Cookietective

Automated Cookie Vulnerability Scanner

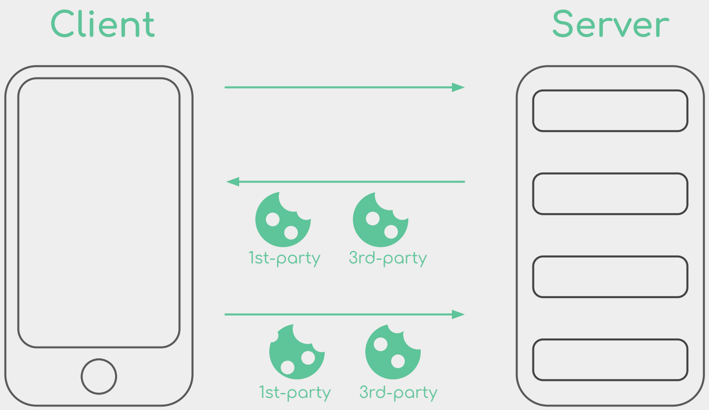
Timothy Borunov
Andres Garcia
Cole Wentzel
Jeffrey Chen
Trevor Chan

```
      dbFile='../database.db
    ath.abspath(os.path.join(os.path.dirn
    qlite3.connect(dbFile)
    onn.cursor()
   execute("SELECT hasAType FROM CNAMEpackets WHERE domainName
  result = c.fetchone()
  conn.close()
  if result is not None:
      return result[0]
  return None

def hasCNAMErecord(domainName, dbFile='../database.db'):
    dbFile = os.path.abspath(os.path.join(os.path.dirname(__file__), dbFile))
    conn = sqlite3.connect(dbFile)
    c = conn.cursor()
    c.execute("SELECT EXISTS(SELECT 1 FROM CNAMEpackets WHERE domainName = ?
    result = c.fetchone()[0]
    conn.close()
    return result

def firstPartyCheck(domainName, dbFile='../database.db'):
    dbFile = os.path.abspath(os.path.join(os.path.dirname(__file__), dbFile
    conn = sqlite3.connect(dbFile)
    c = conn.cursor()
    c.execute(f"SELECT CNAMEAlias FROM CNAMEpackets WHERE domainName='
    result = c.fetchone()
    onn.close()

      lt is None:
          0
```

```
    ference("network.
.set_preference("permissions.defa
opts.set_preference("http.response.timeout

driver = webdriver.Firefox(options=opts)

def scrape_links(url, stop_time):
    queue = set()

    try:
        driver.get(url)

        WebDriverWait(driver, scan_time).until(
            EC.presence_of_element_located((By.TAG_NAME, "
        )

        links = driver.find_elements(By.TAG_NAME, "a")

        for link_to_add in links:
            href = link_to_add.get_attribute("href"

            if href and href.startswith("http
                queue.add(href)
```

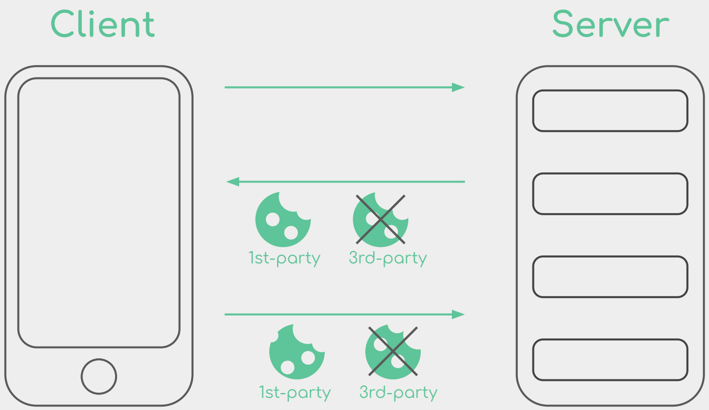
```
  E_packet():
  f __init__(self, packet, A, ip):
        self.packet = packet
        self.has_A = A
        self.domain = packet.rrname
        self.cname = packet.rdata
        if len(ip) < 1:
            self.ip = None
        else:
            self.ip = ip

def parseDNS(packet):
    if (packet and packet.haslayer('DNS') and packet.haslayer('UDP'
        udp = packet['UDP']
        dns = packet['DNS']

        has_CNAME = False
        has_Atype = False
        src_ip = packet['IP'].src

        CNAME_index = 0
        ip = []

        if (int(udp.sport) == 53):
            for i in range(dns.ancount):
                dnsrr = dns.an[i]
                    ield = dnsrr.get field(
```

Background



- **Cookies Often Used For:**
  - Session ID
  - Tracking / Advertising Services

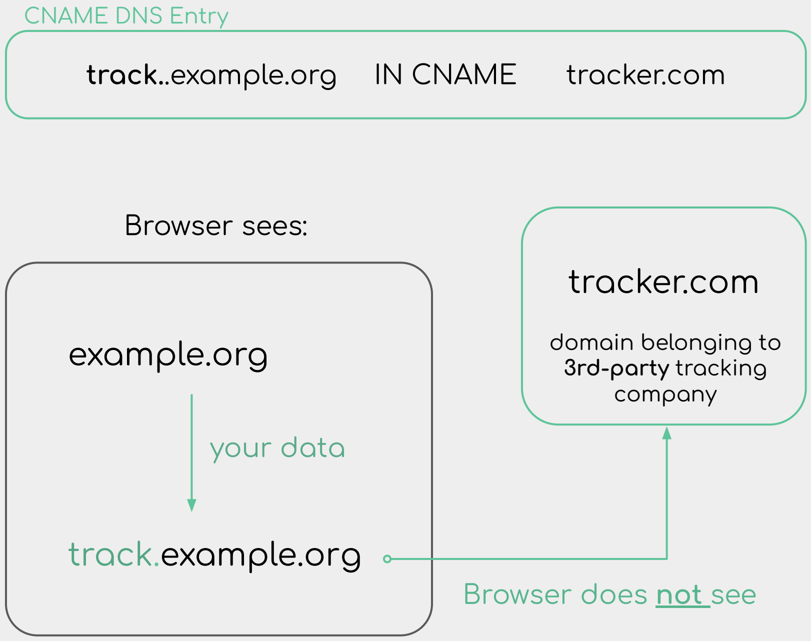
Background



- Cookies Often Used For:
  - Session ID
  - Tracking / Advertising Services
- Modern browsers & extensions:
  - Increasingly block T/A cookies
  - Prevent 3rd party tracking
- T/A services want to bypass
  - Can lead to security vulnerabilities

What is CNAME Cloaking?

- **CNAME DNS entry aliases:**
  - Domain (what it actually is)
  - CNAME (what browser sees)
- **CNAME Cloaking**
  - Disguise 3rd-party (foreign host) as 1st-party (original domain)
  - T/A cookies **trusted as 1st party**
- Can cause severe security vulnerabilities



Vulnerability

- Leaks cookies to 3rd party if:
  - CNAME cloaked
  - Lax cookie settings (`Domain`)
- **Any** 3rd party admin can:
  - Access leaked cookies
  - Potentially access **session info**

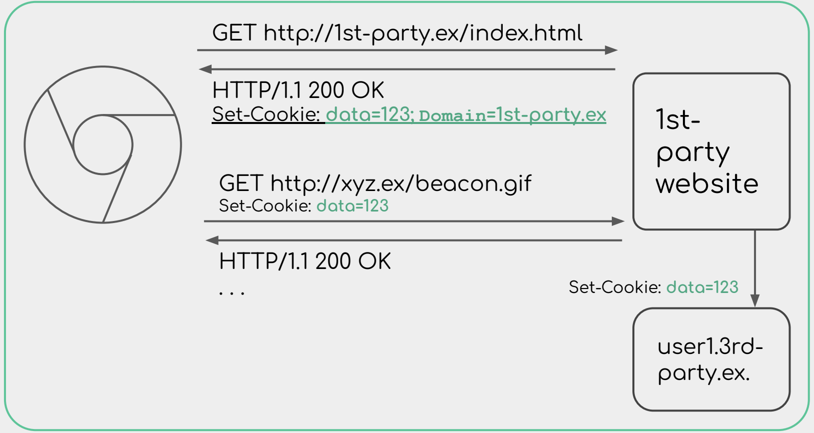
DNS resource records registered by first-party webmaster

1st-party.ex.	IN A	192.168.01
xyz.1st-party.ex.	IN CNAME	user1.3rd-party.ex.

DNS resource records registered by third-party vendor

user1.3rd-part.ex.	IN A	172.16.0.1.
user2.3rd-part.ex.	IN A	172.16.0.1.
user3.3rd-part.ex.	IN A	172.16.0.1.
. . .		

First-party cookie shared by CNAME cloaking



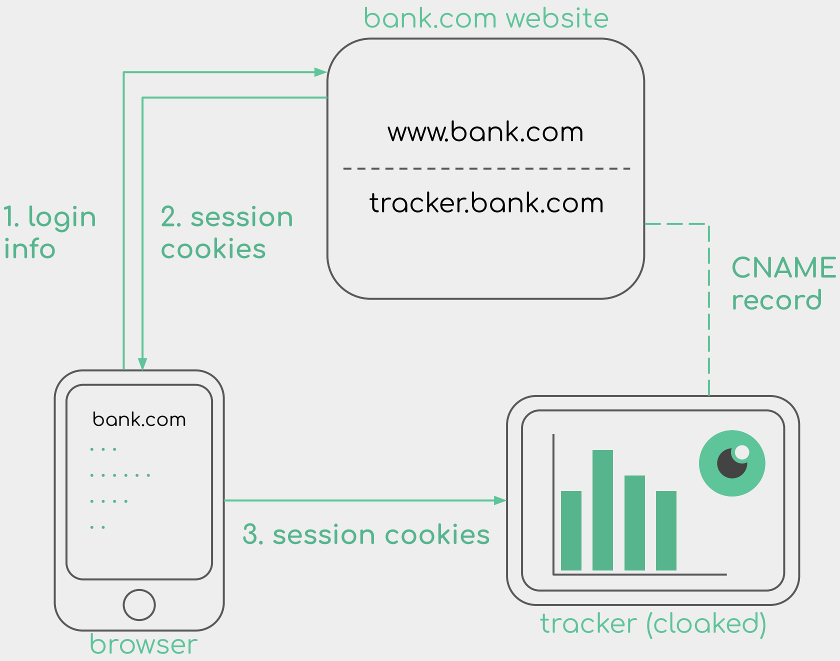
GET http://1st-party.ex/index.html

HTTP/1.1 200 OK
Set-Cookie: data=123; `Domain`=1st-party.ex

GET http://xyz.ex/beacon.gif
Set-Cookie: data=123

HTTP/1.1 200 OK
. . .

Why You Should Be Concerned



- Possibility of **session respring**
- Multiple widely-used banking sites found to be vulnerable
- Easy to set up, easy to miss, **drastic consequences**

Solution: Cookietective

Automated Scanner

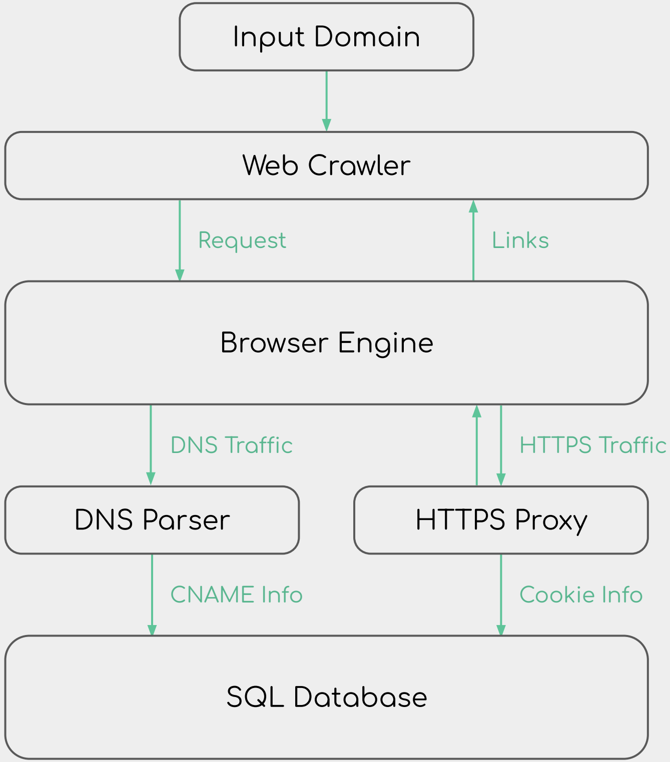
1. **Information Gathering:**
   - Parse through websites
   - Scan for CNAME cloaking and Cookie information

2. **Analysis:**
   - **Label domains as 1st-party or 3rd-party**
   - If 3rd party, scan if `Domain` setting leaks cookies

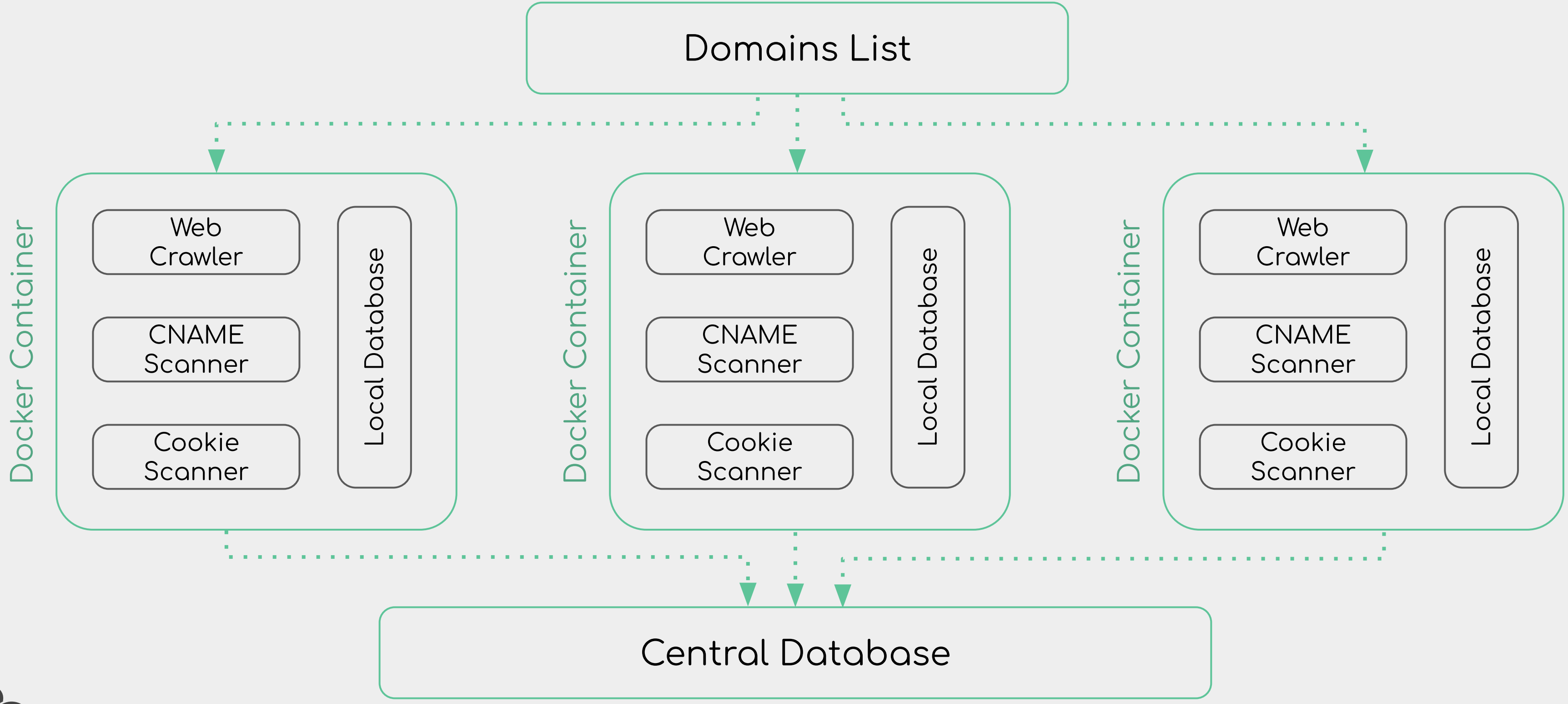
3. **Measure Accuracy:**
   - Check if domain name is present in Majestic Million or NoTracking lists

1. Information Gathering

- **Web Crawler**
  - Send request to input domain
  - Locate and traverse links in BFS fashion
- **Traffic Parser scans:**
  - DNS traffic for CNAME packets
  - HTTPS traffic for Cookie settings
- **Records:**
  - CNAME alias, domain
  - Set-cookie settings
  - Original Domain being scanned



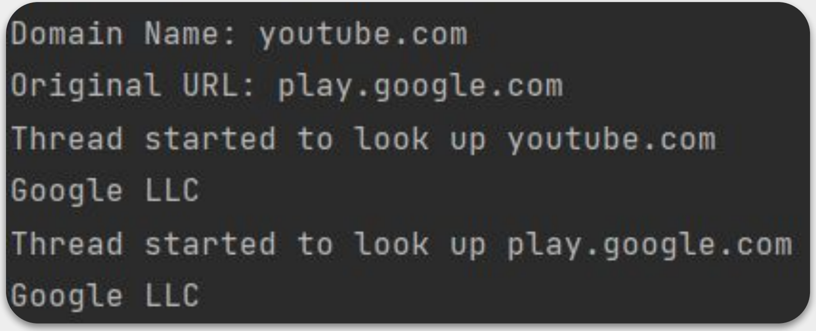
Scaling Crawler & Scanner



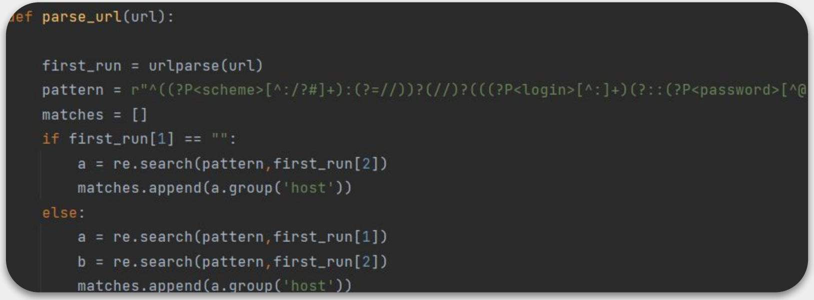
2. Analysis

- **WHOIS data**
  - Provides domain owner info
  - If owners match, **label 1st party**
  - If owners mismatch, **label 3rd party**

- **URL Parser**
  - Check Domain vs Original URL
  - Check CNAME vs Original URL
  - If mismatch, scan Cookie settings for vulnerabilities

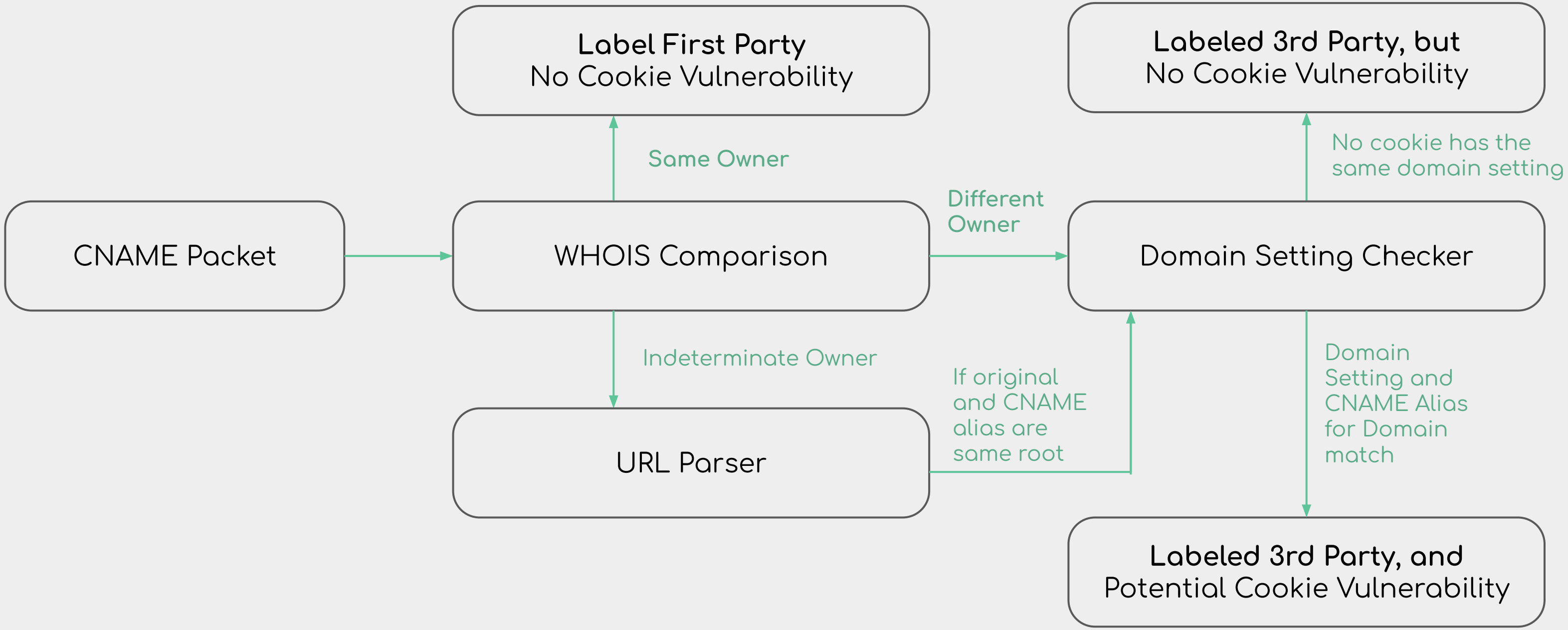


```
Domain Name: youtube.com
Original URL: play.google.com
Thread started to look up youtube.com
Google LLC
Thread started to look up play.google.com
Google LLC
```

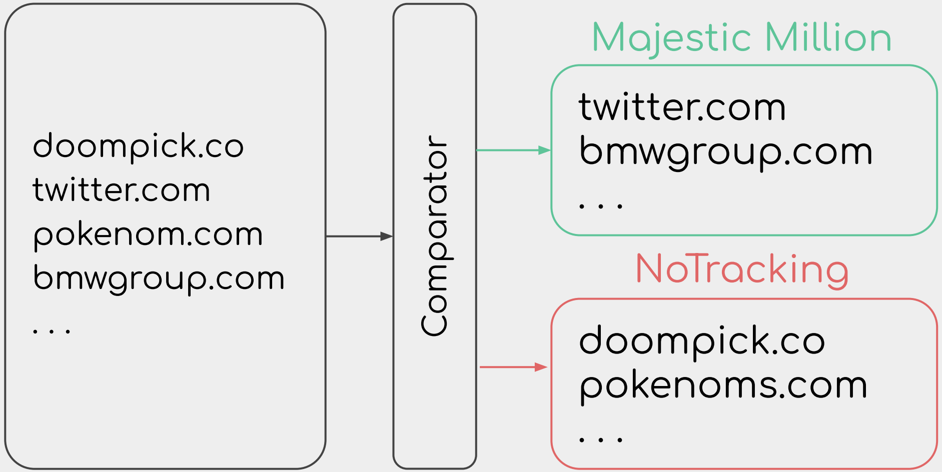


```
def parse_url(url):

    first_run = urlparse(url)
    pattern = r"^((?P<scheme>[^:/?#]+):(?=//))?(//)?(((?P<login>[^:]+)(?::(?P<password>[^@
    matches = []
    if first_run[1] == "":
        a = re.search(pattern,first_run[2])
        matches.append(a.group('host'))
    else:
        a = re.search(pattern,first_run[1])
        b = re.search(pattern,first_run[2])
        matches.append(a.group('host'))
```

Analysis Explanation



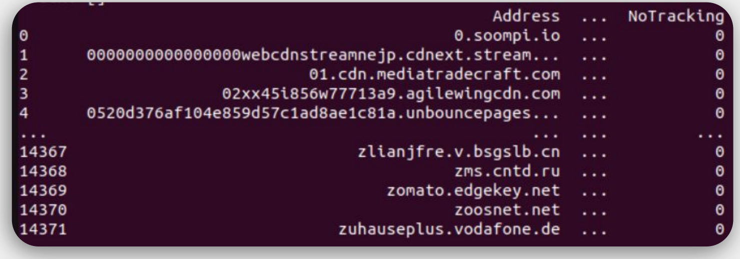
3. Accuracy Measuring



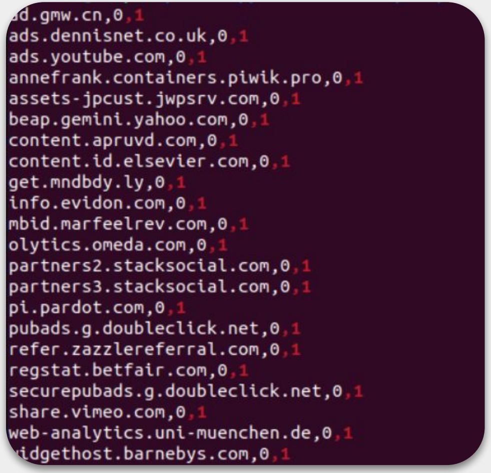
doompick.co
twitter.com
pokenom.com
bmwgroup.com
. . .

twitter.com
bmwgroup.com
. . .

doompick.co
pokenoms.com
. . .



```
                                                     Address  ...  NoTracking
0                                               0.soompi.io  ...           0
1      0000000000000000webcdnstreamnejp.cdnext.stream...  ...           0
2                                  01.cdn.mediatradecraft.com  ...           0
3                           02xx45i856w77713a9.agilewingcdn.com  ...           0
4      0520d376af104e859d57c1ad8ae1c81a.unbouncepages...  ...           0
...                                                      ...  ...         ...
14367                                    zlianjfre.v.bsgslb.cn  ...           0
14368                                            zms.cntd.ru  ...           0
14369                                       zomato.edgekey.net  ...           0
14370                                           zoosnet.net  ...           0
14371                                 zuhauseplus.vodafone.de  ...           0
```



```
d.gmw.cn,0,1
ads.dennisnet.co.uk,0,1
ads.youtube.com,0,1
annefrank.containers.piwik.pro,0,1
assets-jpcust.jwpsrv.com,0,1
beap.gemini.yahoo.com,0,1
content.apruvd.com,0,1
content.id.elsevier.com,0,1
get.mndbdy.ly,0,1
info.evidon.com,0,1
mbid.marfeelrev.com,0,1
olytics.omeda.com,0,1
partners2.stacksocial.com,0,1
partners3.stacksocial.com,0,1
pi.pardot.com,0,1
pubads.g.doubleclick.net,0,1
refer.zazzlereferral.com,0,1
regstat.betfair.com,0,1
securepubads.g.doubleclick.net,0,1
share.vimeo.com,0,1
web-analytics.uni-muenchen.de,0,1
idgethost.barnebys.com,0,1
```

Results

Scanner Results

Chart 1: Scanned Domains

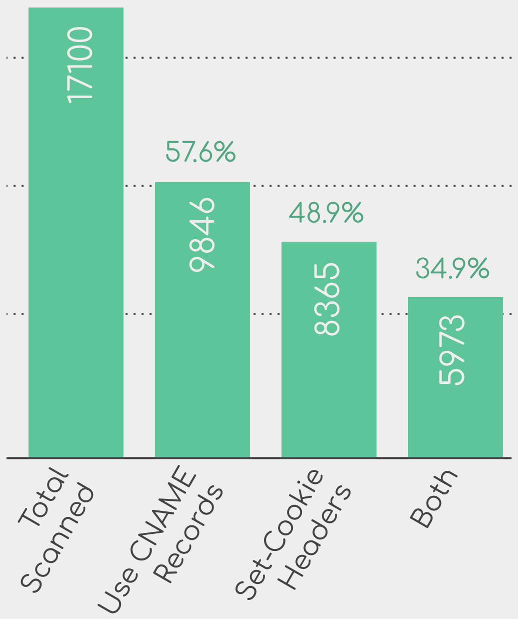


Table 2: Collected CNAME Packets

URLs with CNAME Aliasing	CNAME Entries	Distinct CNAME Aliased Domains
9846	15381	14372

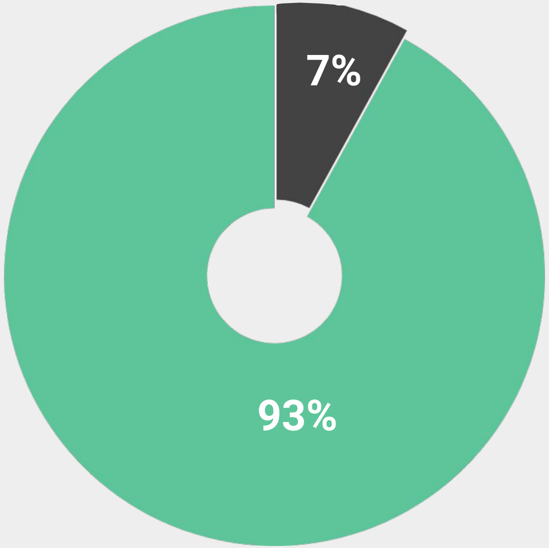
Table 3: Collected Set-Cookie Information

URLs w/ Set - Cookie Headers	Cookies Received	Specified `Domain` Attributes
8365	25758	7362

Analysis Results

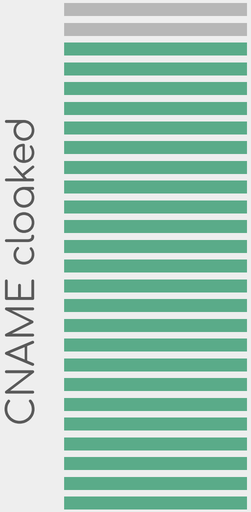
- Ran multiple analyses on the database
  - Consistent results with ~450 vulnerabilities found
- 5751 URLs contain 3rd party CNAME cloaking
  - 33.6% of the domains scanned
- 419 unique URLs are vulnerable
  - 7.28% of the domains utilize CNAME cloaking

% Websites that use CNAME Cloaking with Vulnerabilities



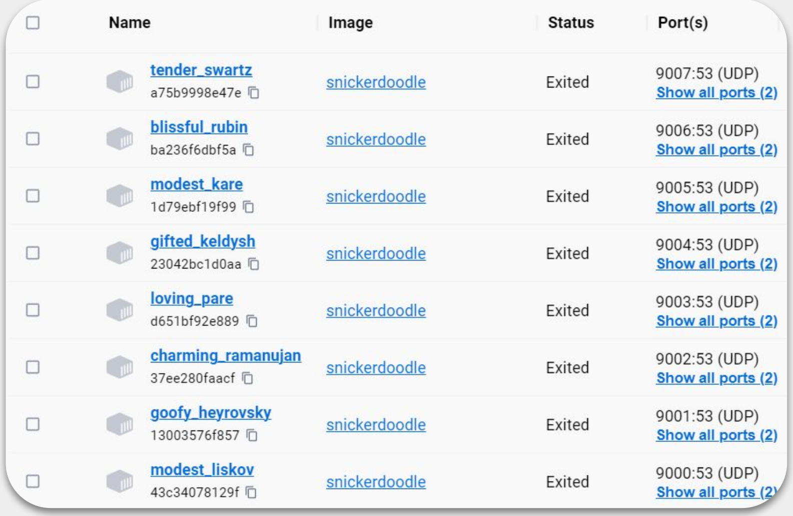
Accuracy Measurement Results

- 24 / 26 CNAME aliased domains on notrack marked
    - 92.3% of **known** T/A services labeled as CNAME cloaked
- 417 domains marked as using CNAME Cloaking **not** listed on the notracking list
- 7 on MM and **not** on NT labeled as vulnerable
    - Base assumption by paper
    - Could represent inaccuracies in analysis

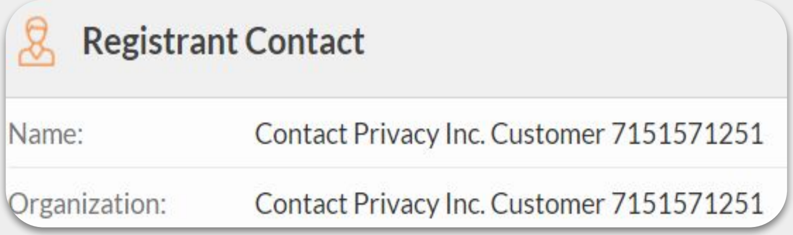


Limitations

- Computational:
  - Number of containers/workers
  - Scan time per domain
  - Resource allocation
- 1st vs 3rd Party Categorizing:
  - WHOIS data can be hidden by domain owner
  - Limited by WHOIS server speed
  - Parser may falsely flag related domains that look different



□	Name	Image	Status	Port(s)
□	tender_swartz a75b9998e47e	snickerdoodle	Exited	9007:53 (UDP) Show all ports (2)
□	blissful_rubin ba236f6dbf5a	snickerdoodle	Exited	9006:53 (UDP) Show all ports (2)
□	modest_kare 1d79ebf19f99	snickerdoodle	Exited	9005:53 (UDP) Show all ports (2)
□	gifted_keldysh 23042bc1d0aa	snickerdoodle	Exited	9004:53 (UDP) Show all ports (2)
□	loving_pare d651bf92e889	snickerdoodle	Exited	9003:53 (UDP) Show all ports (2)
□	charming_ramanujan 37ee280faacf	snickerdoodle	Exited	9002:53 (UDP) Show all ports (2)
□	goofy_heyrovsky 13003576f857	snickerdoodle	Exited	9001:53 (UDP) Show all ports (2)
□	modest_liskov 43c34078129f	snickerdoodle	Exited	9000:53 (UDP) Show all ports (2



Registrant Contact

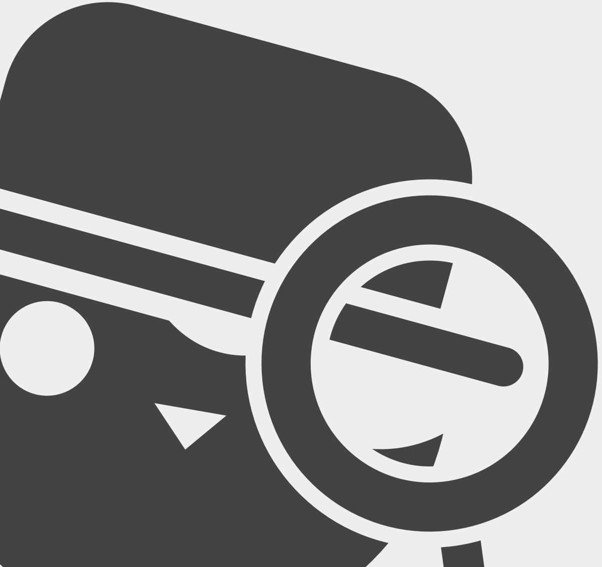
Name:	Contact Privacy Inc. Customer 7151571251
Organization:	Contact Privacy Inc. Customer 7151571251

Final Insights

- Previous paper found ~21.2% of websites used CNAME Cloaking
  - We found ~33.6%
  - Similar and could mean more websites are utilizing CNAME cloaking
- Previous paper found 50% of websites with CNAME cloaking were vulnerable
  - We found only 7.28%
  - Website owners may be more aware

Thank you!

Any questions?



Sources

- Risk Analysis of Cookie Sharing by Link Decoration and CNAME Cloaking:
  - *https://www.jstage.jst.go.jp/article/ipsjjip/29/0/29_649/_pdf/-char/en*
- Oversharing is Not Caring: How CNAME Cloaking Can Expose Your Session Cookies:
  - *http://megele.io/cname_cloaking-asiaccs2021.pdf*
- The CNAME of the Game: Large-scale Analysis of DNS-based Tracking Evasion:
  - *https://arxiv.org/pdf/2102.09301.pdf*
- Cookie Synchronization: Everything You Always Wanted to Know But Were Afraid to Ask:
  - *https://arxiv.org/pdf/1805.10505.pdf*
- CNAME Cloaking: Disguising Third Parties Through the DNS:
  - *https://unit42.paloaltonetworks.com/cname-cloaking/*