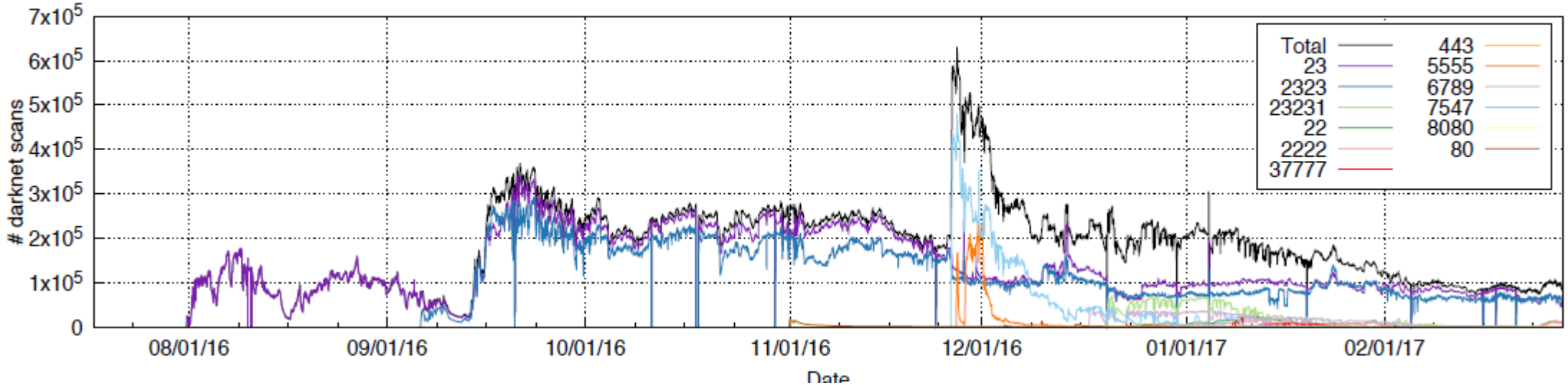
CWMP (28.30%)		Telnet (26.44%)		HTTPS (19.13%)		FTP (17.82%)		SSH (8.31%)	
Router	4.7%	Router	17.4%	Camera/DVR	36.8%	Router	49.5%	Router	4.0%
		Camera/DVR	9.4%	Router	6.3%	Storage	1.0%	Storage	0.2%
				Storage	0.2%	Camera/DVR	0.4%	Firewall	0.2%
				Firewall	0.1%	Media	0.1%	Security	0.1%
Other	0.0%	Other	0.1%	Other	0.2%	Other	0.0%	Other	0.0%
Unknown	95.3%	Unknown	73.1%	Unknown	56.4%	Unknown	49.0%	Unknown	95.6%

Table 6: **Top Mirai Device Types**—We list the top types of infected devices labeled by active scanning, as a fraction of Mirai banners found in Censys. Our data suggests that consumer routers, cameras, and DVRs were the most prevalent identifiable devices.

- Manual analysis to match brute-force password dictionaries (from malware source code) to default device credentials
- Analyze active scan (Censys) data of infected devices to determine device type

Results: Ownership / Attribution

- Mirai's source code is publicly released on Sep 2016: allows for anyone to modify & deploy their own variants



Results: Ownership / Attribution

How can we infer which Mirai-infected devices belong to different cybercrime groups (or at least use different variants of the malware)?

Results: Ownership / Attribution

How can we infer which Mirai-infected devices belong to different cybercrime groups (or at least use different variants of the malware)?

- Cluster based on C2 infrastructure
- Cluster based on malware behavior (binaries from honeypots / VT)
- Cluster based on scanning/brute force behavior (password dictionary)

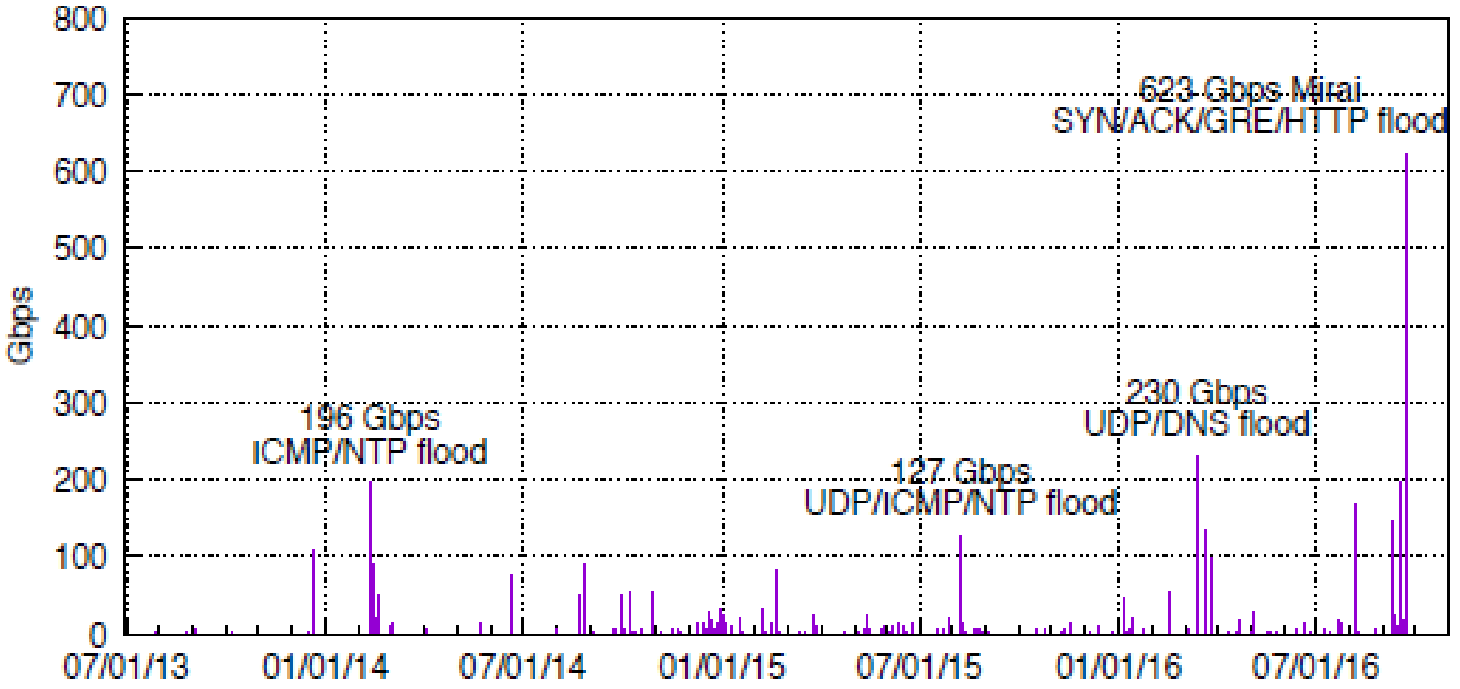
Results: Attack Victims

Target	Attacks	Cluster	Notes
Lonestar Cell	616	2	Liberian telecom targeted by 102 reflection attacks.
Sky Network	318	15, 26, 6	Brazilian Minecraft servers hosted in Psychz Networks data centers.
1.1.1.1	236	1,6,7,11,15,27,28,30	Test endpoint. Subject to all attack types.
104.85.165.1	192	1,2,6,8,11,15,21,23,26,27,28,30	Unknown router in Akamai's AS.
feseli.com	157	7	Russian cooking blog.
minomortaruolo.it	157	7	Italian politician site.
Voxility hosted C2	106	1,2,6,7,15,26,27,28,30	C2 domain from DNS expansion. Exists in cluster 2 seen in Table 8.
Tuidang websites	100	—	HTTP attacks on two Chinese political dissidence sites.
execrypt.com	96	—	Binary obfuscation service.
auktionshilfe.info	85	2,13	Russian auction site.
houtai.longqikeji.com	85	25	SYN attacks on a former game commerce site.
Runescape	73	—	World 26 of a popular online game.
184.84.240.54	72	1,10,11,15,27,28,30	Unknown target hosted at Akamai.
antiddos.solutions	71	—	AntiDDoS service offered at `react.su`.

Table 10: **Mirai DDoS Targets**—The top 14 victims most frequently targeted by Mirai run a variety of services. Online games, a

Results: Attack Capability & Impact

- One of largest DoS attacks by volume
- Prominent targets: Krebs on Security, Dyn (DNS provider: Amazon, Netflix, Github, etc.)
  - Broke load-distribution protection (Akami)
- Dyn collateral for gaming DDoS: risks of centralization & shared infra?



Implications & Recommendations for Future

- Security hardening: basic software/networking/application practices
- Automatic updating & patching
  - Incentives & End-of-life concerns?
- Vulnerability notification
  - Challenges?
- Network-level device identification
  - Risks?

Next Class

Read & Respond to Applied Crypto (SSL/TLS) Papers

Paper Presenter / Lead Signups: Posted on Canvas at 5:00pm

- Presentations start next class
- Click Trajectories & Backscatter good examples (less fancy + fewer details fine)
- 20min content and additional 10min discussion
  - Describe problem (research questions) + technical background from paper
  - Key methodology (dataset/collection and/or new technique/system)
  - Evaluation / analysis procedure
  - Key results (takeaways or evaluation performance)
  - Limitations / future work