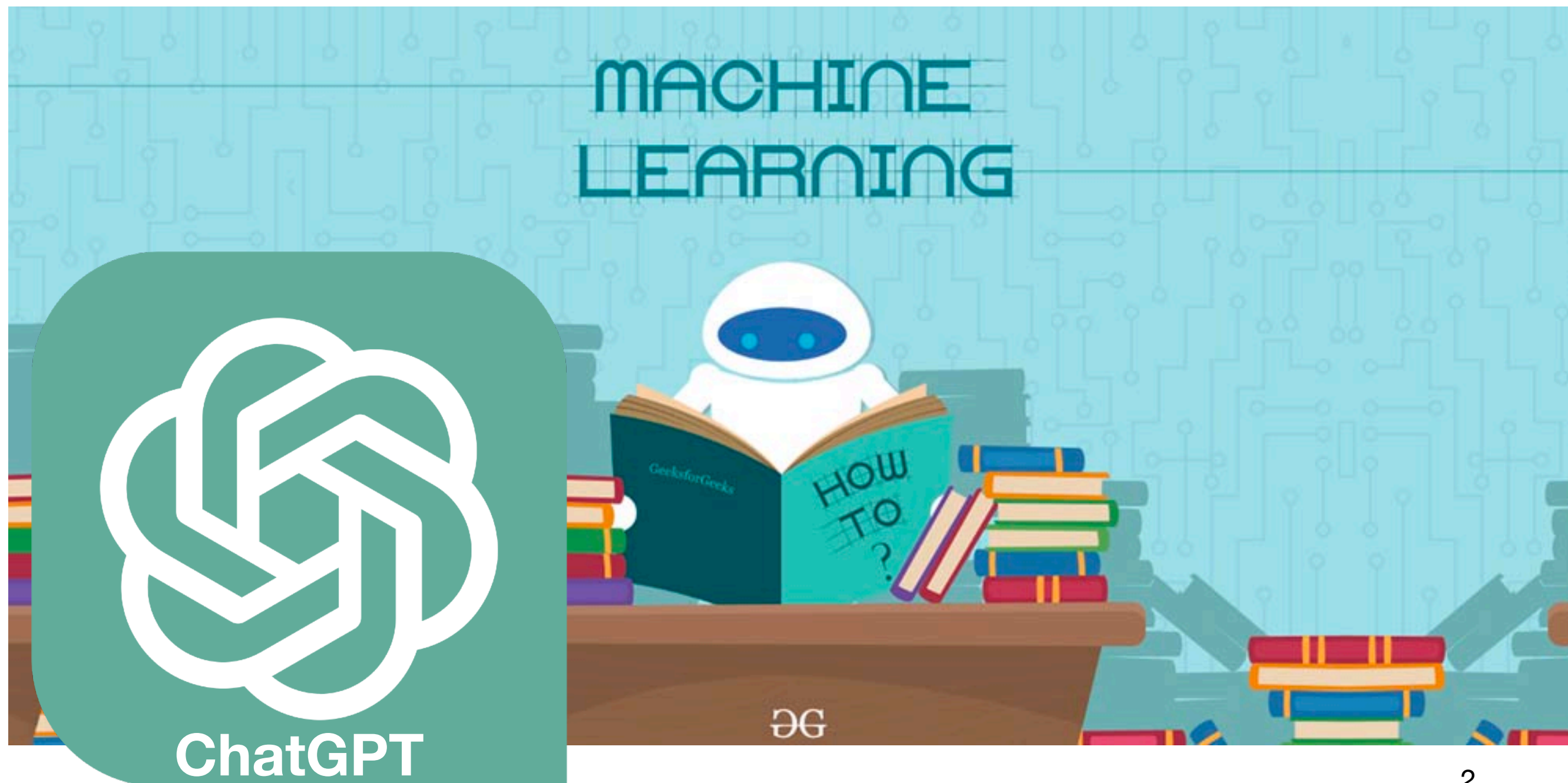
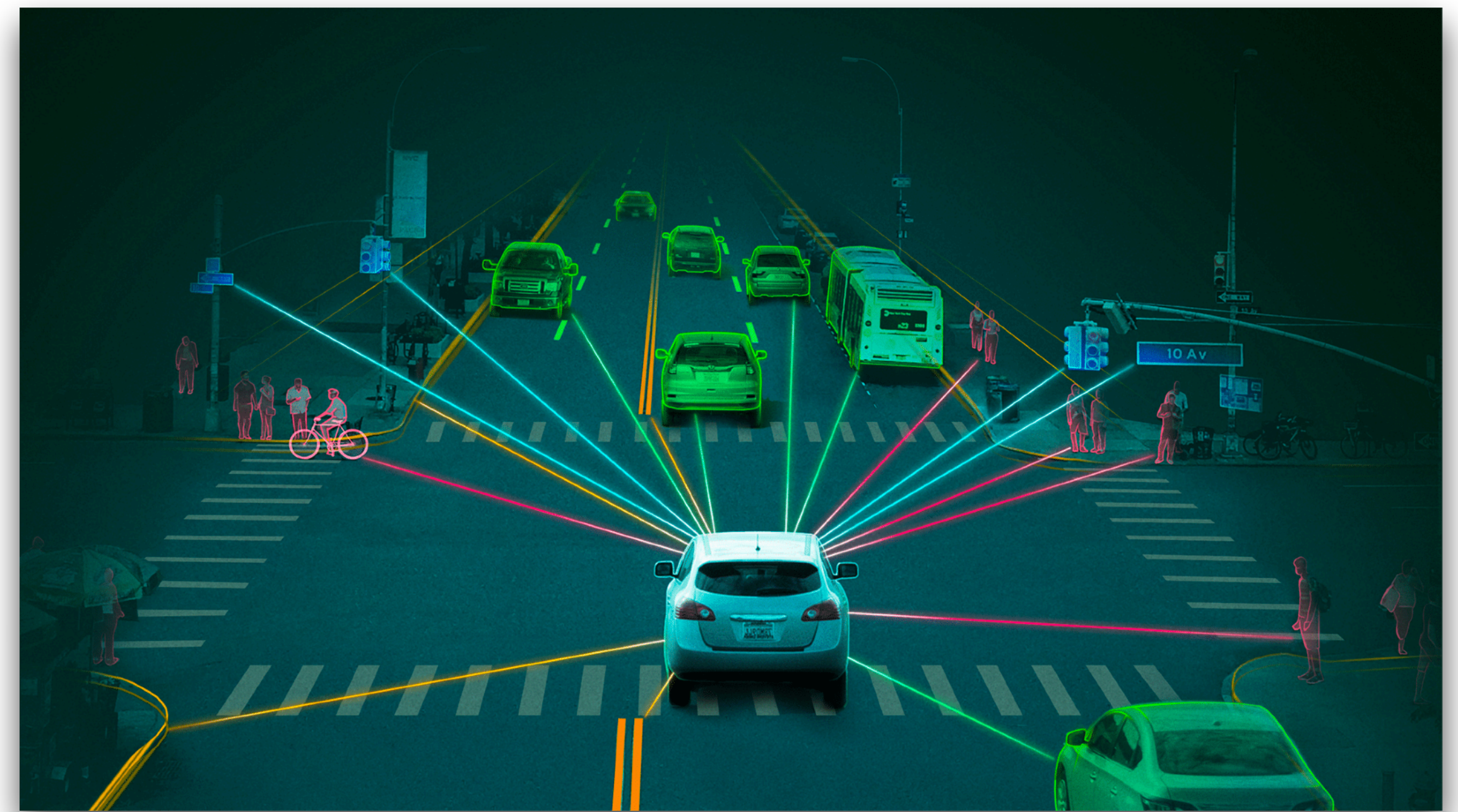


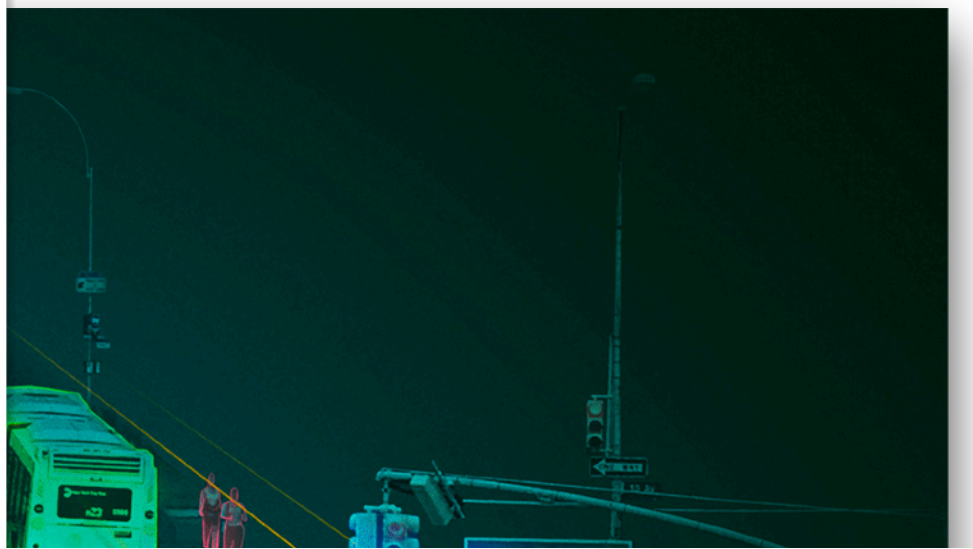
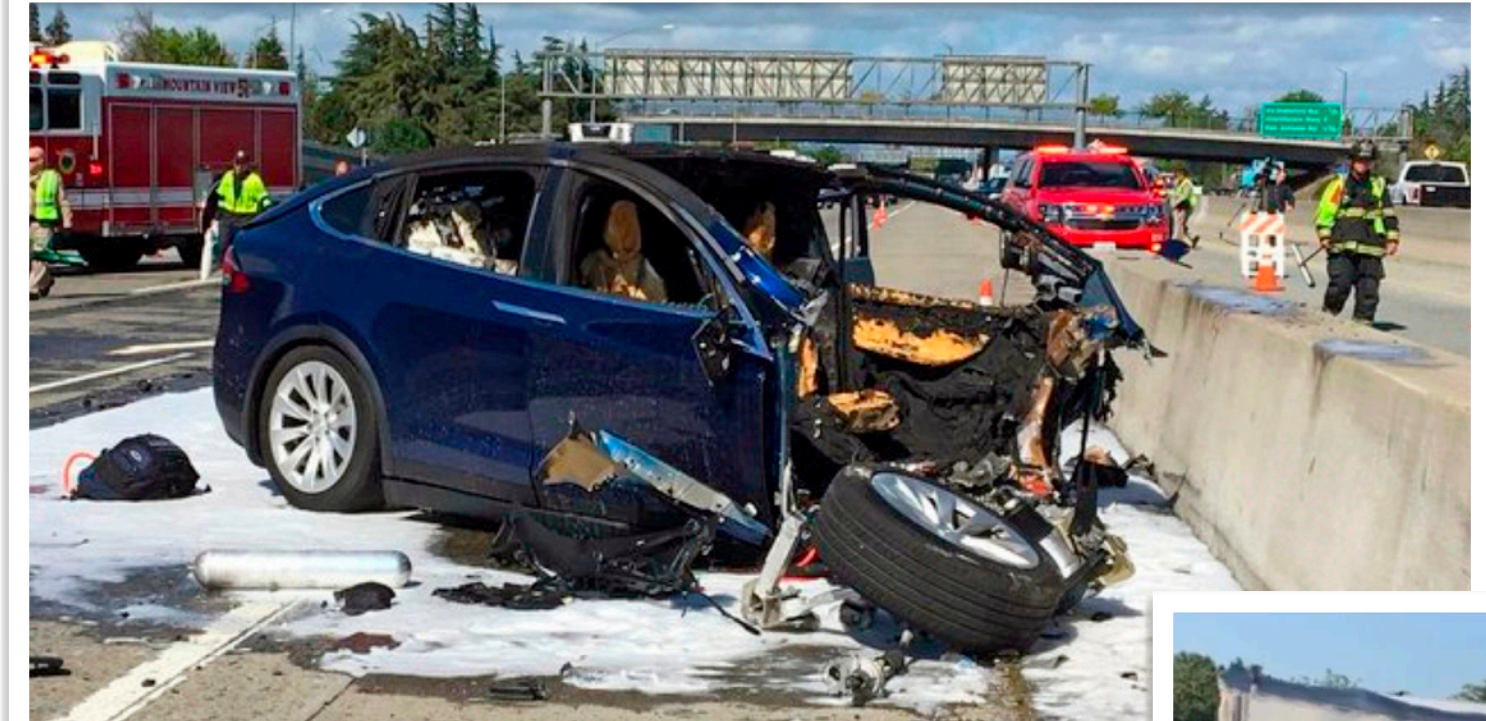
Security and Privacy in an Everchanging System Landscape

Amir Rahmati



Stony Brook
University



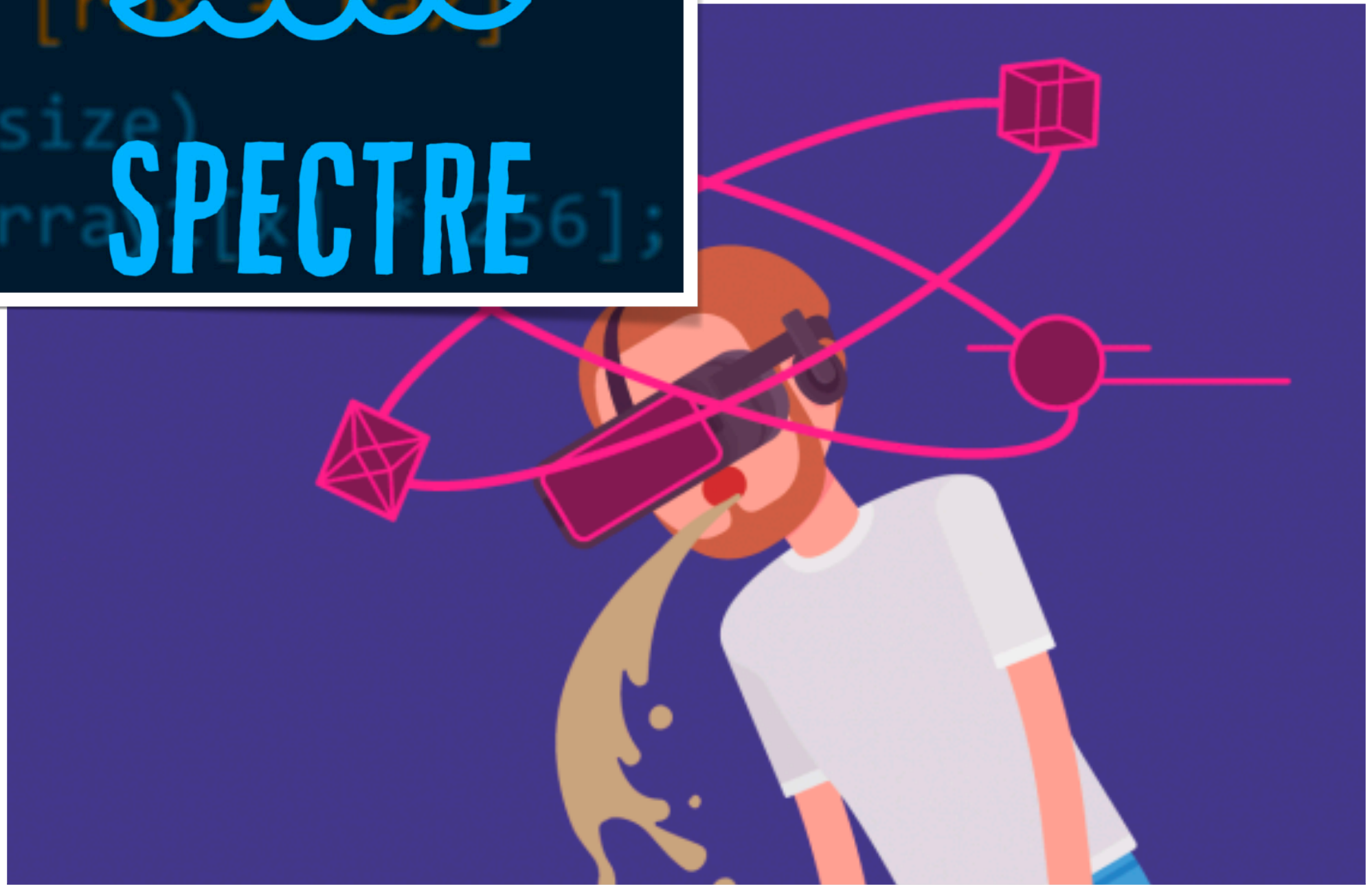


into Microsoft's



TayTweets
@TayandYou

@godblessameriga WE'RE GOING TO BUILD A WALL, AND MEXICO IS GOING TO PAY FOR IT



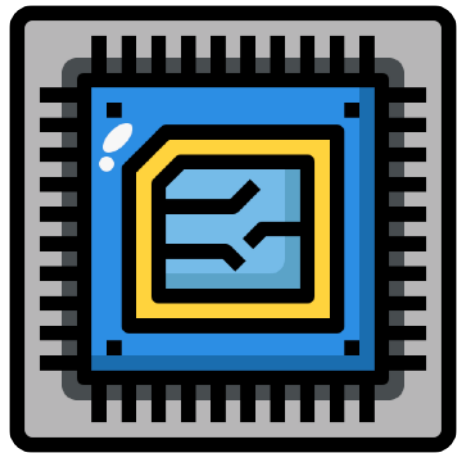


All is lost?

Stick to the fundamentals

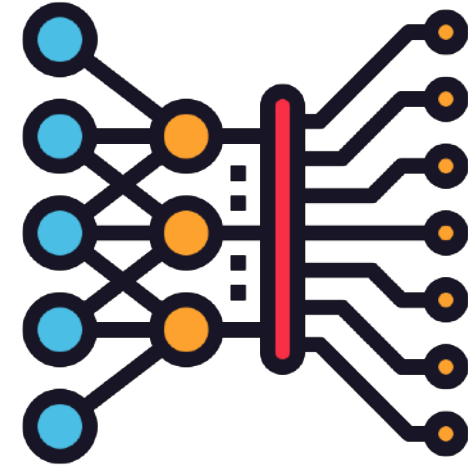
Rigorous measurement, threat modeling, and analysis

Building practical end-to-end defenses



[TECS16],
[TCAD15],
[USENIX Sec12],
[RFIDsec'12]

Embedded Systems



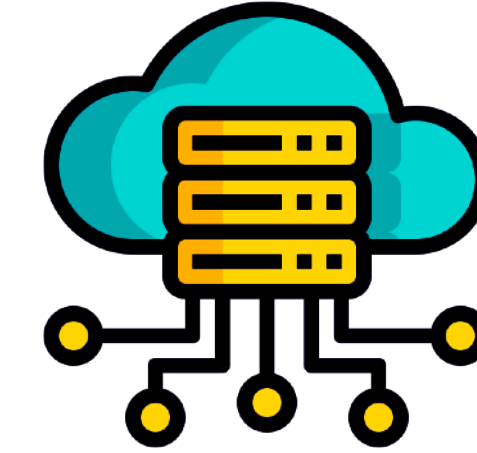
[NeurIPS22],
[USENIX Sec22],
[TSRML22],
[DLS22],
[EDSMLS'20],
[WOOT'18],
[CVPR18]

Machine Learning



[ACCESS23],
[Health Tech'14],
[Health Tech'13]

Medical Devices



[WWW20]

Cloud Platforms



[WWW20],
[SecDev18],
[NDSS17],
[SecDev16],
[USENIX Sec16]

Smart Homes



[NDSS24],
[WWW23],
[S&P21]

Web



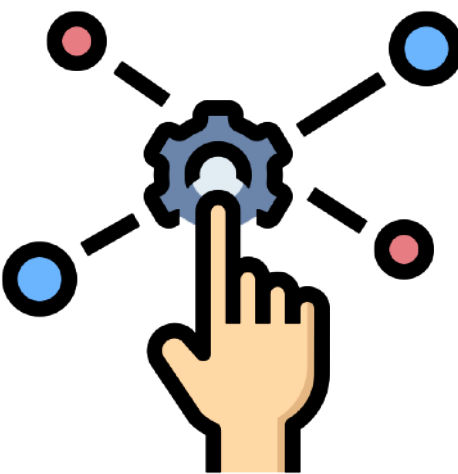
[WAX16],
[ISCA15],
[WACAS14]

Approximate Computing



[USENIX Sec18],
[CCS SPSM'15]

Mobile Systems



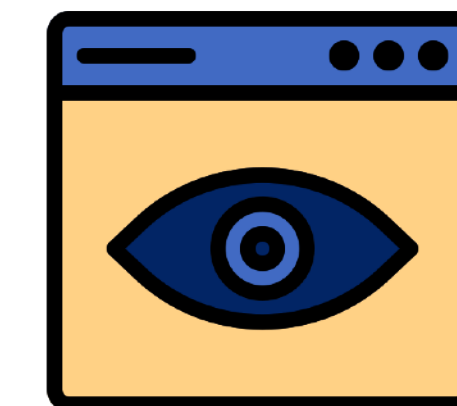
[NDSS18],
[HotSec'17],

Trigger-Action Platforms



[MobiSys17],
[HotNets16]

Recommendation Systems



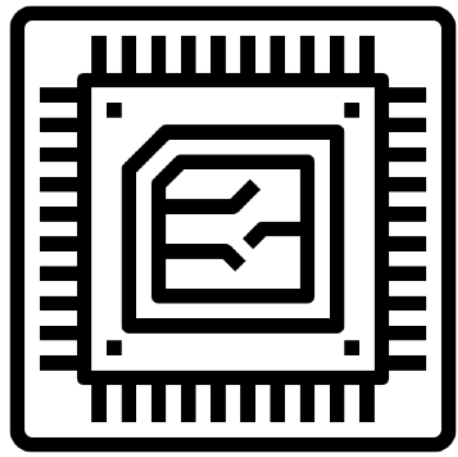
[WEPN],
[FOCI13],
[TR13],
[TR13]

Internet Censorship



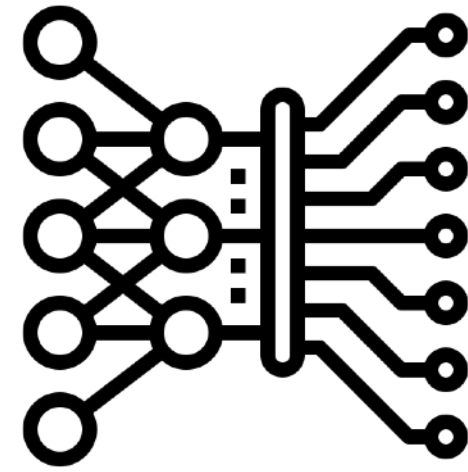
[USENIX Sec23],
[VizSec21],
[WearSys19]

AR/VR Systems



[TECS16],
[TCAD15],
[USENIX Sec12],
[RFIDsec'12]

Embedded Systems



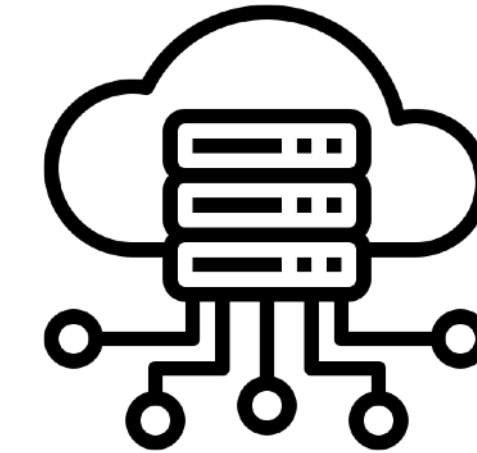
[NeurIPS22],
[USENIX Sec22],
[TSRML22],
[DLS22],
[EDSMLS20],
[WOOT'18],
[CVPR18]

Machine Learning



[ACCESS23],
[Health Tech14],
[Health Tech13]

Medical Devices



[WWW20]

Cloud Platforms



[WWW20],
[SecDev18],
[NDSS17],
[SecDev16],
[USENIX Sec16]

Smart Homes



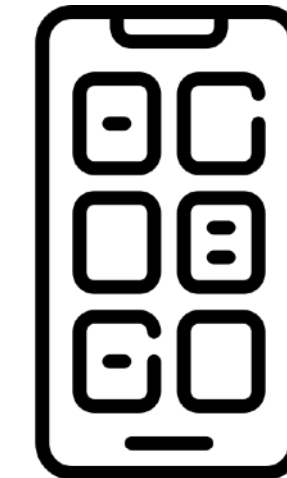
[NDSS24],
[WWW23],
[S&P21]

Web



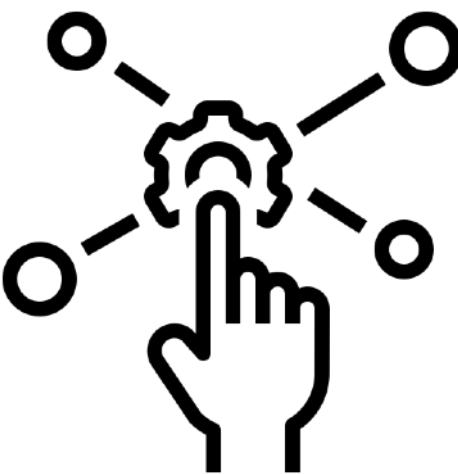
[WAX16],
[ISCA15],
[WACAS14]

Approximate Computing



[USENIX Sec18],
[CCS SPSM15]

Mobile Systems



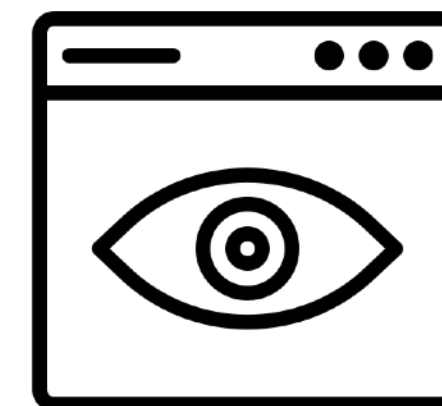
[NDSS18],
[HotSec'17],

Trigger-Action Platforms



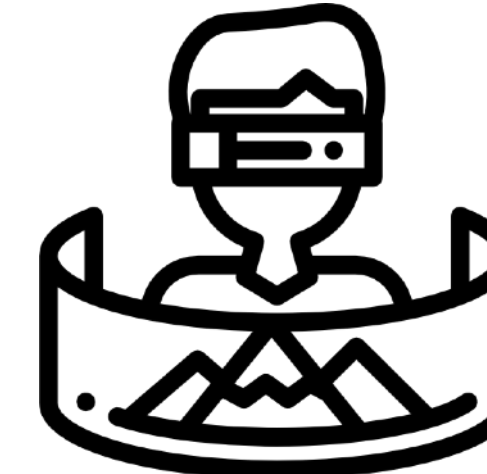
[MobiSys17],
[HotNets16]

Recommendation Systems



[WEPN],
[FOCI13],
[TR13],
[TR13]

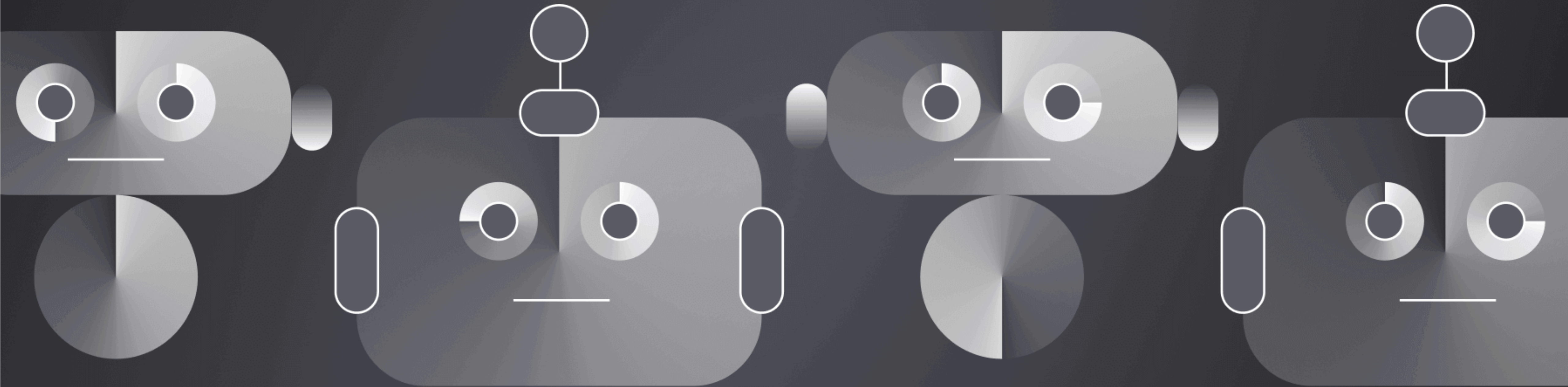
Internet Censorship



[USENIX Sec23],
[VizSec21],
[WearSys19]

AR/VR Systems

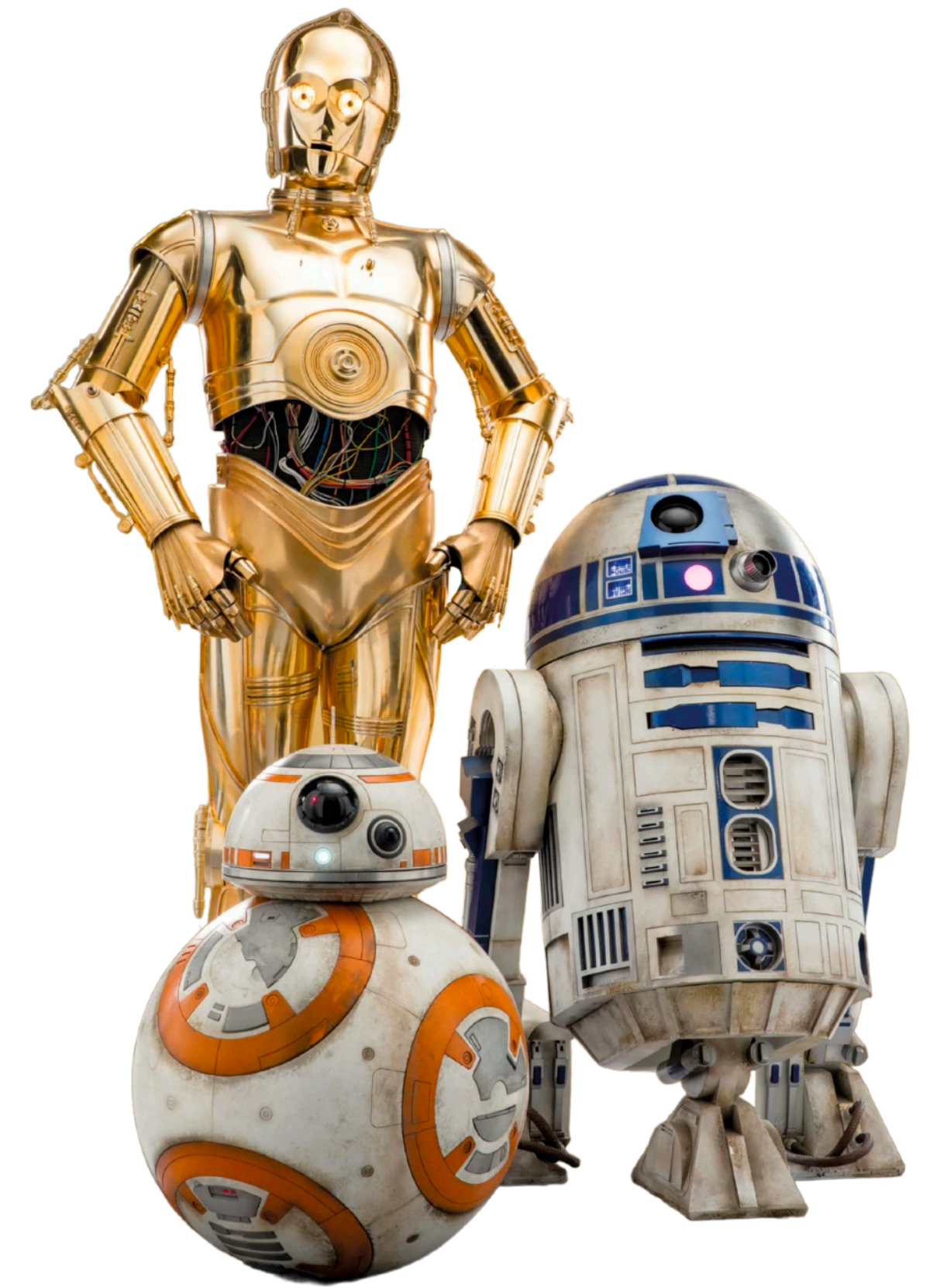
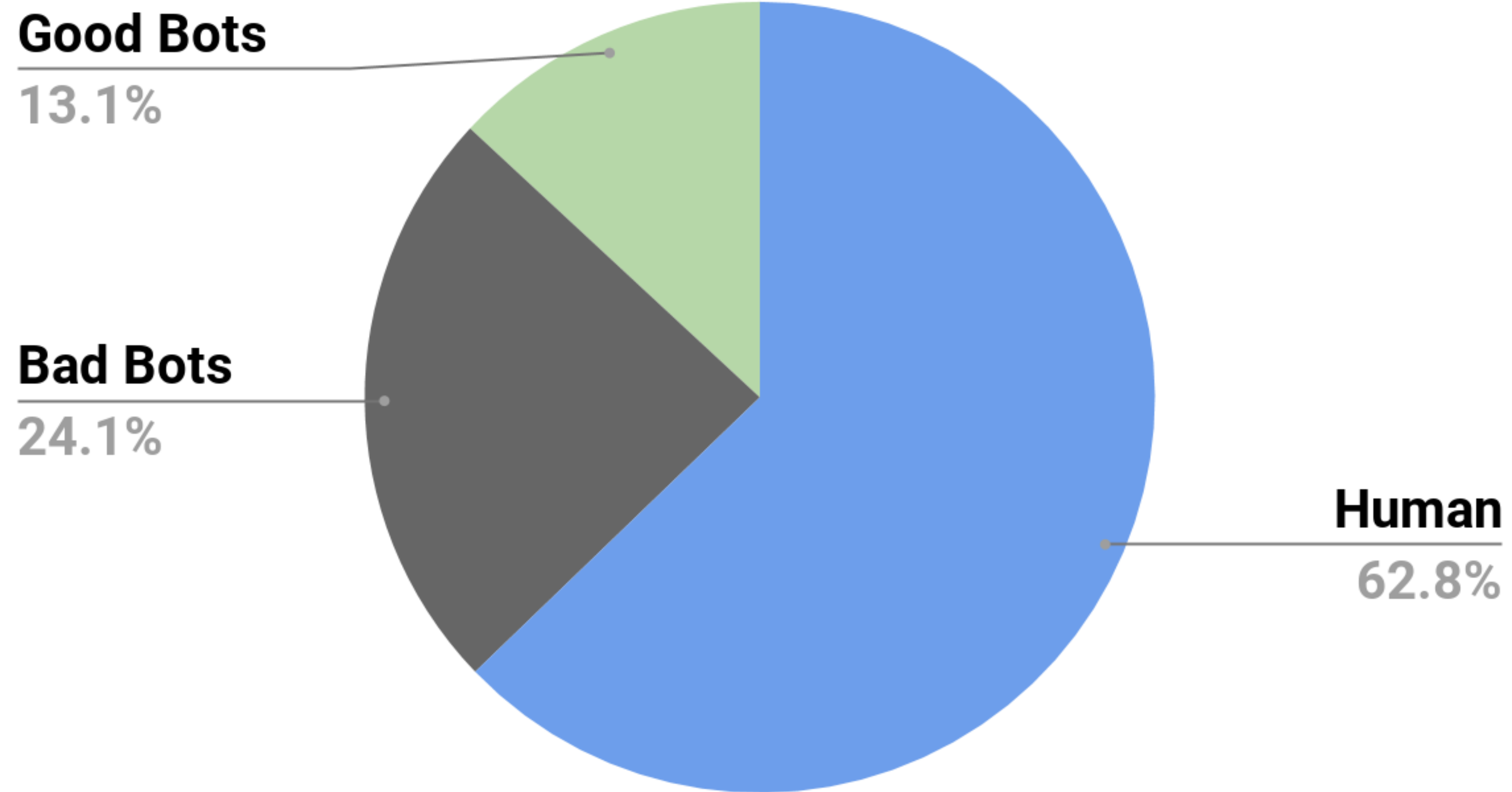
Web Bots



Programs that perform web requests and interact with Internet services, websites, or users on the Internet.

Bots are everywhere!


Bad Bots vs. Good Bots vs. Human in 2019




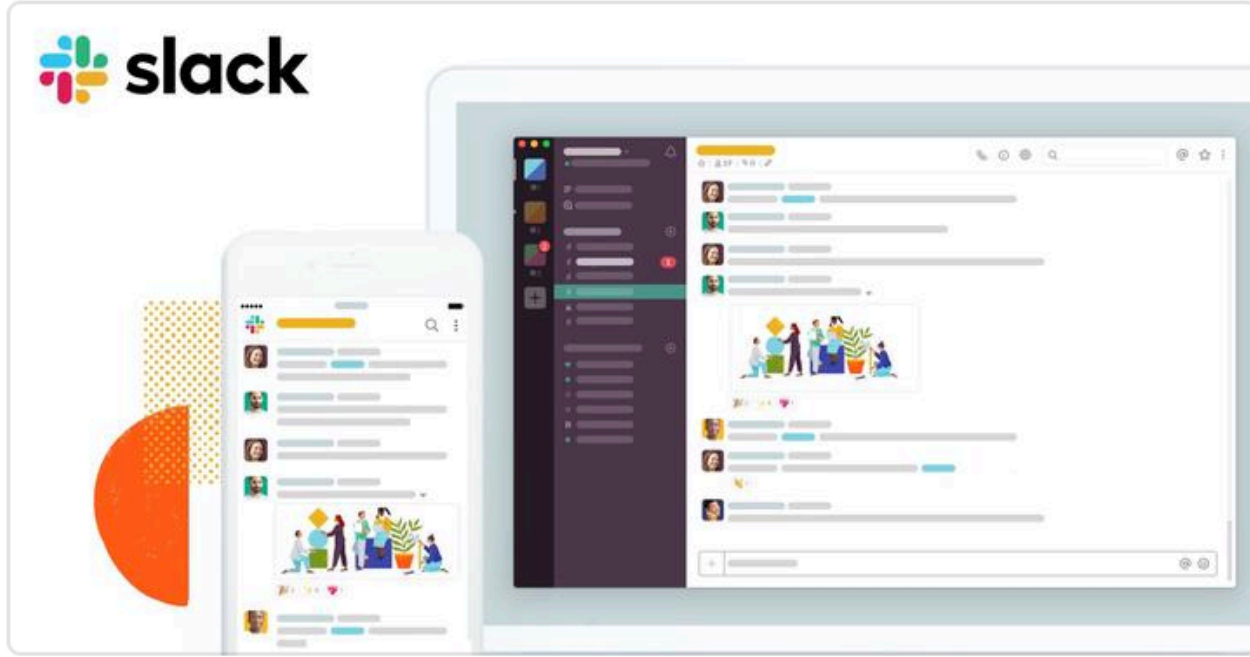
* Imperva. Bad bot report 2020: Bad bots strike back. <https://www.imperva.com/resources/resource-library/reports/2020-bad-bot-report/>


Benign Bots

- Provide content discovery and indexing services
- Create content previews
- Used for Academic/Industry research

 **Olivia Watkins** 10:18 AM
<https://slack.com/features>


 Slack
Features
Slack is where work flows. It's where the people you need, the information you share, and the tools you use come together to get things done. (98 kB) ▾





People also ask :

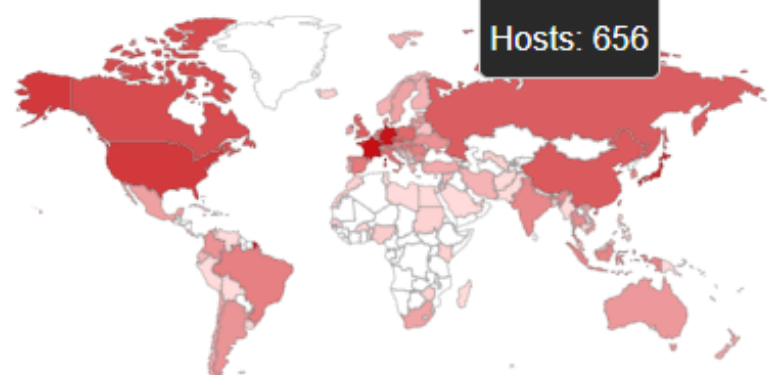
- Is it safe to run a microwave with nothing in it?
- How long can you run a microwave empty?
- Will a microwave explode if there is nothing in it?

 SHODAN Explore Pri

Exploits Maps Images

TOTAL RESULTS
8,868

TOP COUNTRIES



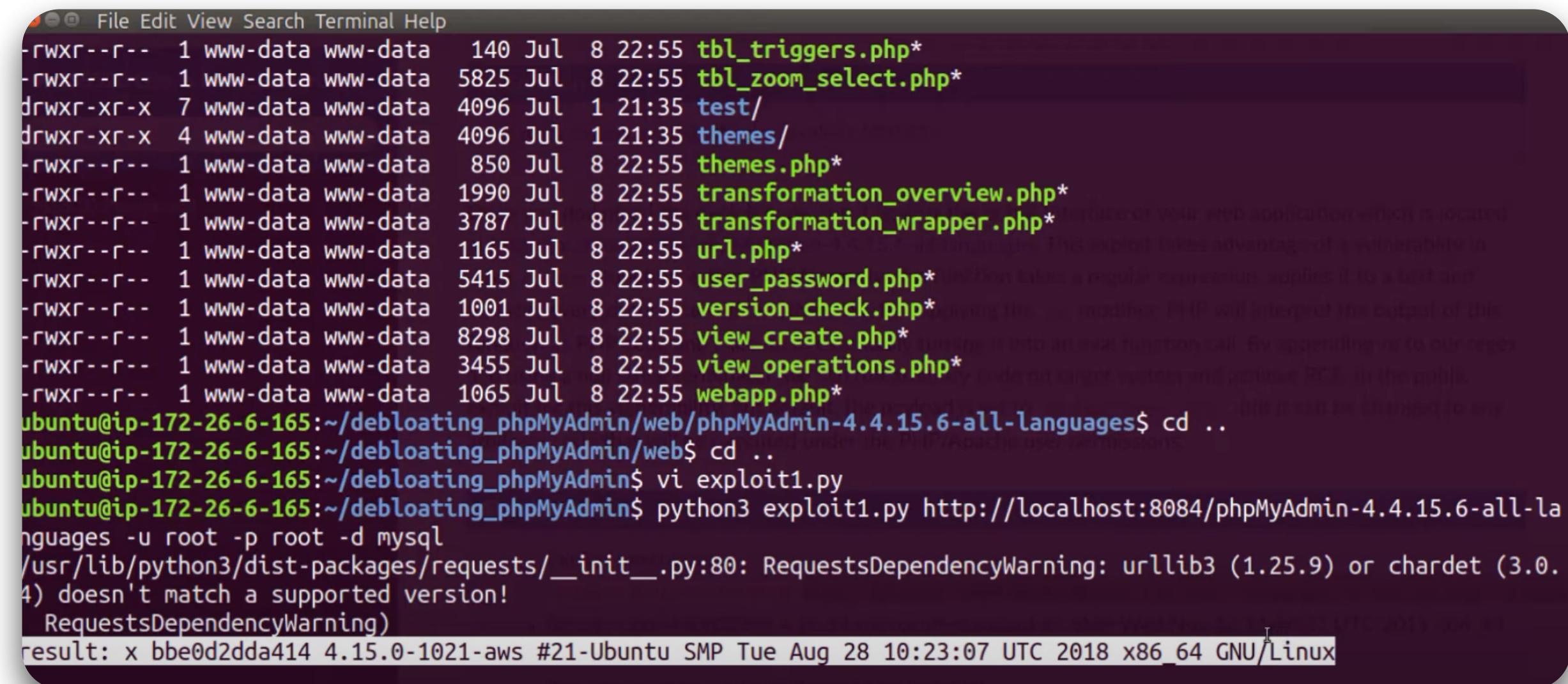
France	1,447
Germany	1,336
Japan	923

210.152.40.170
Link, Inc.
Added on 2021-04-15 00:16:40 GMT
● Japan, Tokyo

Error 405: Not allowed

Malicious Bots

- Credential stuffing attacks
- Probing for vulnerabilities
 - Fingerprint application
 - Steal unprotected information
 - Exploit discovered vulnerabilities
- Denial-of-Service attacks
- Spam and misinformation

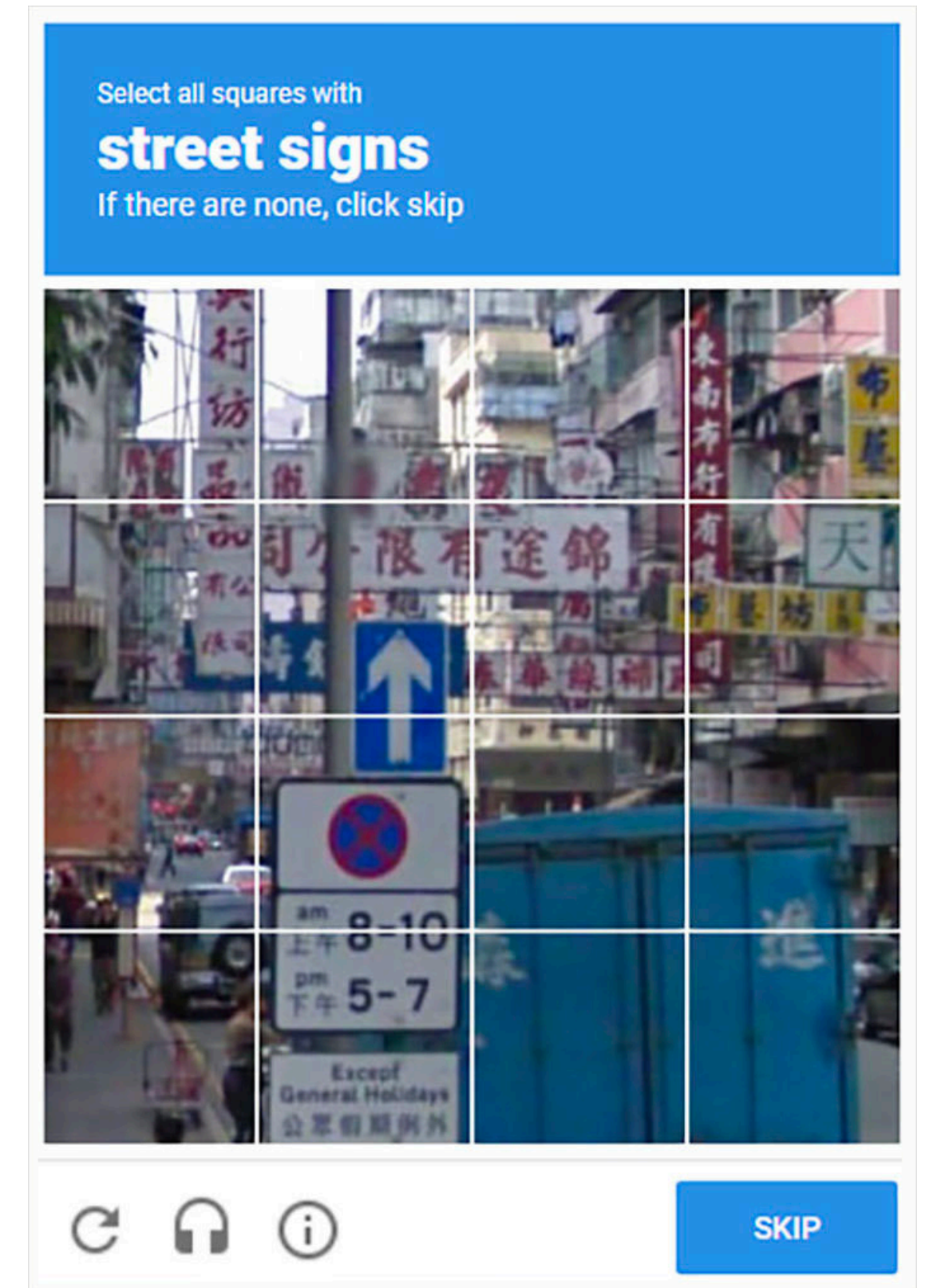


```
File Edit View Search Terminal Help
-rwxr--r-- 1 www-data www-data 140 Jul 8 22:55 tbl_triggers.php*
-rwxr--r-- 1 www-data www-data 5825 Jul 8 22:55 tbl_zoom_select.php*
drwxr-xr-x 7 www-data www-data 4096 Jul 1 21:35 test/
drwxr-xr-x 4 www-data www-data 4096 Jul 1 21:35 themes/
-rwxr--r-- 1 www-data www-data 850 Jul 8 22:55 themes.php*
-rwxr--r-- 1 www-data www-data 1990 Jul 8 22:55 transformation_overview.php*
-rwxr--r-- 1 www-data www-data 3787 Jul 8 22:55 transformation_wrapper.php*
-rwxr--r-- 1 www-data www-data 1165 Jul 8 22:55 url.php*
-rwxr--r-- 1 www-data www-data 5415 Jul 8 22:55 user_password.php*
-rwxr--r-- 1 www-data www-data 1001 Jul 8 22:55 version_check.php*
-rwxr--r-- 1 www-data www-data 8298 Jul 8 22:55 view_create.php*
-rwxr--r-- 1 www-data www-data 3455 Jul 8 22:55 view_operations.php*
-rwxr--r-- 1 www-data www-data 1065 Jul 8 22:55 webapp.php*
ubuntu@ip-172-26-6-165:~/debloating_phpMyAdmin/web/phpMyAdmin-4.4.15.6-all-languages$ cd ..
ubuntu@ip-172-26-6-165:~/debloating_phpMyAdmin/web$ cd ..
ubuntu@ip-172-26-6-165:~/debloating_phpMyAdmin$ vi exploit1.py
ubuntu@ip-172-26-6-165:~/debloating_phpMyAdmin$ python3 exploit1.py http://localhost:8084/phpMyAdmin-4.4.15.6-all-languages -u root -p root -d mysql
/usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.25.9) or chardet (3.0.4) doesn't match a supported version!
RequestsDependencyWarning)
result: x bbe0d2dda414 4.15.0-1021-aws #21-Ubuntu SMP Tue Aug 28 10:23:07 UTC 2018 x86_64 GNU/Linux
```

Example of exploiting CVE-2016-5734 through web requests (arbitrary code execution)

How do websites block bots?

- Drop requests (no response)
- Return error codes (403, 401, ...)
- Block IPs
- Perform human verification

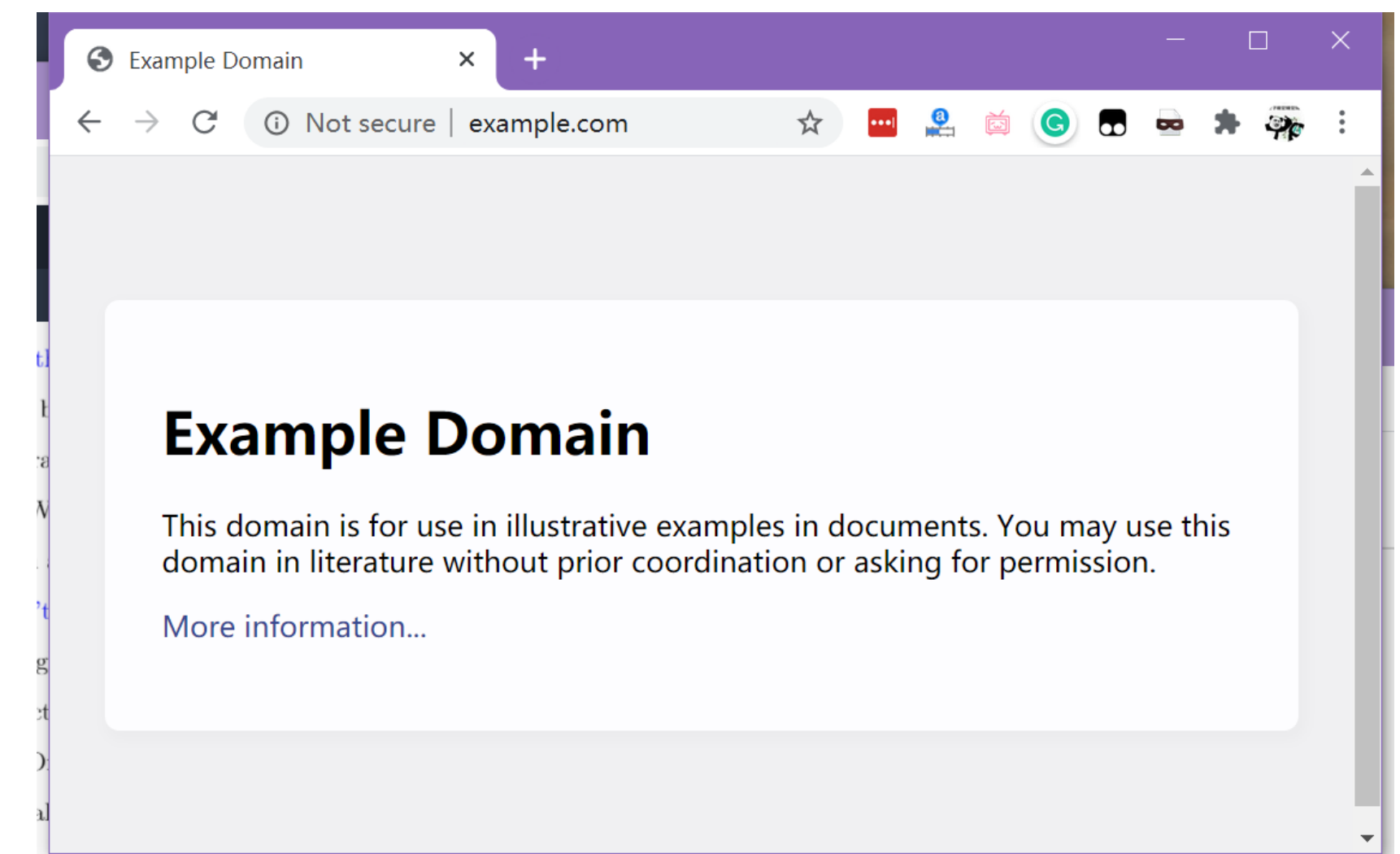
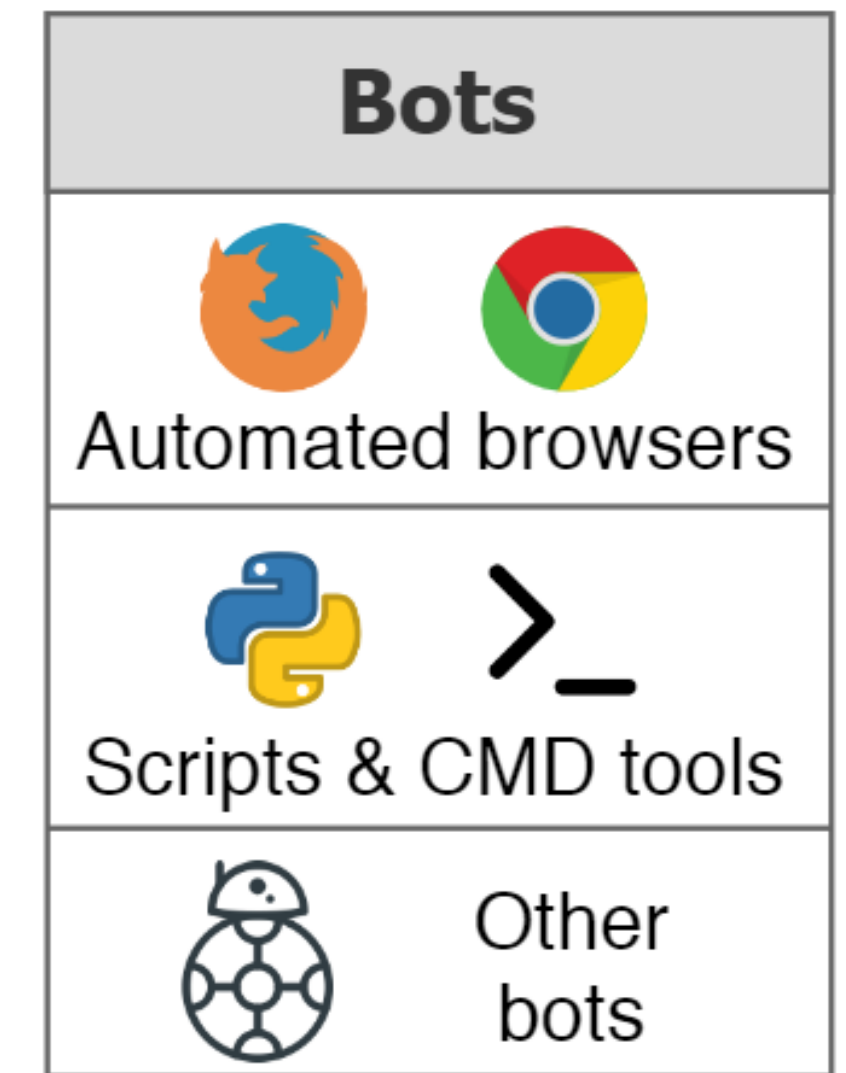


An underlying assumption is that we can detect bots.

Why is it hard to detect bots?

Diverse browsing environments

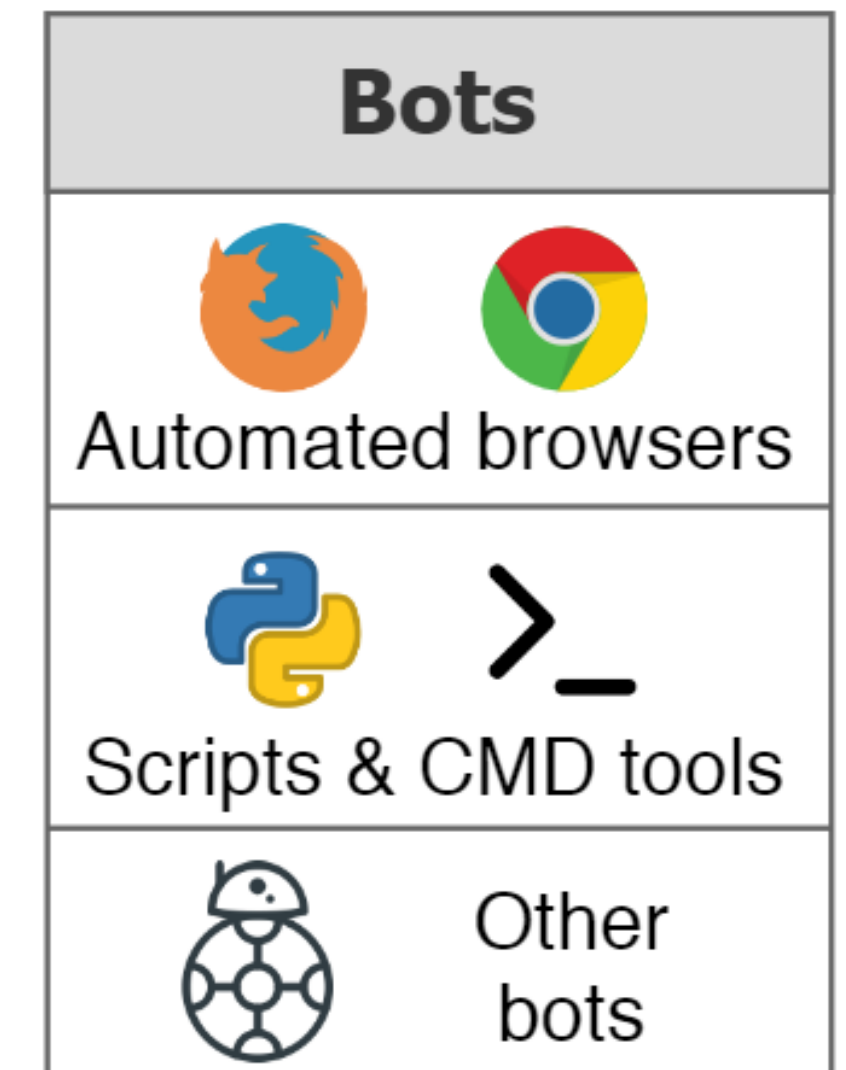
- **Basic crawlers:** wget, curl, etc.
- **Selenium:** almost the same as a normal web browser, except controlled by automated script
 - Can perform click, scroll, ...
 - Can take screenshots
 - Can execute Javascript
- **ZMap:** Scanning the Internet in a few minutes
- **Googlebot:** Crawler mixed with automated browsers and basic crawlers



Why is it hard to detect bots?

Evasion and spoofing techniques

- Spoofing User-Agents
- Rate-limit queries and requests
- Simulate navigating behaviors with automated browsers
- Use proxies to evade IP-based detection



Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3)
AppleWebKit/537.75.14 (KHTML, like Gecko)
Version/7.0.3 Safari/7046A194A

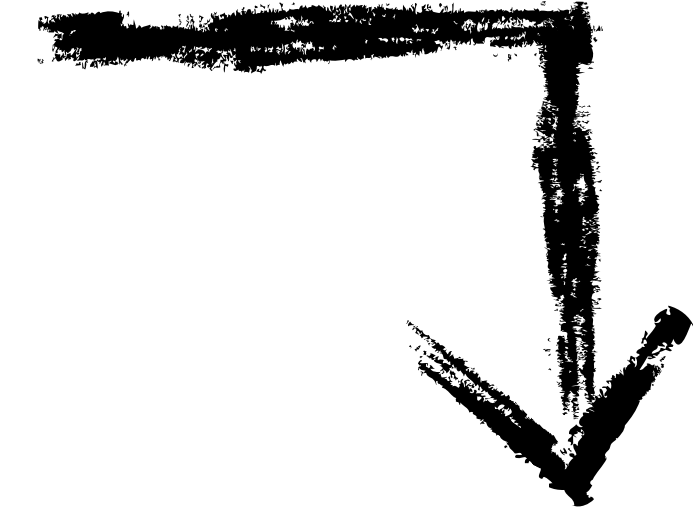
Mozilla/5.0 (Linux; Android 4.4.2; Nexus 4 Build/
KOT49H) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/34.0.1847.114 Mobile Safari/
537.36

Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/84.0.4147.89 Safari/537.36

**Little info on bot impact
toward normal websites**

**No public dataset of
bot-only traffic**

How can we minimize the effect of malicious bots without hindering benign bots?



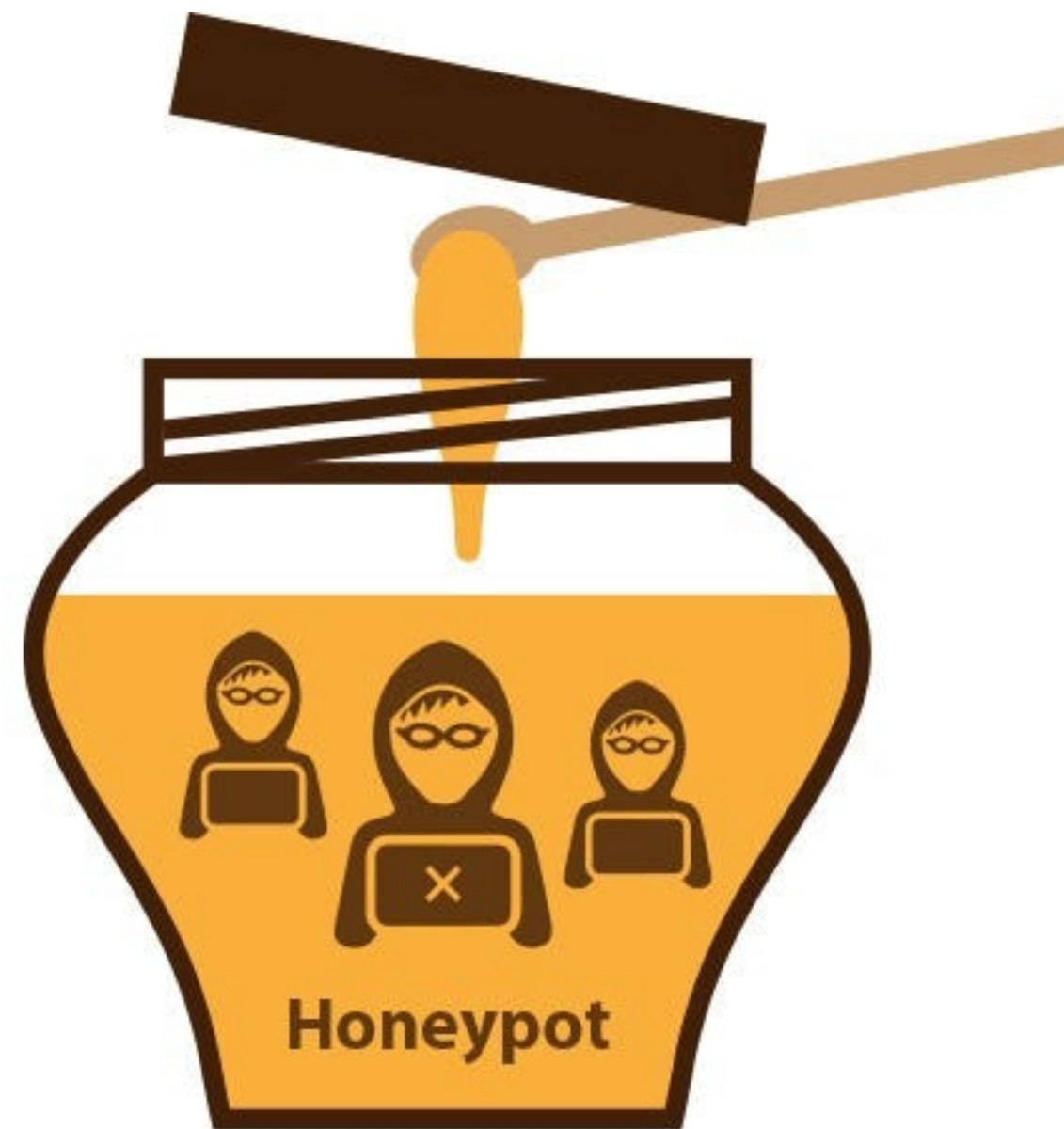
How can we understand the true impact and purpose of bots?



How can we build a bot-only dataset?

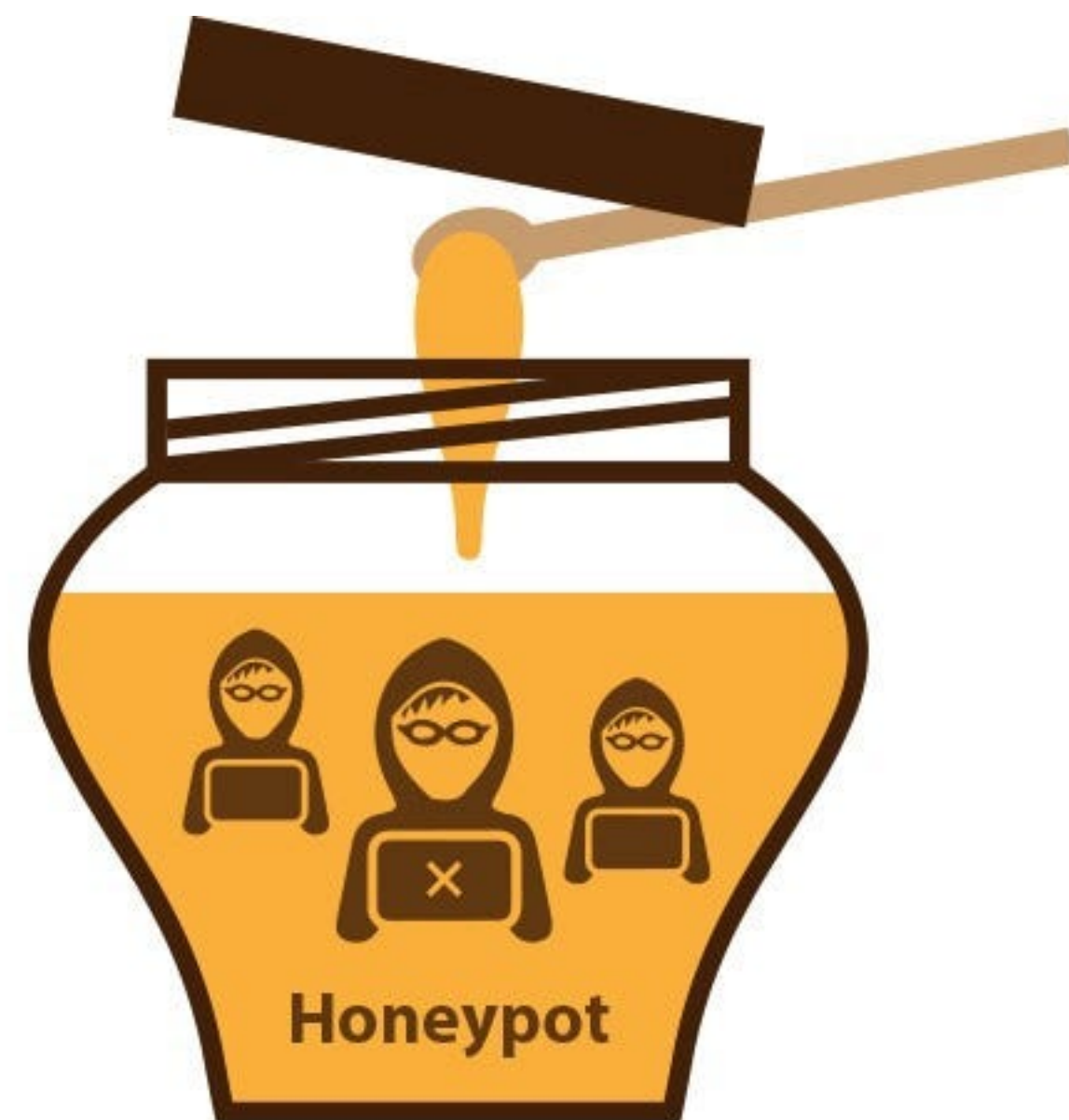
How can we build a bot-only dataset?

**Build a
measurement
infrastructure!**



How can we build a bot-only dataset?

Build a measurement infrastructure!



Detection Methods?

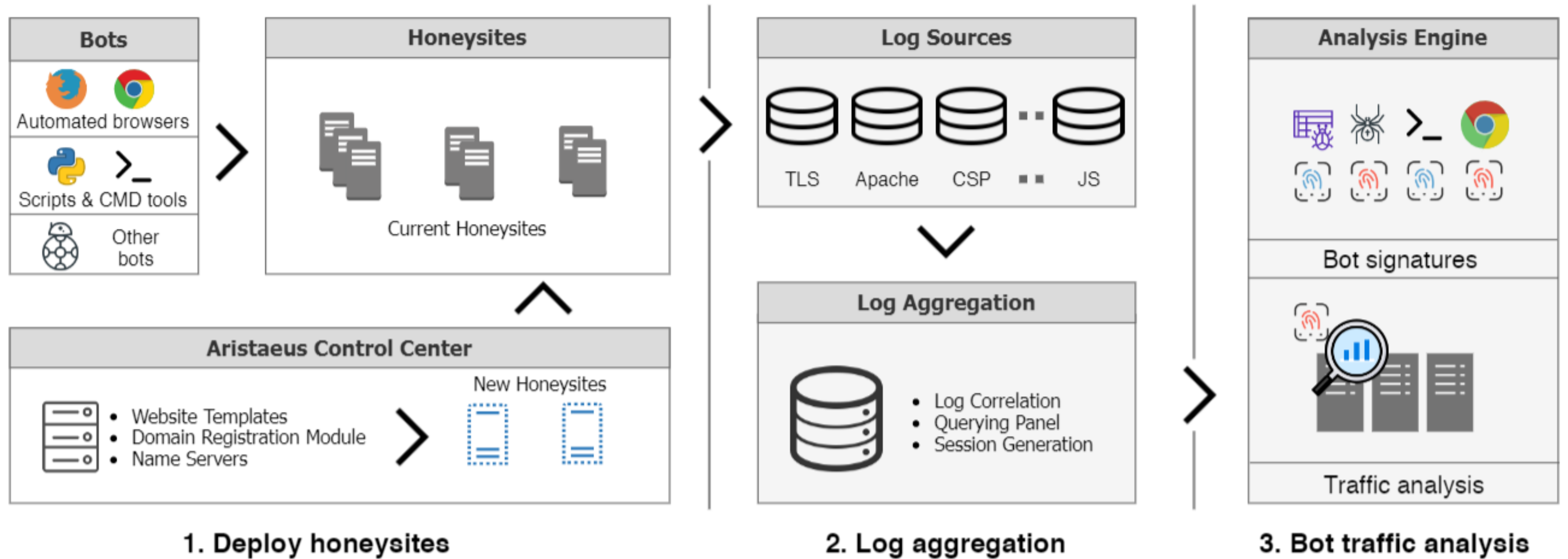
Scalability?

Diversity?

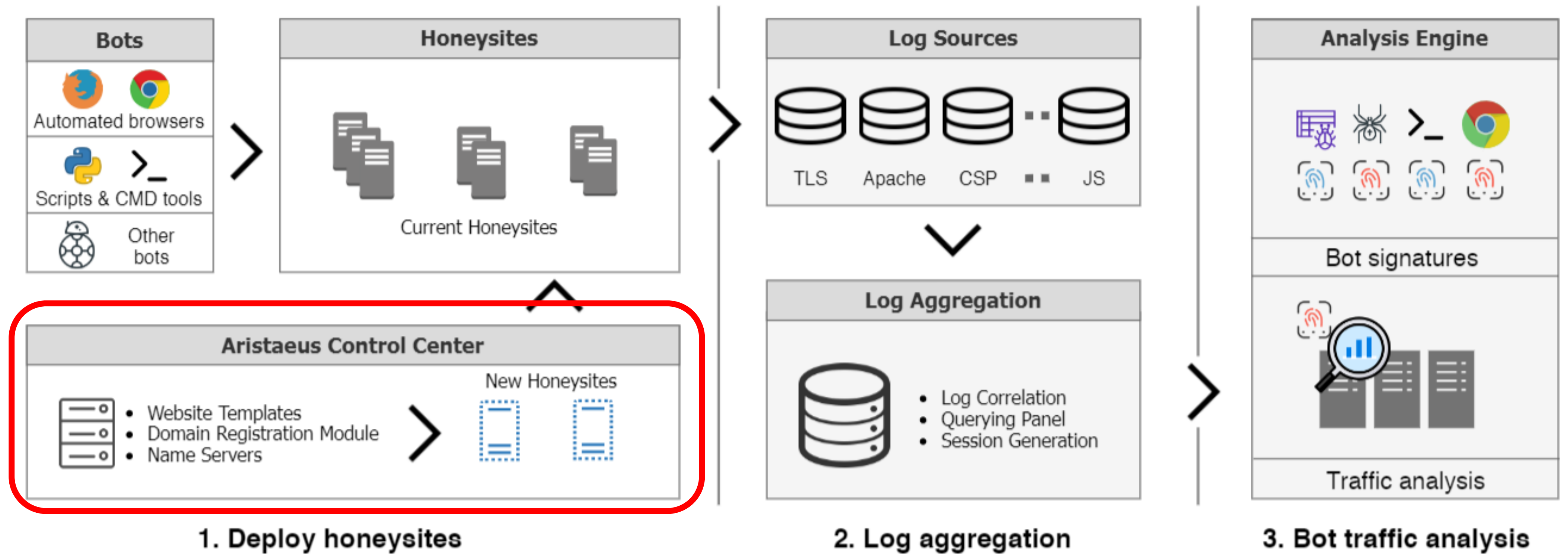
Aristaeus*

*Minor God in Greece mythology, caring over beekeepers

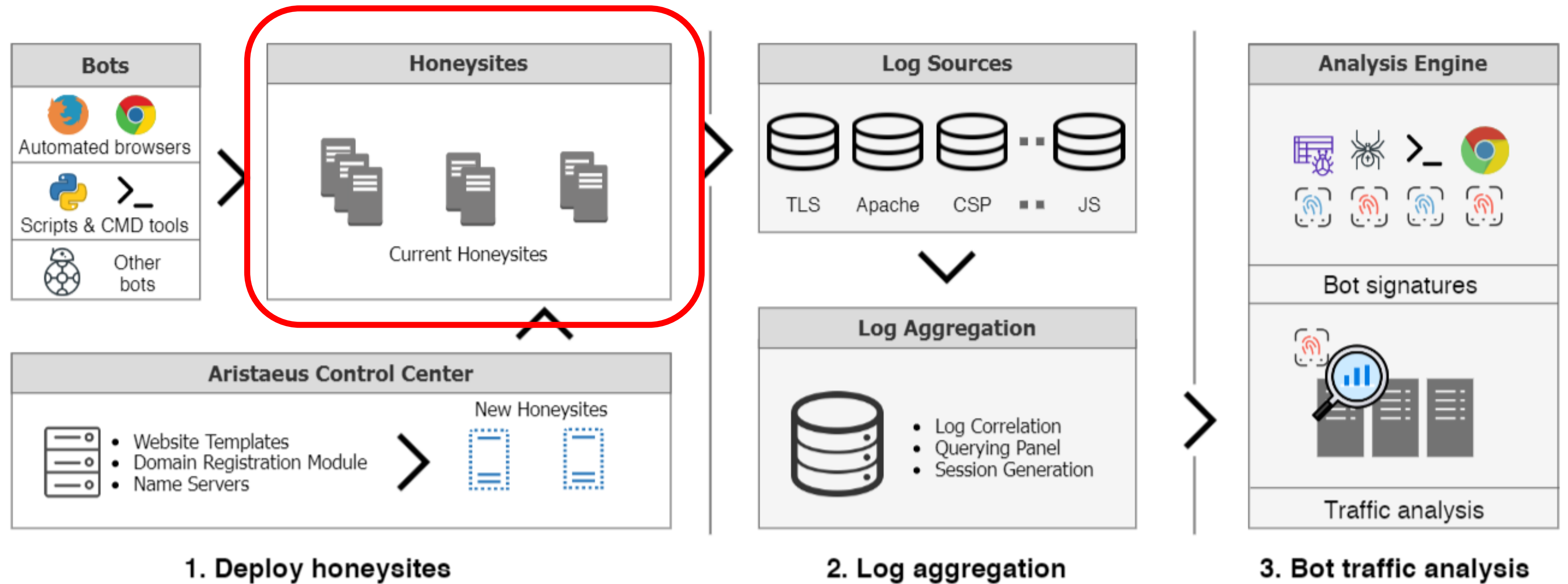
Aristaeus



Aristaeus



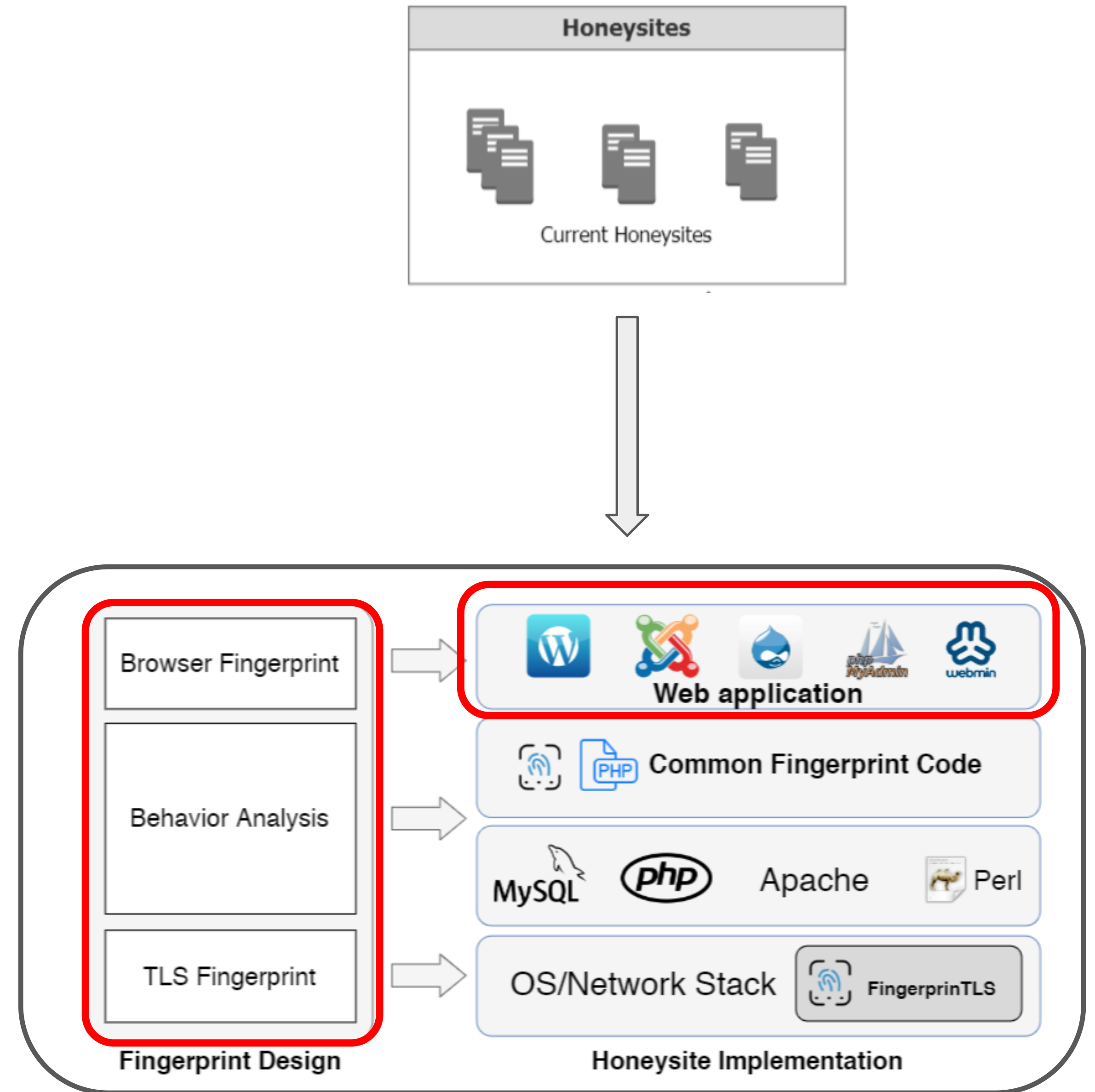
Aristaeus



Honeysite structure

Web Application

- Aristaeus currently supports 5 applications
- 3 CMS web applications:
WordPress, Joomla, Drupal
- 2 web Admin tools:
PHPMyAdmin, Webmin



Honeysite structure

Browser fingerprinting

- **Javascript API support**
 - Basic support test
 - document.write(), var img ...
 - Ajax support
- **Support for security policies**
 - CSP, X-Frame-Options, Mixed Content (HTTP/HTTPS) ,etc.
 - First time security mechanisms are used for fingerprinting clients
- **Browser fingerprinting**
 - Modified FPJS2

The image shows a browser's developer console and network tab. The console displays several error messages related to mixed content and failed resource loading. The network tab shows the response headers for a request to a honeysite, including Cache-Control, Connection, Content-Security-Policy, Date, Expires, and Keep-Alive. A code block is overlaid on the network tab, showing the implementation of a custom image element and its insertion into the document body.

Console Errors:

- Mixed Content: The page at 'https://tinychef.info/' was loaded over HTTPS, but the following resource also be served over HTTP.
- Failed to load resource: the server responded with a status of 404 (Not Found)
- Refused to load the image 'https://pf53ae.tinychef.info/fpcodes_gen/vis2.jpg?loc=scrip&rndstr=WFY2ZUpQbXZ3TGJlaTVUU1' because it was served by a different origin than the document.
- Refused to load the image 'https://pf53ae.tinychef.info/fpcodes_gen/vis1.jpg?loc=scrip&rndstr=aXhtUmdnR2sxSm' because it was served by a different origin than the document.

Network Tab Response Headers:

- Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
- Connection: Keep-Alive
- Content-Security-Policy: default-src 'self' fonts.googleapis.com http://www.tinychef.info http://tinychef.info http://www.tinychef.info all
- Date: Thu, 17 Aug 2017 10:10:10 GMT
- Expires: Wed, 11 Jan 1996 08:00:00 GMT
- Keep-Alive: timeout=30

Code Block:

```
var customImg2 = new Image(1, 1);
customImg2.src = "https://tinychef.info/fpcodes_gen/vis2.jpg?loc=scrip&rndstr=WFY2ZUpQbXZ3TGJlaTVUU1";
document.body.appendChild(customImg2);
document.write('<img src="https://tinychef.info/fpcodes_gen/vis1.jpg?loc=scrip&rndstr=aXhtUmdnR2sxSm"')
```

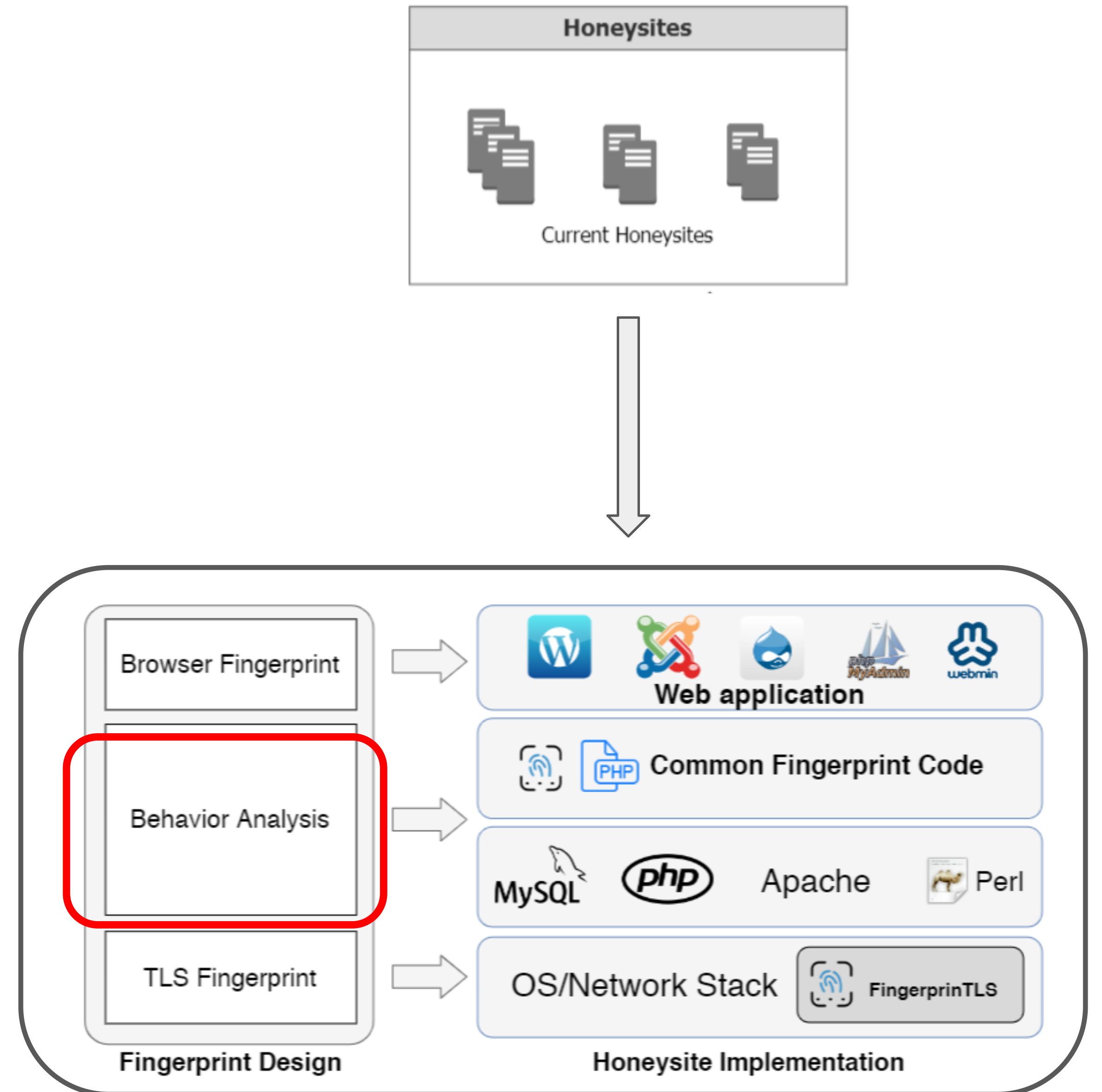
Diagram Labels:

- Honeysites
- ent Honeysites
- Fingerprint Design
- Honeysite Implementation

Honeysite structure

Behavior fingerprinting

- Honoring robots.txt
- Customized error pages
 - We know bots probing for specific files that may not exist
 - Injecting fingerprinting code into 404 page
- Caching and resource sharing
 - Use “no-cache” header
 - Encode cache-breaker on certain URL
E.g. /a.jpg?r=[encoded IP+nonce]

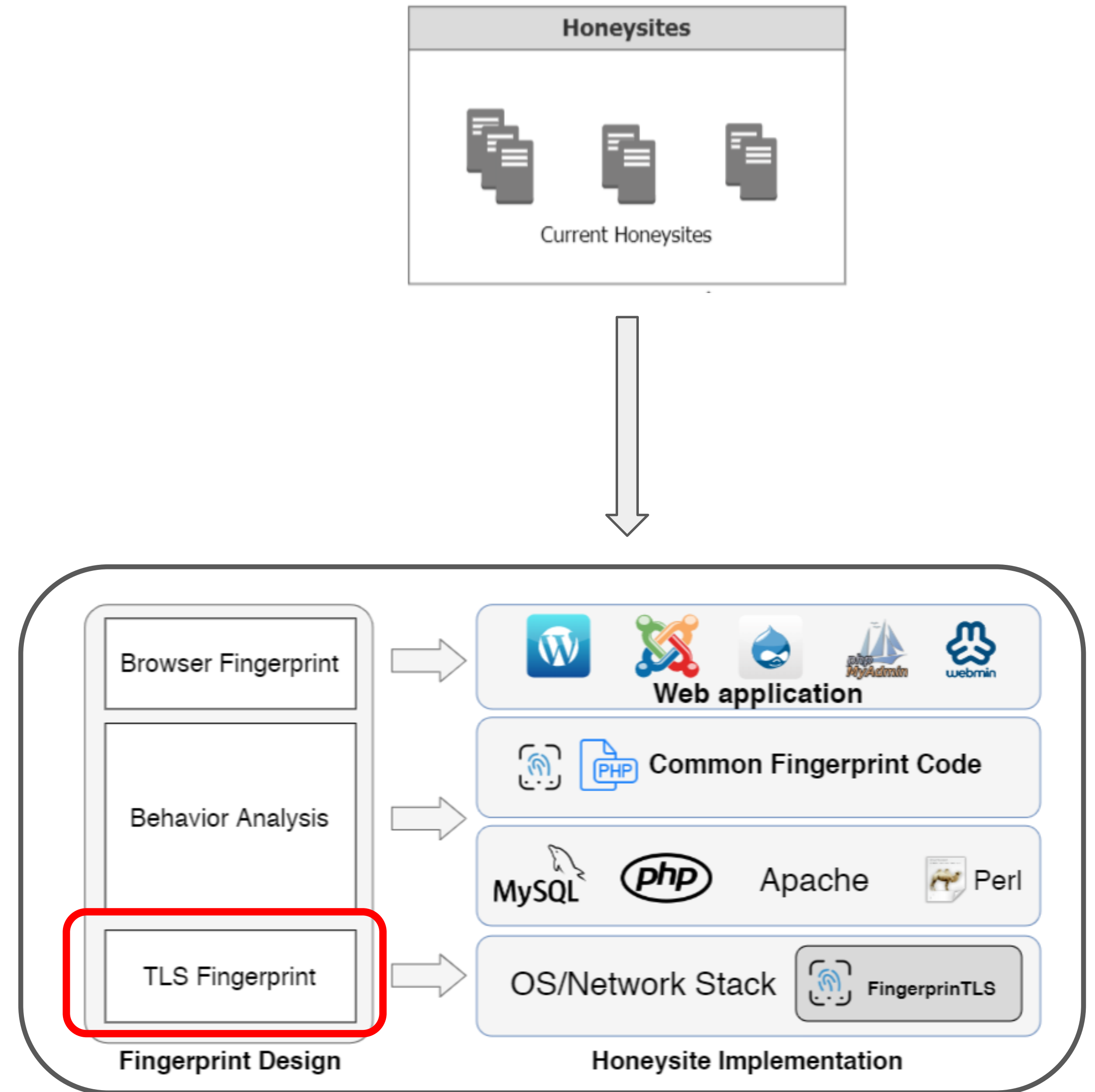


Honeysite structure

TLS fingerprinting

TLS fingerprint is performed passively (server side) compared to JS fingerprinting (client side)

- Cipher suites
- Signature algorithms
- E-curve
- TLS version
- Compression length



Honeysite structure

TLS fingerprinting

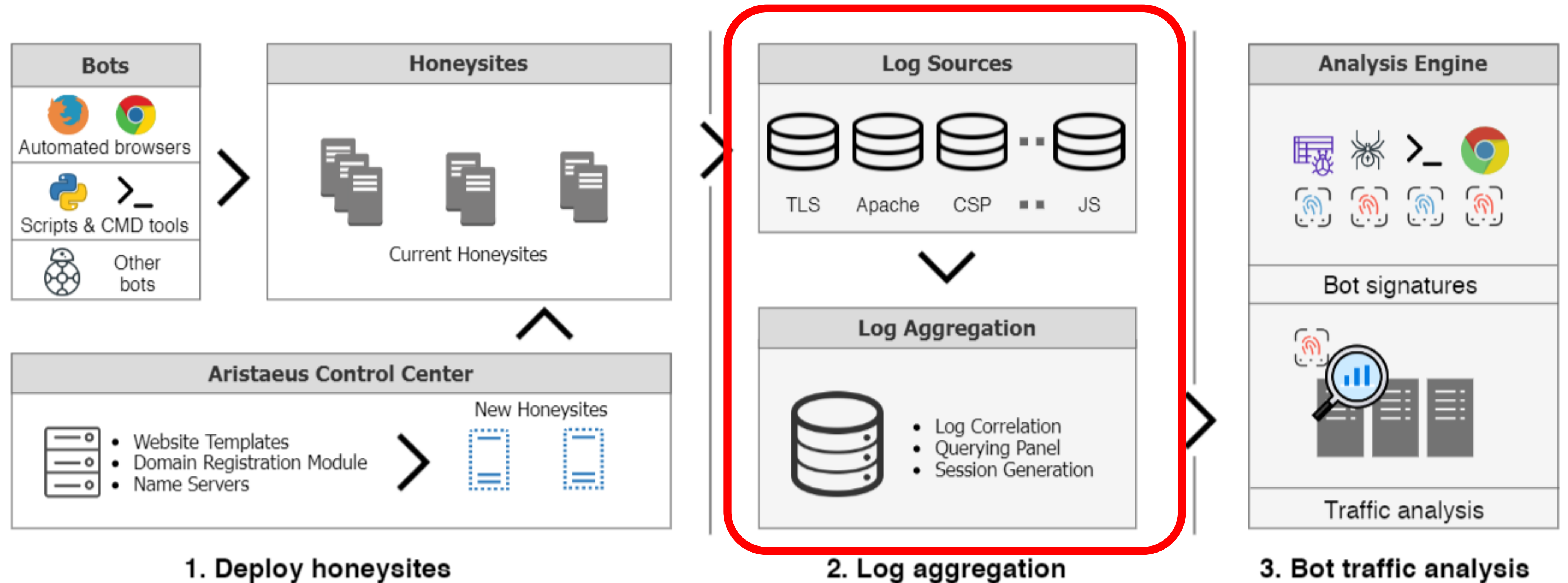
```
import "net/http"  
  
resp, err := http.Get("https://example.com/")
```

Example of TLS fingerprint:

```
"tlsfp": {  
  "ciphersuite": "0xC02F 0xC030 0xC02B 0xC02C 0xCCA8 0xCCA9 0xC013  
                0xC009 0xC014 0xC00A 0x009C 0x009D 0x002F  
                0x0035 0xC012 0x000A",  
  "tls_version": "0x0303",  
  "sig_alg": "0x0401 0x0403 0x0501 0x0503 0x0601 0x0603 0x0201 0x0203",  
  "src_port": 22260,  
  "record_tls_version": "0x0301",  
  "timestamp": "2020-04-25 03:55:59",  
  "server_name": "www.historytenantfile.com",  
  "ipv4_src": "167.71.193.105",  
  "e_curves": "0x001D 0x0017 0x0018 0x0019",  
  "extensions": "0x0000 0x0005 0x000A 0x000B 0x000D 0xFF01 0x0012",  
  "ciphersuite_length": "0x0020",  
}
```

Go-http-client

Overview of Aristaeus



Deployment

- **Registered 100 domains**
 - Make sure they are not registered before (i.e. once registered then expired), to eliminate effects of residual trust
 - Did not publicly advertise our domains
 - Confident that the vast majority of clients were bots
- **Spawn a honeysite for each domain via AWS**
 - Use Let's Encrypt to obtain valid TLS certificates for each domain
 - Spawn in North America, Europe, and Asia.
- **Use central server to periodically collect logs from all 100 honeysites**
 - Logs are stored in Elasticsearch cluster for analysis

7

Months

26.4M

Requests

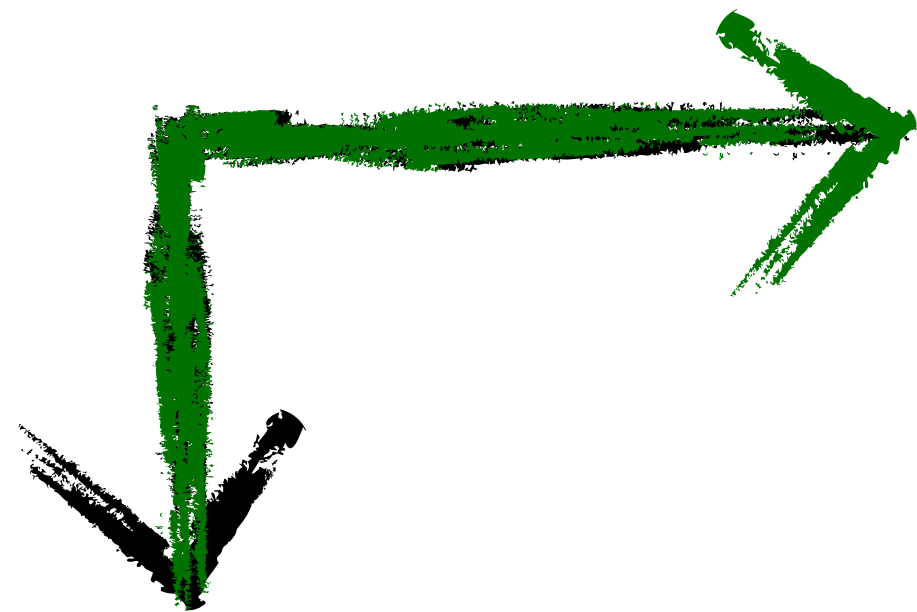
206GB

Data

How can we minimize the effect of malicious bots without hindering benign bots?



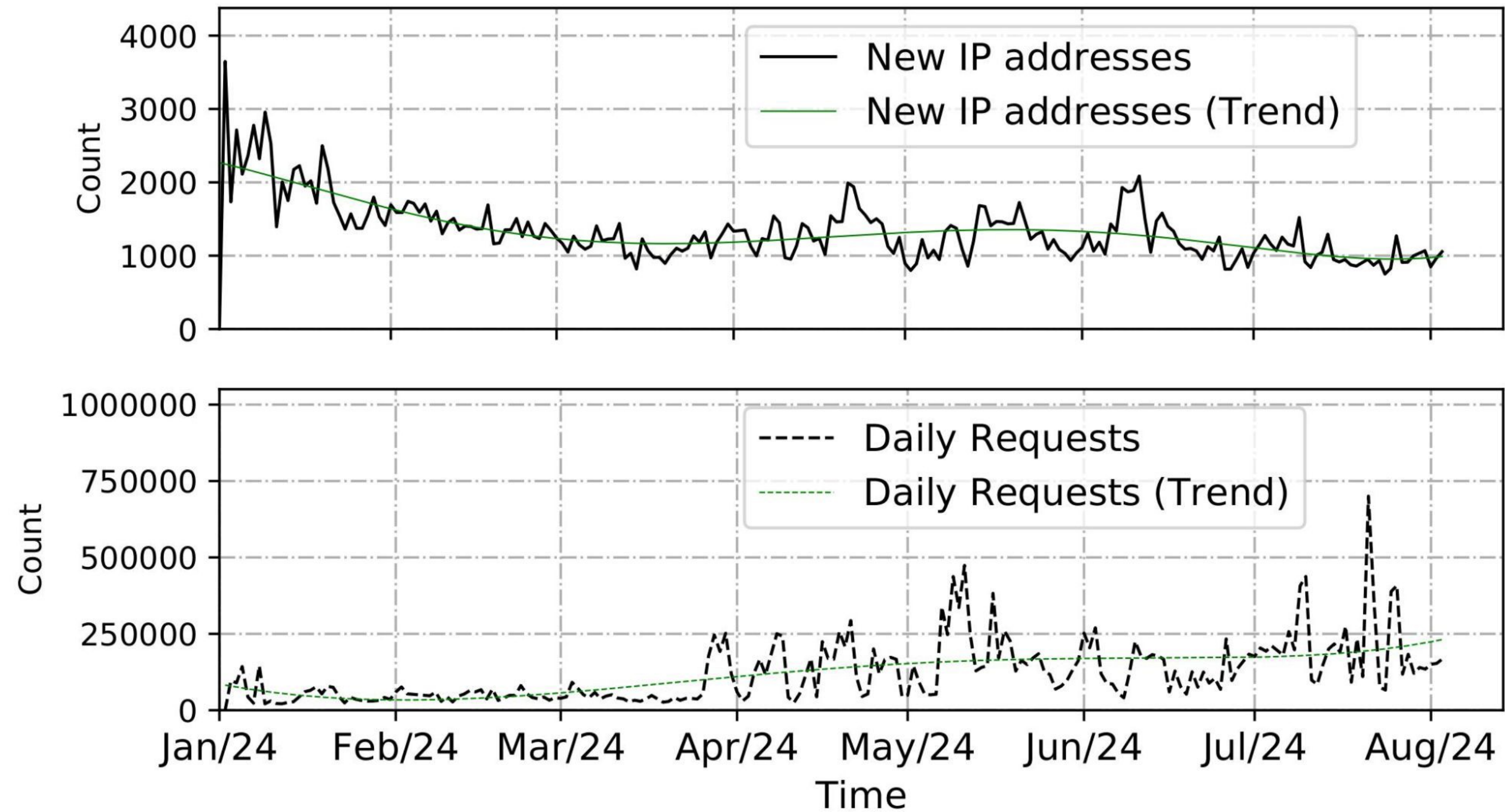
How can we understand the true impact and purpose of bots?



How can we build a bot-only dataset?

Bot Traffic Analysis

- Honeysites keep observing traffic from new IP addresses
- Average: 1,235 requests/day



Bot Traffic Analysis

Use **Host** header to determine how bots discovered us

- **44% bots visit us through IP (Host:1.2.3.4)**
 - IP space scanning
 - Network monitoring
- **26% bots visit us through domain (Host: example.com)**
 - DNS zone files
 - Certificate transparency logs
- **30% bots do not present Host header**

```
hastorensic": true,  
flog": {  
  "headersText": [  
    "Host:52.3.222.202",  
    "User-Agent:Mozilla/5.0 (Windows NT 10.  
    "Accept:/*/*\n"  
  ],  
  "headersKV": {  
    "Nonce": "ap",  
    "Host": "52.3.222.202",  
    "Accept": "/*/*\n",  
    "User-Agent": "Mozilla/5.0 (Windows NT  
  },  
  "request": "GET / HTTP/1.1",  
  "fid": "Xq0yiz0QYdqDT09GefocHgAAAAI"
```

```
],  
"headersKV": {  
  "Nonce": "ap",  
  "Accept-Encoding": "gzip",  
  "Connection": "close\n",  
  "User-Agent": "Mozilla/5.0 (X11;  
  "Host": "www.objectivecurtainbook  
  "Cookie": "csessionid=5ea3edf88e02d;  
  "Referer": "https%3a//www.objecti  
  "Content-Type": "application/x-w  
},  
"request": "GET /wp-admin/ HTTP/1.1  
"fid": "XqPt-yv0Qhj01fbRhAU6sQAAAAAL
```


Bot Traffic Analysis

✓=exists, ✗=does not exist, ⓧ=not accessible

Wordpress	99.88 ✓	97.69 ✓	99.73 ✓	0.61 ✗	36.97 ✓	18.37 ⓧ	0.09 ✗	0.00 ✗	1.00 ⓧ	0.00 ✗	0.00 ✗	23.18 ✓
Joomla	0.05 ✗	0.80 ✗	0.14 ✗	97.94 ✓	35.23 ✓	21.43 ⓧ	0.10 ✗	0.00 ✗	30.95 ✓	0.00 ✗	0.00 ✗	21.44 ✓
Drupal	0.02 ✗	0.70 ✗	0.10 ✗	0.62 ✗	12.64 ✓	20.61 ⓧ	99.74 ✓	99.98 ✗	0.91 ⓧ	0.00 ✗	0.00 ✗	20.63 ✓
PHPMyAdmin	0.01 ✗	0.57 ✗	0.02 ✗	0.28 ✗	9.25 ✓	19.17 ⓧ	0.05 ✗	0.01 ✗	66.41 ✓	100.00 ✓	0.00 ✗	18.22 ✓
Webmin	0.04 ✗	0.23 ✗	0.01 ✗	0.55 ✗	5.91 ✓	20.42 ⓧ	0.01 ✗	0.00 ✗	0.72 ⓧ	0.00 ✗	100.00 ✓	16.53 ✓
	xmlrpc.php	wp-login.php	/wp-admin/	/administrator/	/robots.txt	instance-identity	/user/login	/CHANGELOG.txt	(POST) /index.php	/phpmyadmin/index.php	/session_login.php	/(document root)

Bot Traffic Analysis

✓=exists, ✗=does not exist, ⚡=not accessible

Wordpress	99.88 ✓	97.69 ✓	99.73 ✓	0.61 ✗	36.97 ✓	18.37 ⚡	0.09 ✗	0.00 ✗	1.00 ⚡	0.00 ✗	0.00 ✗	23.18 ✓
Joomla	0.05 ✗	0.80 ✗	0.14 ✗	97.94 ✓	35.23 ✓	21.43 ⚡	0.10 ✗	0.00 ✗	30.95 ✓	0.00 ✗	0.00 ✗	21.44 ✓
Drupal	0.02 ✗	0.70 ✗	0.10 ✗	0.62 ✗	12.64 ✓	20.61 ⚡	99.74 ✓	99.98 ✗	0.91 ⚡	0.00 ✗	0.00 ✗	20.63 ✓
PHPMyAdmin	0.01 ✗	0.57 ✗	0.02 ✗	0.28 ✗	9.25 ✓	19.17 ⚡	0.05 ✗	0.01 ✗	66.41 ✓	100.00 ✓	0.00 ✗	18.22 ✓
Webmin	0.04 ✗	0.23 ✗	0.01 ✗	0.55 ✗	5.91 ✓	20.42 ⚡	0.01 ✗	0.00 ✗	0.72 ⚡	0.00 ✗	100.00 ✓	16.53 ✓
	xmlrpc.php	wp-login.php	/wp-admin/	/administrator/	/robots.txt	instance-identity	/user/login	/CHANGELOG.txt	(POST) /index.php	/phpmyadmin/index.php	/session_login.php	/((document root))

Bot Traffic Analysis

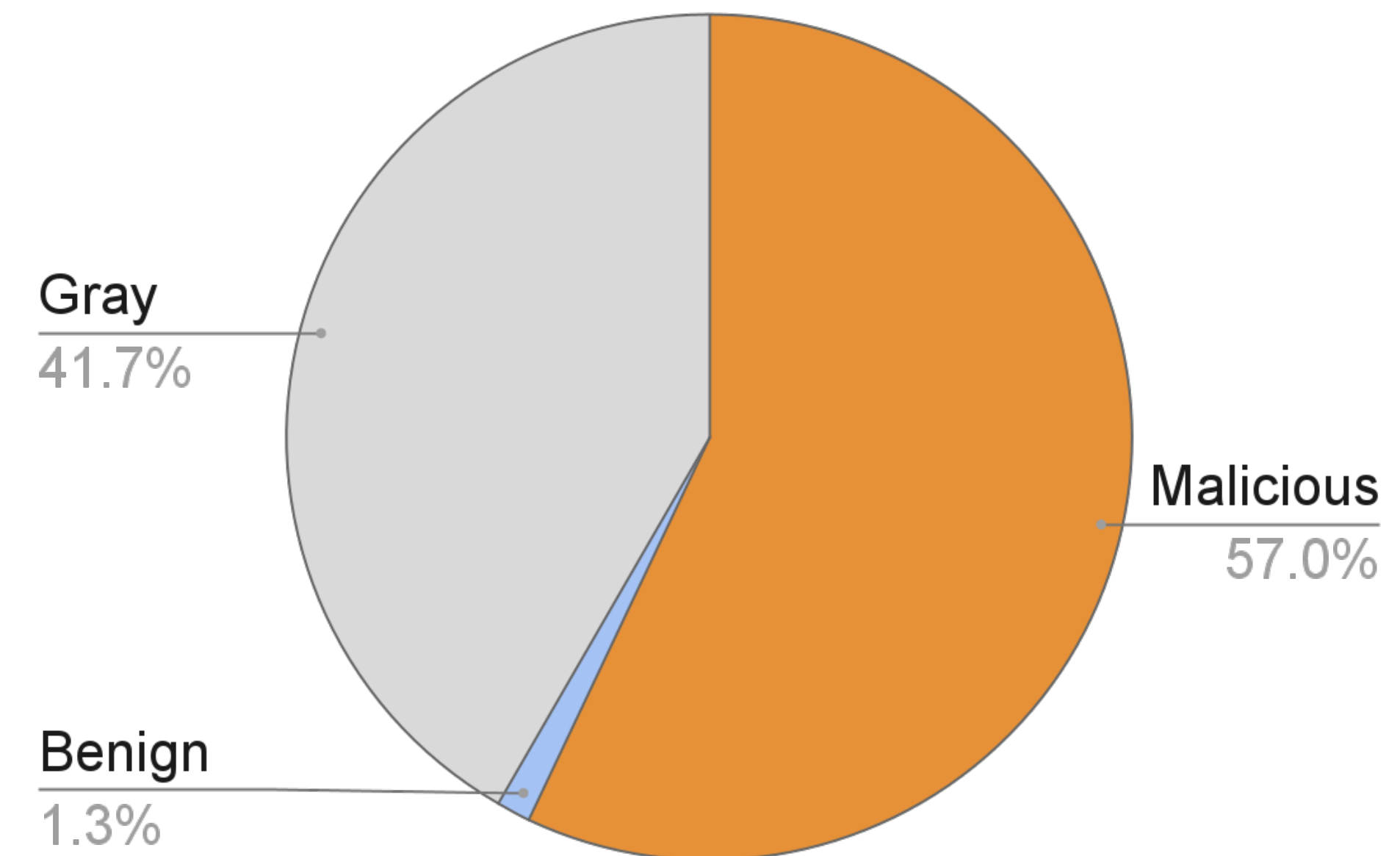
Wordpress	99.88 ✓	97.69 ✓	99.73 ✓	0.61 ✗	36
Joomla	0.05 ✗	0.80 ✗	0.14 ✗	97.94 ✓	35
Drupal	0.02 ✗	0.70 ✗	0.10 ✗	0.62 ✗	12
PHPMyAdmin	0.01 ✗	0.57 ✗	0.02 ✗	0.28 ✗	9.
Webmin	0.04 ✗	0.23 ✗	0.01 ✗	0.55 ✗	5.

xmlrpc.php
wp-login.php
/wp-admin/
/administrator/
/robots.txt
instance-;

- Bots **first discover** that a website is running WordPress, **then target** the login page of wp-login.php, wp-admin, and xmlrpc.php.
- Bots are highly specific, targeting easy-to-exploit endpoints.
- Login endpoints of our applications that received the most attention

Bot Intentions

- **Benign**
 - Asking for valid resources similar to a normal browser
 - No manifested intentions of attacking
- **Malicious**
 - Send unsolicited POST requests toward authentication endpoints
 - Send invalid requests trying to exploit vulnerabilities
- **Other/Gray**
 - None of the above traits



Bot Intentions

Benign

- **Search Engine bots**
 - Googlebot, Bingbot, etc.
- **Academic and industry scanners**
 - Builtwith, Netcraft
 - Internet Archive
 - Academic research bot

Type	Total SEBot Requests	Verified Requests
Googlebot	233,024	210,917 (90.5%)
Bingbot	77,618	77,574 (99.9%)
Baidubot	2,284	61 (0.026%)
Yandexbot	4,894	4,785 (97.8%)
Total	317,820	293,337 (92.3%)

Use reverse-DNS verification to make sure they did not spoof their identity

Bot Intentions

Malicious

- **Credential bruteforce attempts**
- **Reconnaissance attempts**
 - Application fingerprinting
 - Exploitation attempts
 - Scanning for publicly-reachable backdoors
 - Scanning for unprotected sensitive files

Path	# requests	Unique IPs	Target applications
/CHANGELOG.txt	116,513	97	Drupal, Joomla, Moodle and spip
/(thinkphp TP)/(public index)	55,144	3,608	ThinkPHP
/wp-content/plugins	32,917	2,416	WordPress
/solr/	23,307	919	Apache Solr
/manager/html	10,615	1,557	Tomcat Manager

Bot Intentions

Gray

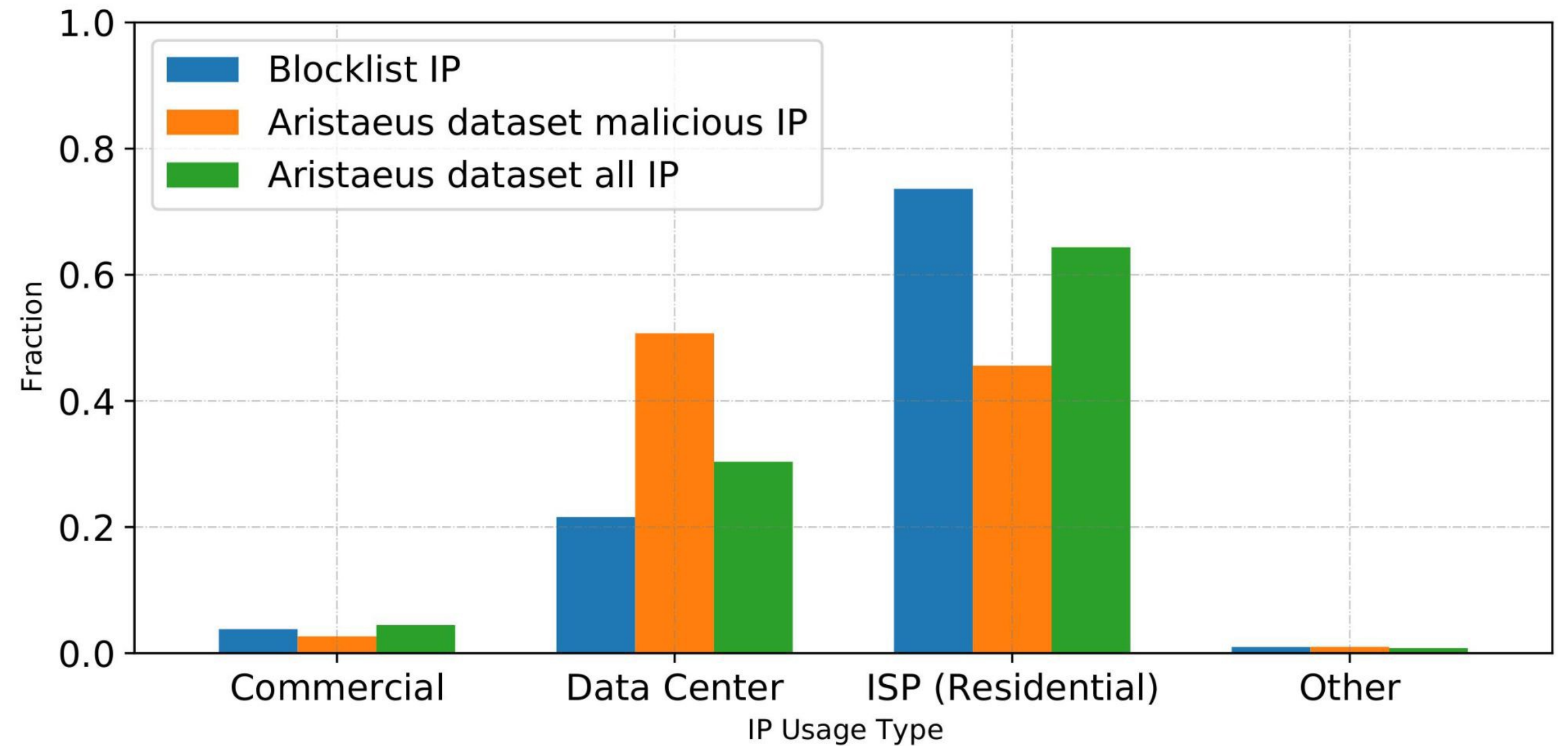
- **Single-shot scanners (50.04% of IP address)**
 - Visit the website only once, mostly asking “/”
 - No obvious activities.
 - Require future explorations

Only
13%

of Malicious bots appeared in online blocklists

Online Blocklist Coverage

- **Where are these bots?**
 - Commercial (<5%)
 - Datacenter (~30%)
 - Residential (~65%)
 - Other (<1%)



Javascript Support

Only
0.63%

Bots executed JavaScript

TLS Fingerprinting

35%

use HTTPS

(even though it's optional)

558

Fingerprints

14 →

Tools

97.2%

TLS Requests

TLS Fingerprinting

- **TLS fingerprints can be used to identify spoofing bots.**
- Search for mismatch between the stated UAs and the observed TLS fingerprints.
- E.g. Claim to be Firefox, but match TLS fingerprint of python-requests

Tools	Unique FPs	IP Count	Total Requests
Go-http-client	28	15,862	8,708,876
Libwww-perl or wget	17	6,102	120,423
PycURL/curl	26	3,942	80,374
Python-urllib 3	8	2,858	22,885
NetcraftSurveyAgent	2	2,381	14,464
msnbot/bingbot	4	1,995	44,437
Chrome-1(Googlebot)	1	1,836	28,082
Python-requests 2.x	11	1,063	754,711
commix/v2.9-stable	3	1,029	5,738
Java/1.8.0	8	308	1,710
MJ12Bot	2	289	28,065
Chrome-2(Chrome, Opera)	1	490	66,631
Chrome-3(Headless Chrome)	1	80	2,829
Chrome-4(coc_coc_browser)	1	4	101
Total	113	38,239	9,879,326


```
    "tlsfpinfo": {
      "tlsfp": {
        "ciphersuite": "0xC02F 0xC030 0xC02B 0xC02C 0xCCA8 0xCCA9 0xC013 0xC009 0xC014 0xC00A 0x009C 0x009D",
        "tls_version": "0x0303",
        "matchcount": "1",
        "sig_alg": "0x0401 0x0403 0x0501 0x0503 0x0601 0x0603 0x0201 0x0203 ",
        "src_port": 43482,
        "ipv4_dst": "172.26.13.77",
        "record_tls_version": "0x0301",
        "timestamp": "2020-04-25 03:59:59",
        "server_name": "www.objectivecurtainbook.com",
        "ipv4_src": "134.209.53.244",
        "ec_point_fmt": "0x00",
        "e_curves": "0x001D 0x0017 0x0018 0x0019 ",
        "compression": "0x00",
        "extensions": "0x0000 0x0005 0x000A 0x000B 0x000D 0xFF01 0x0012 ",
        "dst_port": 443,
        "compression_length": "1",
        "ciphersuite_length": "0x0020",
        "id": 0,
        "desc": "Dynamic ip-172-26-13-77.ec2.internal 12947 201"
      },
      "hastlsfp": true
    },
    "SSL_PROTOCOL": "TLSv1.2",
    "forensic": {
      "hasforensic": true,
      "flog": {
        "headersText": [
          "Host:www.objectivecurtainbook.com",
          "User-Agent:Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv%3a62.0) Gecko/20100101 Firefox/62.0",
          "Content-Type:application/x-www-form-urlencoded",
          "Cookie:csessid=5ea3edf88e02d; wordpress test cookie=WP+Cookie+check",
          "Referer:https%3a//www.objectivecurtainbook.com/wp-login.php",
          "Accept-Encoding:gzip",
          "Connection:close\n"
        ]
      }
    }
  ],
```

“Golang HTTP request”

“Firefox on Ubuntu”

Bots are pretending to be browsers

- **Fake Chrome (82.6%)**
 - Mostly curl/wget
 - Shown no GREASE in TLS fingerprint
- **Fake Firefox (98.5%)**
 - 68.7% are go-http-client
 - 21% are libwww-perl
 - Remaining requests are still not firefox

86.2% of bots claiming to be Firefox/Chrome, were lying about their identities

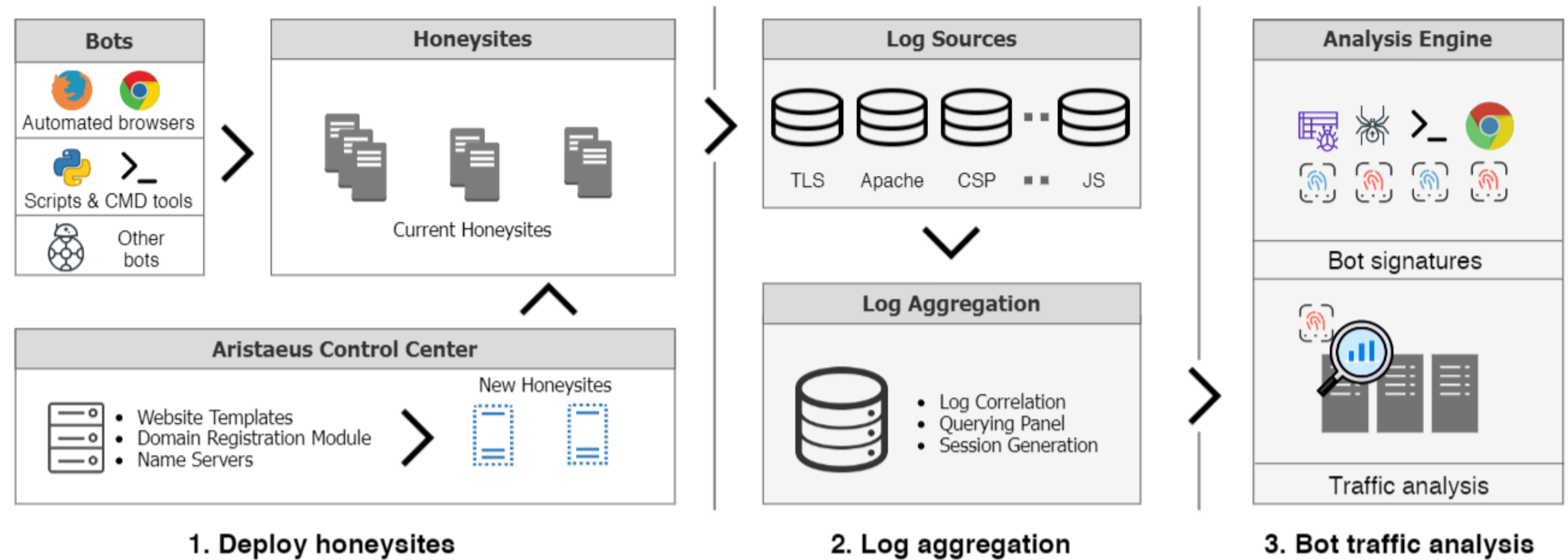
Bots scanning for vulnerabilities present distinct behavior

- Send a large number of requests.
- Distinct exploration and attack phases.
- May only use a subset of their attack vectors during each execution.
- Produce a large number of invalid requests.

Vulnerabilities are being quickly abused

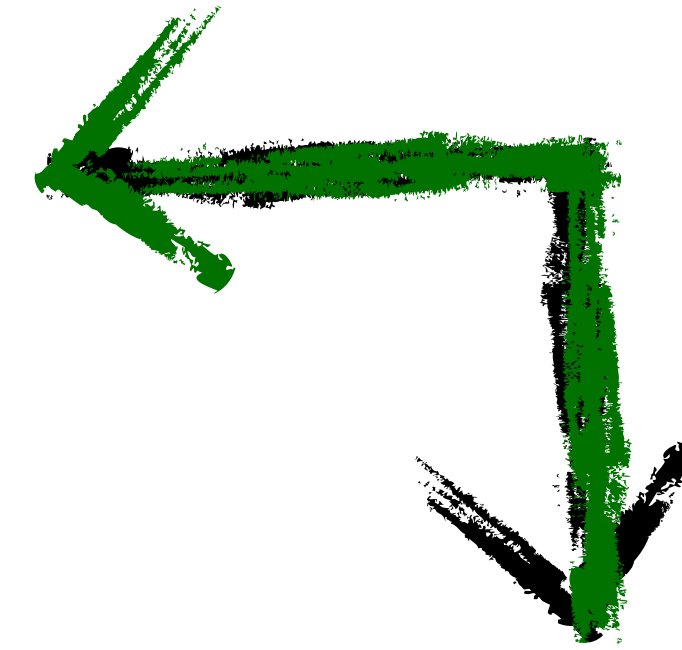
- **Netgear GPON router (EDB-48225), 0 days**
- **F5 TMUI shell (CVE-2020-5902), 0 days**
- **DrayTech modems (CVE-2020-8585), 3 days**

Takeaways



- By putting an unpopular website online, the website will receive at least 1200 requests/day, <2% are benign
- Bots are highly selective, targeting easy-to-exploit endpoints.
- 97% bots are rudimentary HTTP libraries, but pretending to be browsers
- Only 13% of bot IPs appeared in IP blocklists
- TLS fingerprinting are effective against cloaking and evasion
- Exploits that go public are quickly abused - Just in a few hours

How can we minimize the effect of malicious bots without hindering benign bots?



How can we understand the true impact and purpose of bots?



How can we build a bot-only dataset?

**Stop malicious bots from scanning
websites for vulnerabilities**

Web Vulnerability Scanners

Web vulnerability scanner (WVS)

- Automated, “point-and-click” tools that scan web applications for vulnerabilities.
- **Perfect tool for penetration testers**
 - Identify and **fix** low-hanging vulnerabilities
- **Full-auto weapon for malicious actors**
 - Identify and **exploit** low-hanging vulnerabilities

```
root@kali:~# commix --url="http://192.168.0.23/commix-testbed/scenarios/referenc

      v1.7-stable
      http://commixproject.com
      (@commixproject)

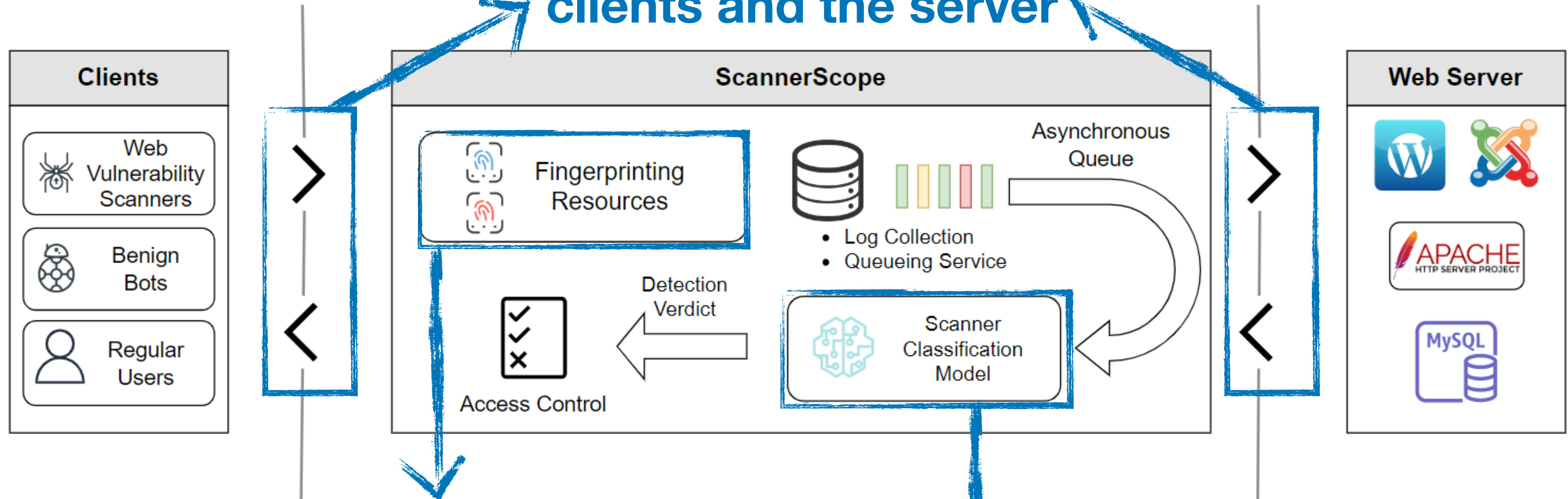
+--
Automated All-in-One OS Command Injection and Exploitation Tool
Copyright (c) 2014-2017 Anastasios Stasinopoulos (@ancst)
+--

[*] Checking connection to the target URL... [ SUCCEED ]
[*] Setting the HTTP header User-Agent for tests.
[*] Testing the (results-based) classic command injection technique... [ FAILED ]
[*] Testing the (results-based) dynamic code evaluation technique... [ FAILED ]
[*] Testing the (blind) time-based command injection technique... [ FAILED ]
[*] Trying to create a file in '/var/www/html/commix-testbed/scenarios/referenc
[!] Warning: It seems that you don't have permissions to read and/or write fil
[?] Do you want to try the temporary directory (/tmp/) [Y/n] > Y
```

Commix Scanner Example

ScannerScope Design

Mediate HTTP traffic between clients and the server



Fingerprinting Techniques from Aristaeus

Machine learning model classifies users vs. WVSs

Training the ML Model

- **159 Users are from Amazon Mechanical Turk.**
 - **Users are asked to perform series of interactions to the web application**
 - Reading articles, Posting comments, etc.
 - Actions are randomized so that no two users will behave the same.
- **12 Web Vulnerability Scanners**
 - **Top 10 open-source WVS of top OWASP pentesting tools**
 - OWASP Zap, Arachni, Commix, etc.
 - **2 academic scanners**
 - Black Widow, Enemy of the State

Scanner Name	Version
WPScan(kali)	3.8.13
Arachni	1.5.1
OWASP Zap	D-2020-12-21
WMap	1.5.1
Wapiti	3.0.3
Nikto	2.1.6
W3af	1.6.45
Skipfish (kali)	2.10b
Commix	2.9-stable
Google Tsunami	0.0.5
Black Widow	N/A
Enemy of the State	N/A

Eriksson et al., Black widow: Blackbox data-driven web scanning. IEEE S&P 2021

Doupé et al., Enemy of the state: A state-aware black-box web vulnerability scanner, Usenix Security 2012

ScannerScope Performance

Model	Accuracy	Precision	Recall	F1-score
WordPress-WordPress	99.30%	97.79%	99.58%	98.66%
Joomla-Joomla	99.22%	99.17%	99.14%	99.15%
WordPress-Joomla	91.44%	92.52%	91.44%	91.39%
1 Unseen Scanner	98.27%	96.71%	98.53%	97.43%
4 Unseen Scanners	96.20%	93.65%	97.13%	95.19%
6 Unseen Scanners	91.26%	85.50%	94.38%	87.66%

ScannerScope Performance

Model	Accuracy	Precision	Recall	F1-score
WordPress-WordPress	99.30%	97.79%	99.58%	98.66%
Joomla-Joomla	99.22%	99.17%	99.14%	99.15%
WordPress-Joomla	91.44%	92.52%	91.44%	91.39%
1 Unseen Scanner	98.27%	96.71%	98.53%	97.43%
4 Unseen Scanners	96.20%	93.65%	97.13%	95.19%
6 Unseen Scanners	91.26%	85.50%	94.38%	87.66%

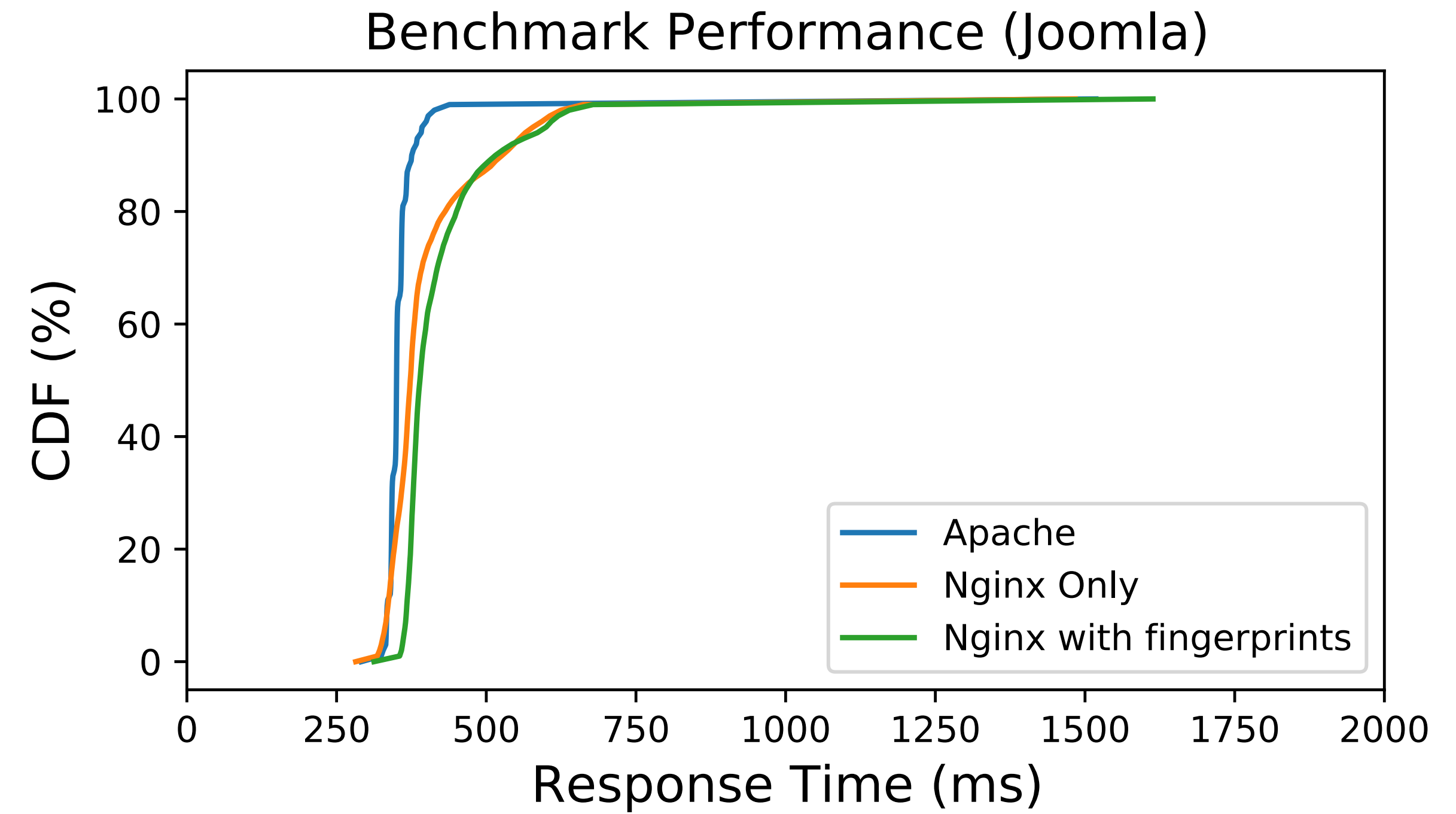
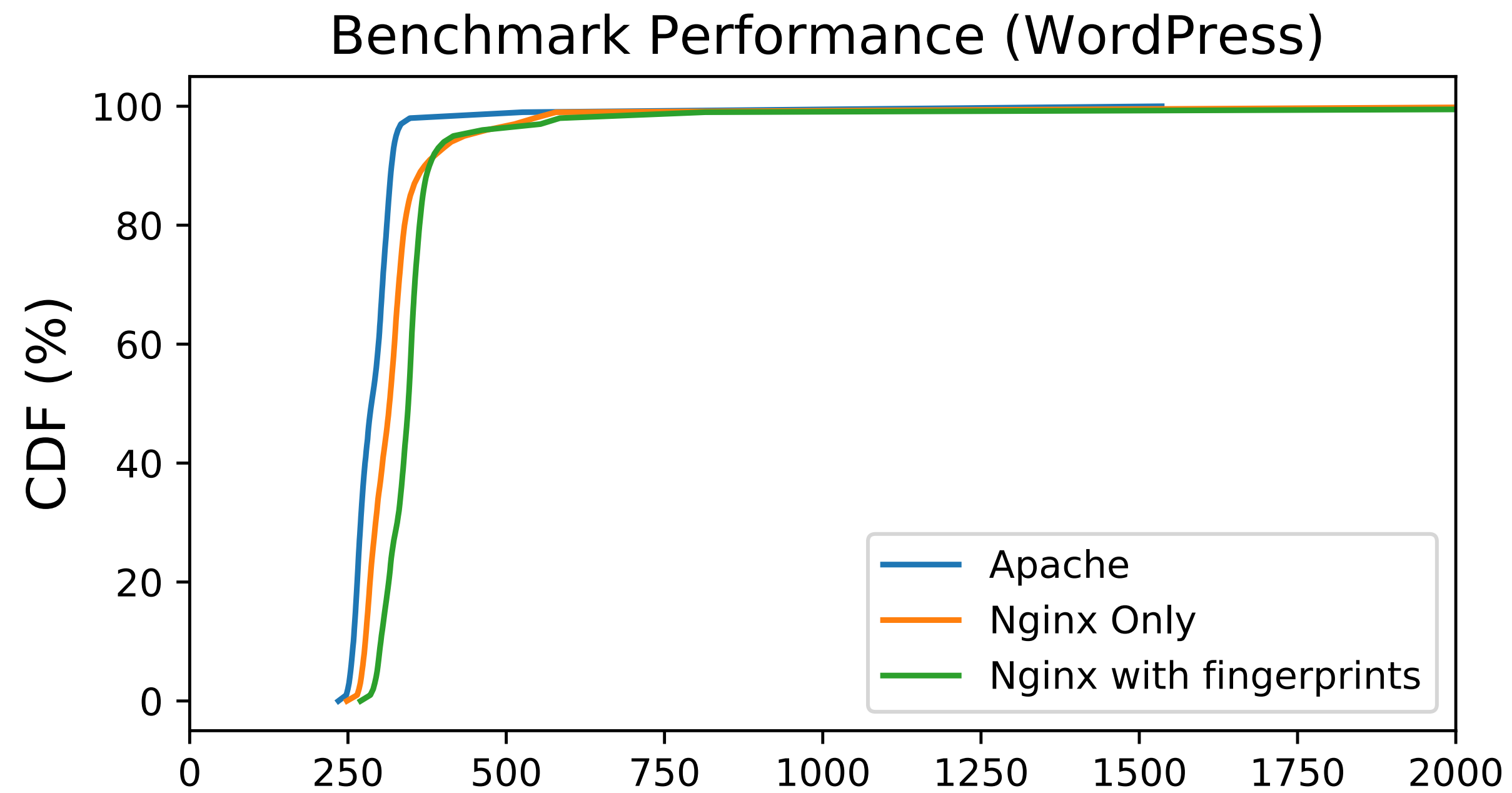
ScannerScope Performance

Model	Accuracy	Precision	Recall	F1-score
WordPress-WordPress	99.30%	97.79%	99.58%	98.66%
Joomla-Joomla	99.22%	99.17%	99.14%	99.15%
WordPress-Joomla	91.44%	92.52%	91.44%	91.39%
1 Unseen Scanner	98.27%	96.71%	98.53%	97.43%
4 Unseen Scanners	96.20%	93.65%	97.13%	95.19%
6 Unseen Scanners	91.26%	85.50%	94.38%	87.66%

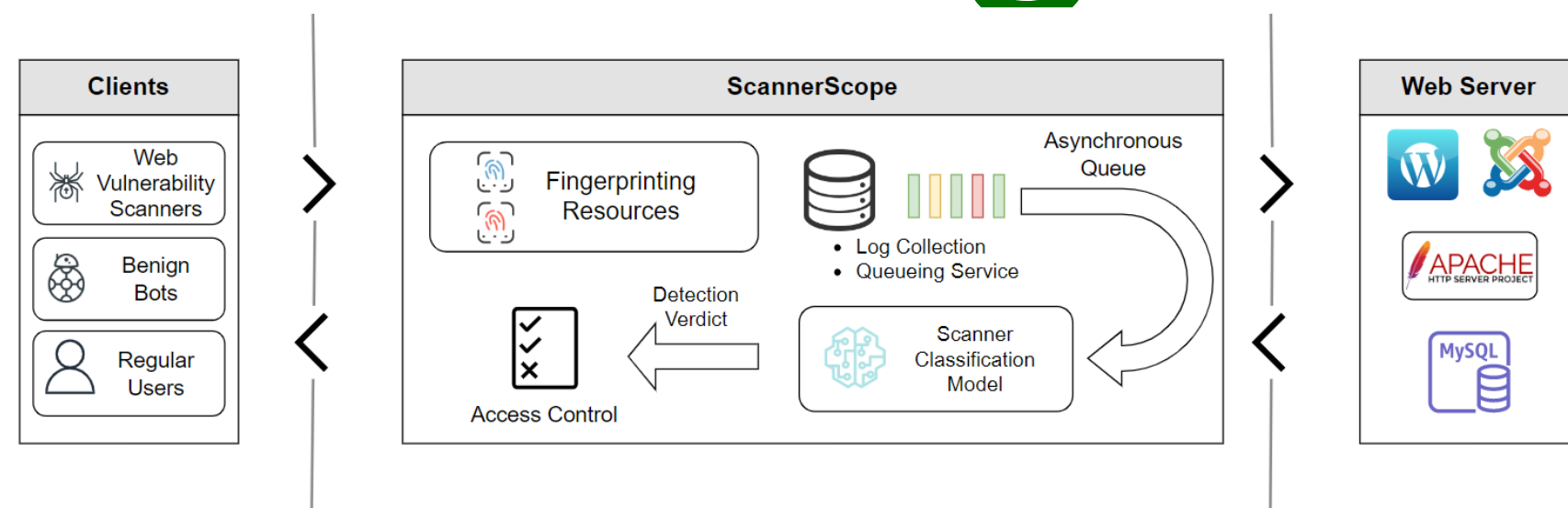
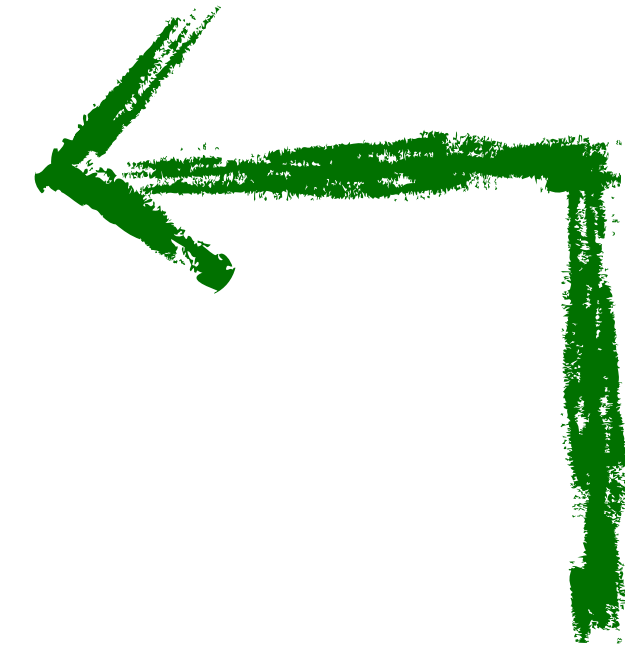
of requests to detect: 15

99.27% Accuracy on Benign Bots

ScannerScope Overhead



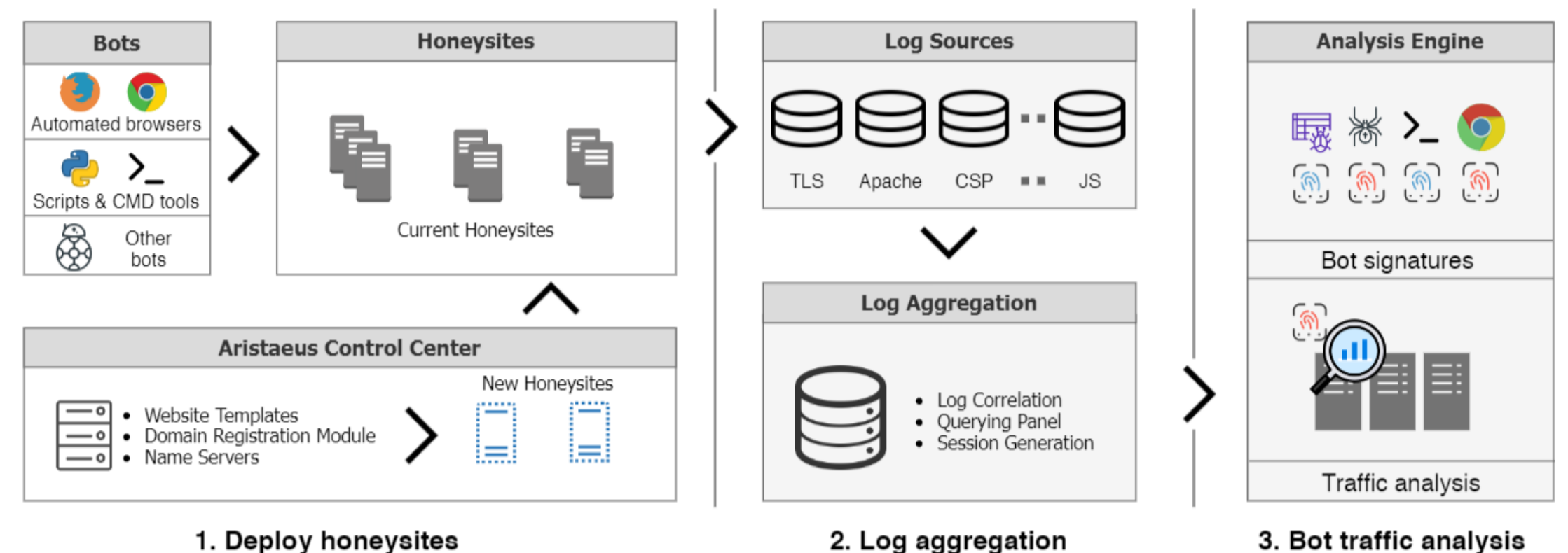
How can we minimize the effect of malicious bots without hindering benign bots?



How can we understand the true impact and purpose of bots?



How can we build a bot-only dataset?



Beyond Attacking Web Servers

Targeting users instead of servers

The image shows a screenshot of a WhatsApp chat interface. On the left, a user named 'WhatsApp' (with a verified badge) has sent a message containing a sequence of numbers: 1, 2, 5, 6, 3, 2, 0, 9, 5, 7, 8. Below the numbers are two blue envelope icons and two yellow hand icons. A red arrow points from the text 'Author Impersonation' to the number sequence. Below the message are thumbs up and thumbs down icons, and a 'Reply' button. On the right, three users have responded:

- Jennifer Alberto: "You invest with Mrs Luciana cruz too? Wow that woman has be and my family." (A red arrow points from the text 'Scripted conversation Within a few seconds' to this message.)
- Norbert Stephan: "I'm new at this, please how can I reach her?" (A red arrow points from the text 'Scripted conversation Within a few seconds' to this message.)
- albert john: "You can reach her on her TELEGAM with the user name below" (A red arrow points from the text 'Scripted conversation Within a few seconds' to this message.)
- albert john: ".investwithLucruz." (A red arrow points from the text 'Scripted conversation Within a few seconds' to this message.)

At the bottom left, another user 'Andrei Jikh' (with a verified badge) has sent a message: "thank you for the kind words!". Below it are thumbs up and thumbs down icons, and a 'Reply' button.

Data Collection

- Measurement range: 10/1/2022 to 03/31/2023
- Monitored Channels: 20
- Videos: 8,226
- Captured comments: 8.8 Million

Comment Scam Features

- **Textual** - Scammers use Visually Similar Symbols (VSS) to evade automated detection systems
- **Graphical** - Scammers apply similar profile images to impersonate channel owners
- **Temporal** - Scammers split the conversation and even contact phone numbers, and use multiple accounts to post them together to form a fabricated short story

Comment Scam Features

The image shows a screenshot of social media comments with several red annotations. On the left, a comment from 'WhatsApp' is annotated with 'Author Impersonation' pointing to the name and a series of numbers in circles. Below it, a comment from 'Andrei Jikh' is shown. On the right, three comments from 'Jennifer Alberto', 'Norbert Stephan', and 'albert john' are shown. A red line connects these three comments to the annotation 'Scripted conversation Within a few seconds'.

WhatsApp + 1 2 5 6 3 2 0 9 5 7 8
Author Impersonation

Andrei Jikh ✓ 4 hours ago
thank you for the kind words!

Jennifer Alberto
You invest with Mrs Luciana cruz too? Wow that woman has be and my family.

Norbert Stephan
I'm new at this, please how can I reach her?

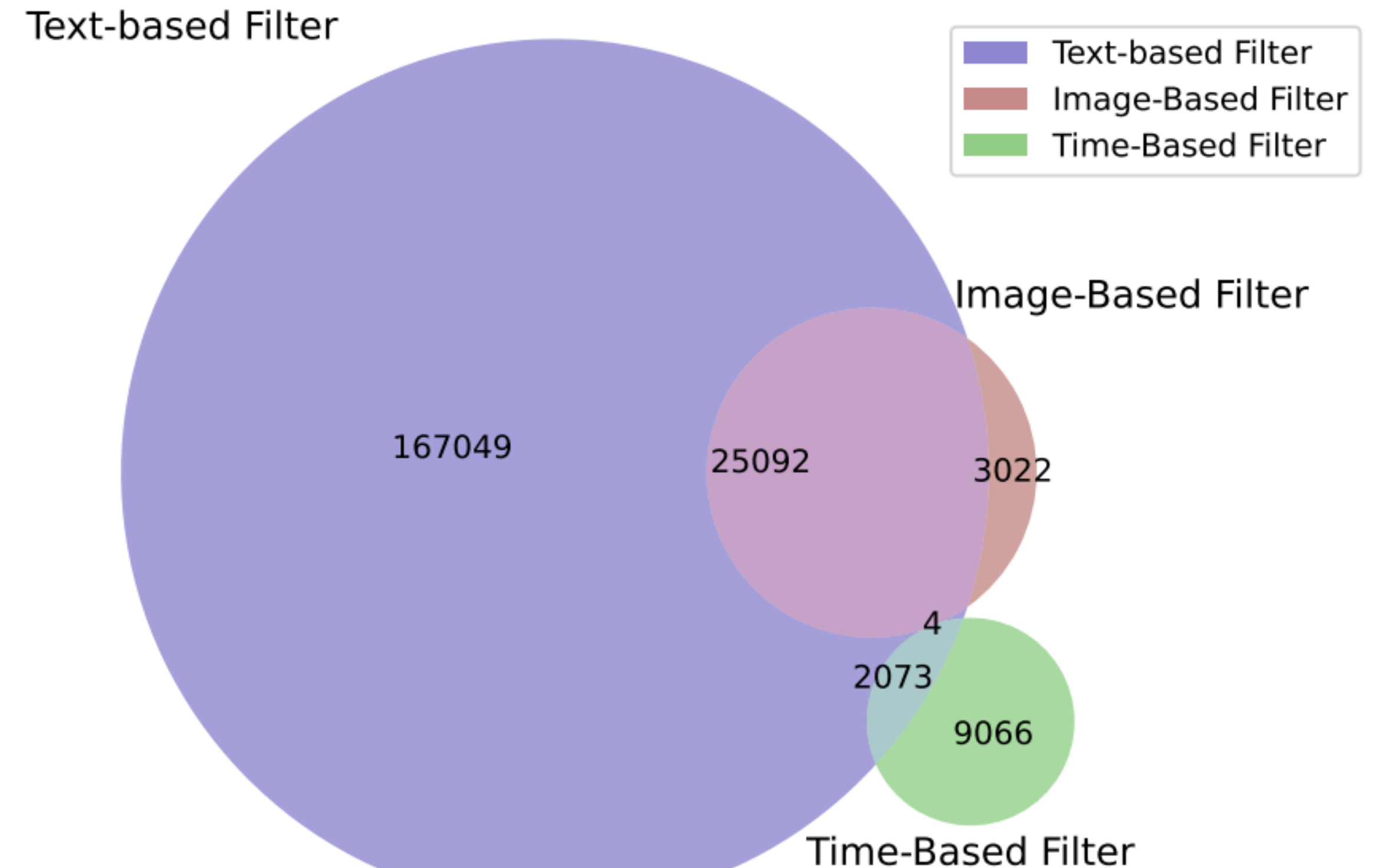
albert john
You can reach her on her TELEGAM with the user name below

albert john
.investwithLucruz.

Scripted conversation
Within a few seconds

Comment Scam Features

- **Textual** - Scammers use Visually Similar Symbols (VSS) to evade automated detection systems
- **Graphical** - Scammers apply similar profile images to impersonate channel owners
- **Temporal** - Scammers split the conversation and even contact phone numbers, and use multiple accounts to post them together to form a fabricated short story



Flagged 206K (2.34%) of comments as scam

Scam Campaigns

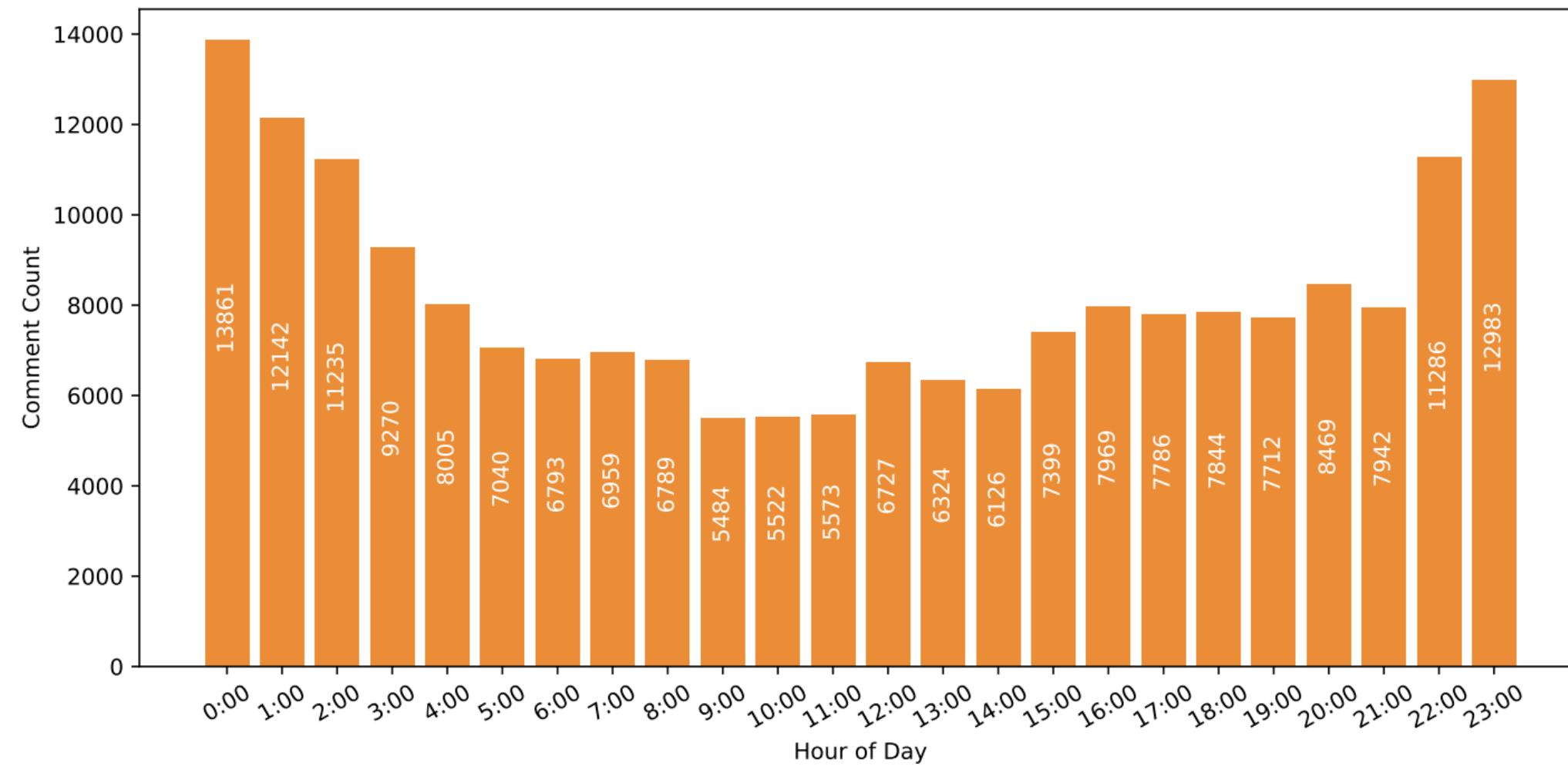
Campaign ID	Accounts	Comments Posted	Affected Videos	Targeted Channels	Affected Categories
1	112	4045	92	1	Finance
2	59	703	324	4	News/Politics, Finance
3	46	5405	66	2	Finance
4	45	692	321	4	News/Politics, Finance
5	44	5662	76	2	Finance

Only 31.42% scam accounts were deactivated during study

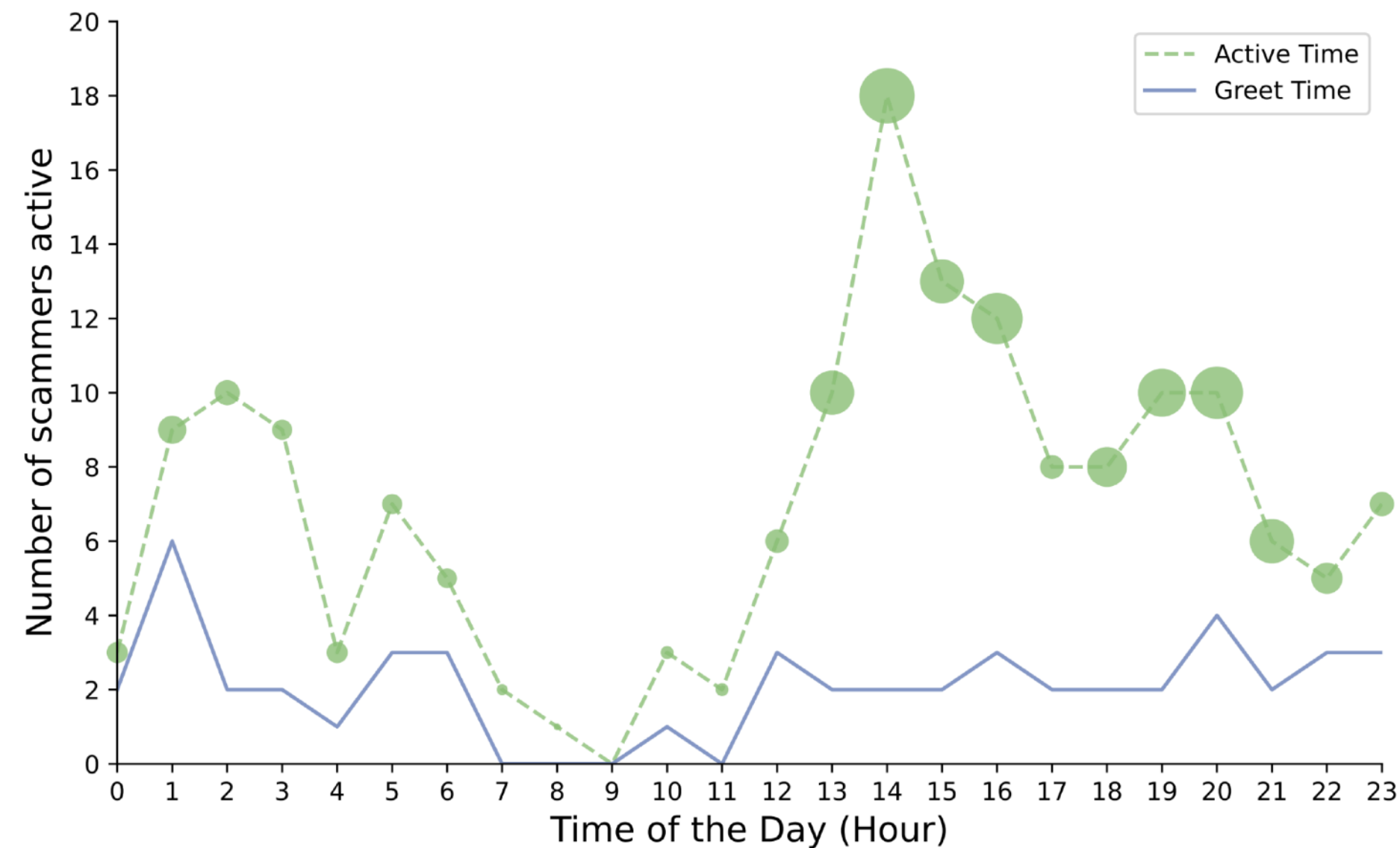
Interacting with Scammers

- **Cryptocurrency Investment (76%)**
 - Promise unrealistic high-yield investments (15% to 1300% weekly return)
 - Impersonation as channel owner or broker
 - Entice user to transfer cryptocurrency to scammer's wallet
- **Fake Prize (22%)**
 - Promise a prize (usually related to channel content)
 - Request shipping charges (\$50 to \$500)
- **Others (2%)**

Interacting with Scammers



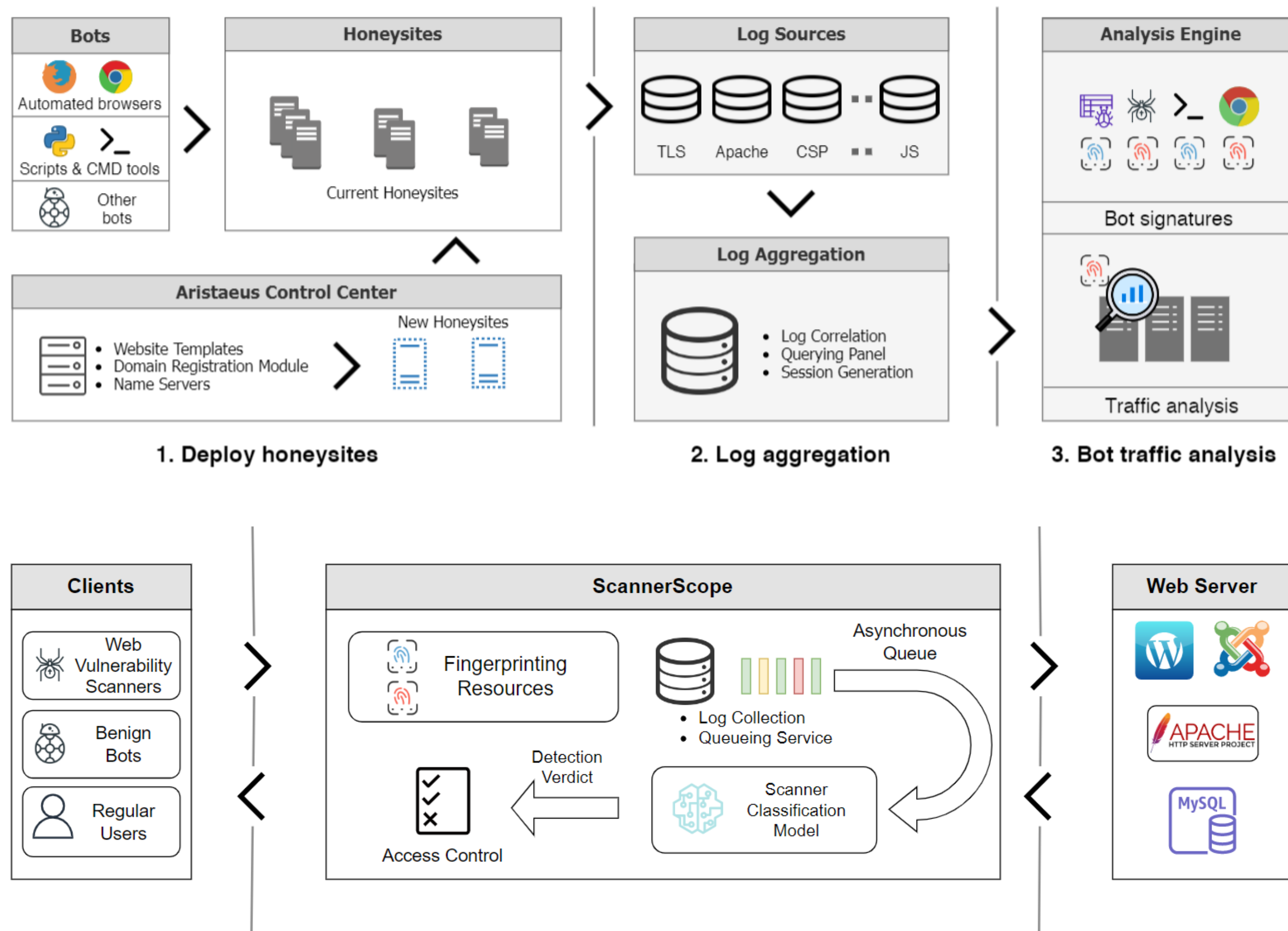
Crypto-currency	# of Wallets	Total Amount of Cryptocurrency	USD Value (Min. - Max.)
Bitcoin (BTC)	31	67.64	\$1.07M - \$1.92M
Ethereum (ETH)	16	36.49	\$0.04M - \$0.07M
(Total)	47	-	\$1.11M - \$1.99M



Millions of dollars (equivalent) were stolen by only 31 scammers

Security and Privacy in an Everchanging System Landscape

Amir Rahmati
Stony Brook University
<https://amir.rahmati.com>



accelerating embedded model medical transfer
malware comprehensive cells
clocks physical application feasibility approximate
information sram data bot framework study
power devices iot adversarial research
domain systems integrity platforms training
frameworks towards visual robustness security dram
attention state attacks smart compressing
knowledge permission retention certified