

DOING RECON LIKE IT'S 2017

APPSECCO

BHARATH KUMAR

BSIDES DELHI | OCTOBER 27, 2017



ABOUT ME

- Bharath Kumar
- Security Engineer @Appsecco
- **Offensive Security Certified Professional**
- I enjoy good books, coffee, camping and stargazing!

DEMO ENVIRONMENT

- Feel free to run the DNSSEC attacks from the talk against the following nameserver & domain:

Nameserver: **ns1.insecuredns.com**

Domain: **insecuredns.com**

WHAT IS RECONNAISSANCE?

Reconnaissance is the act of gathering preliminary data or intelligence on your target. The data is gathered in order to better plan for your attack. Reconnaissance can be performed actively or passively.

WHAT DO WE LOOK FOR DURING RECON?

- Info to increase attack surface(domains, net blocks)
- Credentials(email, passwords, API keys)
- Sensitive information
- Infrastructure details

WHAT'S COVERED IN THIS TALK?

1. Certificate Transparency for recon
2. DNSSEC Zone Walking
3. Hunting for publicly accessible on cloud storage
4. Code repos for recon
5. Passive recon using public datasets

CERTIFICATE TRANSPARENCY

- Under CT, a Certificate Authority(CA) will have to publish all SSL/TLS certificates they issue in a public log
- Anyone can look through the CT logs and find certificates issued for a domain
- Details of known CT log files -
<https://www.certificate-transparency.org/known-logs>

CT - SIDE EFFECT

- CT logs by design contain all the certificates issued by a participating CA for any given domain
- By looking through the logs, **an attacker can gather a lot of information** about an organization's infrastructure i.e. internal domains, email addresses in a **completely passive manner**

<https://blog.appsecco.com/certificate-transparency-part-3-the-dark-side-9d401809b025>

SEARCHING THROUGH CT LOGS

- There are various search engines that collect the CT logs and let's anyone search through them
 1. <https://crt.sh/>
 2. <https://censys.io/>
 3. <https://developers.facebook.com/tools/ct/>
 4. <https://google.com/transparencyreport/https/ct/>

Searching SSL/TLS certificates issued for a domain

crt.sh

Identity Search



[Group by Issuer](#)

Criteria

Identity LIKE %

crt.sh ID	Logged At ↑	Not Before	Identity	Issuer Name
	2017-07-01	2017-07-01	dana. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-07-01	2017-07-01	dana. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-07-01	2017-07-01	git. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-06-01	2017-06-01	dana. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-06-01	2017-06-01	dana. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-06-01	2017-06-01	git. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-05-12	2017-05-12	jenkins. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-05-01	2017-05-01	dana. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-05-01	2017-05-01	dana. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-05-01	2017-05-01	git. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-05-01	2017-05-01	www. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-04-01	2017-04-01	dana. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-04-01	2017-04-01	dana. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-04-01	2017-04-01	git. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
	2017-04-01	2017-04-01	www. .com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3

<https://crt.sh>

Output of a script that searches for sub-domains using crt.sh

```
└─>$ python3 ct.py teslamotors.com
shop.eu.teslamotors.com
shop.uk.teslamotors.com
streaming.vn.teslamotors.com
*.teslamotors.com
vpn.teslamotors.com
us.auth.teslamotors.com
shop.teslamotors.com
energystorage.teslamotors.com
sdlcvpn.teslamotors.com
*.vn.teslamotors.com
vn.teslamotors.com
feedback.teslamotors.com
owner-api.teslamotors.com
fleetview.teslamotors.com
sftp.teslamotors.com
cn.teslamotors.com
my.teslamotors.com
www.teslamotors.com
*.dev.teslamotors.com
dev.teslamotors.com
euvpn.teslamotors.com
smswsproxy.teslamotors.com
quickbase.teslamotors.com
trt.teslamotors.com
creditauction.teslamotors.com
```

<https://crt.sh>


Output of a script that searches for sub-domains using censys.io

```
└─>$ python subdomain_enum_censys.py wikimedia.org
[+] Extracting certificates for wikimedia.org using Censys
Starting new HTTPS connection (1): www.censys.io
[+] Extracting sub-domains for wikimedia.org from certificates
[+] Total unique subdomains found: 38
[+] List of subdomains extracted:

ssl.shopify.com
etherpad.wikimedia.org
rt.wikimedia.org
*.planet.wikimedia.org
blog.wikimedia.org
bug-attachment.wikimedia.org
mail.wikimedia.org
ticket.wikimedia.org
*.corp.wikimedia.org
ganglia.wikimedia.org
wikitech.wikimedia.org
*.wikimedia.org
```

<https://censys.io>

KEEPING TRACK OF AN ORGANISATION'S SUB-DOMAINS

facebook for developers  [Products](#) [Docs](#) [Tools & Support](#) [News](#) [Videos](#) [Get Started](#)

Certificate Transparency Monitoring

Certificate Transparency is an open framework to log, audit and monitor all publicly-trusted TLS certificates on the Internet. This tool lets you search for certificates issued for a given domain. Subscribe to email updates to be alerted when new certificates are issued.

Certificates **Subscriptions**

Domains	Notification Email	Remove Subscription
example.com	b[REDACTED]com	×

<https://developers.facebook.com/tools/ct/>

DOWNSIDE OF CT FOR RECON

- CT logs are append-only. There is no way to delete an existing entry
- The domain names found in the CT logs may not exist anymore and thus they can't be resolved to an IP address

<https://blog.appsecco.com/a-penetration-testers-guide-to-sub-domain-enumeration-7d842d5570f6>

CT LOGS + MASSDNS

- You can use tools like [massdns](#) along with CT logs script to quickly identify resolvable domain names.

```
python3 ct.py example.com | ./bin/massdns -r resolvers.txt -t A -a -o -w res
```

```
[~]: ~/tools/massdns
->$ ./ct.py icann.org | ./bin/massdns -r resolvers.txt -t A -q -a -o -w icann_resolvable_domains.txt -
[~]: ~/tools/massdns
->$ cat icann_resolvable_domains.txt
access-mgmt.dc.icann.org. 3405 IN A 10.47.60.10
mi-vsp.icann.org. 3405 IN CNAME mi-vsp.vip.icann.org.
mi-vsp.vip.icann.org. 20 IN A 192.0.32.205
uac-mdr.icann.org. 3406 IN CNAME uac1.mdr.icann.org.
uac1.mdr.icann.org. 3406 IN A 10.36.65.25
access-mgmt.dev.icann.org. 3406 IN A 10.47.60.5
www.aso.icann.org. 21406 IN CNAME aso.icann.org.
aso.icann.org. 106 IN A 193.0.6.147
owa.icann.org. 406 IN CNAME owa.vip.icann.org.
owa.vip.icann.org. 20 IN A 64.78.40.14
automated-ksk-test.research.icann.org. 406 IN A 192.0.34.57
aso.icann.org. 106 IN A 193.0.6.147
schedule.icann.org. 3406 IN CNAME domains.sched.org.
domains.sched.org. 106 IN A 45.56.77.32
dns.icann.org. 28606 IN A 192.0.43.22
features.icann.org. 3406 IN CNAME www.myicann.org.
www.myicann.org. 106 IN CNAME dualstack.myicann-production-155313818.us-east-1.elb.amazonaws.com.
dualstack.myicann-production-155313818.us-east-1.elb.amazonaws.com. 50 IN A 54.235.213.196
dualstack.myicann-production-155313818.us-east-1.elb.amazonaws.com. 50 IN A 54.225.144.95
```

ct.py extracts domains names from CT logs
massdns will find resolvable domains and writes it to a file

FINDING VULNERABLE CMS USING CT

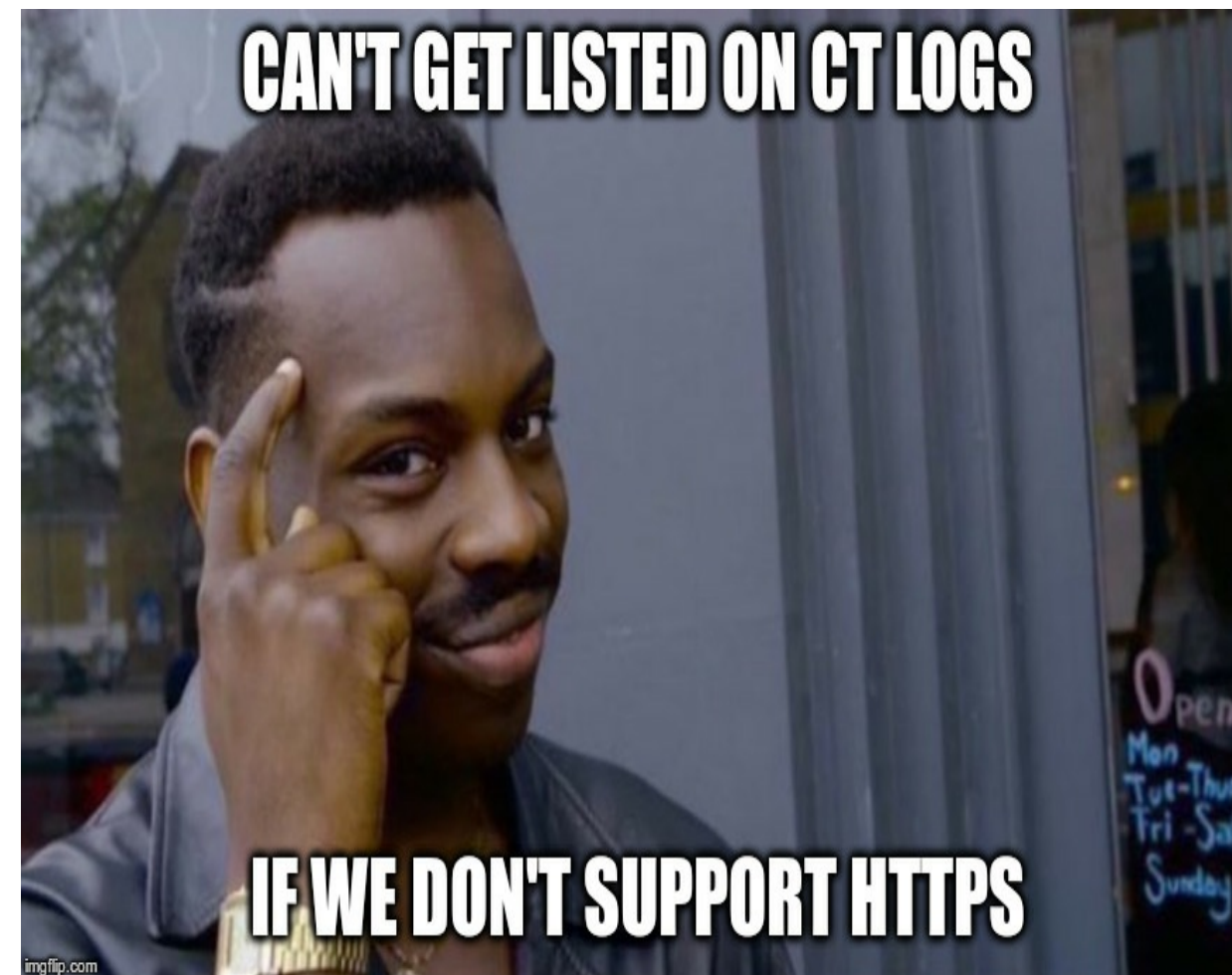
- When setting up some CMSs like Wordpress, Joomla and others, there is a window of time where the installer has no form of authentication
- If the domain supports HTTPS it will end up on a CT log(sometimes in near real time)
- If an attacker can search through CT Logs and find such a web application without authentication then he/she can take over the server

FINDING VULNERABLE CMS USING CT

- This attack has been demonstrated by [Hanno Böck at Defcon 25](#)
- He claimed to have found 5,000 WordPress installations using CT logs over a period of 3 months that he could have potentially taken over
- HD Moore also discussed this technique in his [talk at BSidesLV 2017](#)

CT LOGS - MITIGATION

- Not have SSL/TLS support. This approach is definitely not recommended



CT LOGS - MITIGATION

- Using wildcard certificates will avoid sub-domain names being listed in CT Logs but **wildcard certs are a security risk**

CT LOGS - MITIGATION

- Deploy your own Public Key Infrastructure(PKI)
- [CFSSL](#) project by CloudFlare helps you build an internal PKI.
- [Certmgr](#) by Cloudflare automates certificate management using a CFSSL.
- Opt out of CT logs but you'll miss out on all the security benefits that CT provides
- Name redaction in CT logs let's you hide your sub-domain information in a CT log

DNSSEC

- DNSSEC provides a layer of security by adding cryptographic signatures to existing DNS records
- These signatures are stored alongside common record types like A, AAAA, MX etc

DNSSEC - NEW RECORDS

Record	Purpose
RRSIG	Contains a cryptographic signature.
NSEC and NSEC3	For explicit denial-of-existence of a DNS record
DNSKEY	Contains a public signing key
DS	Contains the hash of a DNSKEY record

DNSSEC - AUTHENTICATED DENIAL OF EXISTENCE(RFC 7129)

In DNS, when client queries for a non-existent domain, the server must deny the existence of that domain. It is harder to do that in DNSSEC due to cryptographic signing.

PROBLEMS WITH AUTHENTICATED DENIAL OF EXISTENCE(DNSSEC)

1. *NXDOMAIN* responses are generic, attackers can spoof the responses
2. Signing the responses on the fly would mean a performance and security problem
3. Pre-signing every possible *NXDOMAIN* record is not possible as there will be infinite possibilities

NSEC

- Zone entries are sorted alphabetically, and the NextSECure(NSEC) records point to the record after the one you looked up
- Basically, **NSEC record says, “there are no subdomains between sub-domain X and sub-domain Y.”**

```
$ dig +dnssec @ns1.insecuredns.com firewall.insecuredns.com
... snipped ...
firewall.insecuredns.com. 604800 IN NSEC mail.insecuredns.com. A RRSIG
... snipped ...
```

ZONE WALKING NSEC - LDNS

- The `ldns-walk`(part of `ldnsutils`) can be used to zone walk DNSSEC signed zone that uses NSEC.

```
# zone walking with ldnsutils
$ ldns-walk iana.org
iana.org. iana.org. A NS SOA MX TXT AAAA RRSIG NSEC DNSKEY
api.iana.org. CNAME RRSIG NSEC
app.iana.org. CNAME RRSIG NSEC
autodiscover.iana.org. CNAME RRSIG NSEC
beta.iana.org. CNAME RRSIG NSEC
data.iana.org. CNAME RRSIG NSEC
dev.iana.org. CNAME RRSIG NSEC
ftp.iana.org. CNAME RRSIG NSEC
^C
```

ZONE WALKING NSEC - LDNS

```
[redacted]: ~/appsecco/conferences/levelup/artifacts]
->$ ldns-walk @ns1.insecuredns.com insecuredns.com
insecuredns.com.      insecuredns.com. A NS SOA TXT RRSIG NSEC DNSKEY
champ.insecuredns.com. A RRSIG NSEC
conference.insecuredns.com. A RRSIG NSEC
damn.insecuredns.com. A RRSIG NSEC
firewall.insecuredns.com. A RRSIG NSEC
mail.insecuredns.com. A RRSIG NSEC
ns1.insecuredns.com. A RRSIG NSEC
ns2.insecuredns.com. A RRSIG NSEC
null.insecuredns.com. A RRSIG NSEC
secrets.insecuredns.com. A RRSIG NSEC
staging.insecuredns.com. A RRSIG NSEC
vpn.insecuredns.com. A RRSIG NSEC
www.insecuredns.com. A RRSIG NSEC
```

INSTALLING LDNSUTILS

```
# On Debian/Ubuntu
```

```
$ sudo apt-get install ldnsutils
```

```
# On Redhat/CentOS
```

```
$ sudo yum install ldns
```

```
# You may need to do
```

```
$ sudo yum install -y epel-release
```

ZONE WALKING NSEC - DIG

- You can list all the sub-domains by following the linked list of NSEC records of existing domains.

```
$ dig +short NSEC api.nasa.gov  
apm.nasa.gov. CNAME RRSIG NSEC
```

```
$ dig +short NSEC apm.nasa.gov  
apmcpr.nasa.gov. A RRSIG NSEC
```

EXTRACTING THE SUB-DOMAIN FROM NSEC

- You can extract the specific sub-domain part using `awk` utility.

```
$ dig +short NSEC api.nasa.gov | awk '{print $1;}'  
apm.nasa.gov.
```

NSEC3

- The NSEC3 record is like an NSEC record, but, NSEC3 provides a signed gap of **hashes of domain names**.
- Returning hashes was intended to prevent zone enumeration(or make it expensive).

```
231SPNAMH63428R68U7BV359PFPJI2FC.example.com. NSEC3 1 0 3 ABCD  
NKDO8UKT2STOL6EJRD1EKVD1BQ2688DM A NS SOA TXT AAAA RRSIG DN
```

```
NKDO8UKT2STOL6EJRD1EKVD1BQ2688DM.example.com. NSEC3 1 0 3 AB  
231SPNAMH63428R68U7BV359PFPJI2FC A TXT AAAA RRSIG
```

NSEC3 - LINKED LIST OF HASHES

Salted hash of
example.com

231SPNAMH63428R68U7BV359PFPJI2FC.example.com. NSEC3 1 0 3 ^{Salt}ABCDEF
NKDO8UKT2STOL6EJRD1EKVD1BQ2688DM A NS SOA TXT AAAA RRSIG DNSKEY NSEC3PARAM

Salted hash of
www.example.com

Salted hash of
www.example.com

NKDO8UKT2STOL6EJRD1EKVD1BQ2688DM.example.com. NSEC3 1 0 3 ^{Salt}ABCDEF
231SPNAMH63428R68U7BV359PFPJI2FC A TXT AAAA RRSIG

Salted hash of
example.com

GENERATING NSEC3 HASH FOR A DOMAIN NAME

- `ldns-nsec3-hash`(part of `ldnsutils`) generates NSEC3 hash of domain name for a given salt value and number of iterations
- Number of iterations & salt value is available as part of NSEC3 record.

```
$ ldns-nsec3-hash -t 3 -s ABCDEF example.com  
231spnamh63428r68u7bv359pfpji2fc.
```

```
$ ldns-nsec3-hash -t 3 -s ABCDEF www.example.com  
nkdo8ukt2stol6ejrd1ekvd1bq2688dm.
```

ZONE WALKING NSEC3

- An attacker can collect all the sub-domain hashes and crack the hashes offline
- Tools like *nsec3walker*, *nsec3map* help us automate collecting NSEC3 hashes and cracking the hashes

ZONE WALKING NSEC3

Zone walking NSEC3 protected zone using
nsec3walker:

```
# Collect NSEC3 hashes of a domain  
$ ./collect insecuredns.com > insecuredns.com.collect
```

```
# Undo the hashing, expose the sub-domain information.  
$ ./unhash < insecuredns.com.collect > insecuredns.com.unhash
```

ZONE WALKING NSEC3

```
# Checking the number of successfully cracked sub-domain hashes  
$ cat icann.org.unhash | grep "icann" | wc -l  
182
```

```
# Listing only the sub-domain part from the unhashed data  
$ cat icann.org.unhash | grep "icann" | awk '{print $2;}'  
del.icann.org.  
access.icann.org.  
charts.icann.org.  
communications.icann.org.  
fellowship.icann.org.  
files.icann.org.  
forms.icann.org.  
mail.icann.org.  
maintenance.icann.org.  
new.icann.org.  
public.icann.org.  
research.icann.org.  
rs.icann.org.
```

INSTALLING NSEC3WALKER

- Installation instructions are available at <https://dnscurve.org/nsec3walker.html>
- I used following commands to install nsec3walker on Ubuntu 16.04.
 - build-essential package is a prerequisite.

```
# Installing nsec3walker
$ wget https://dnscurve.org/nsec3walker-20101223.tar.gz
$ tar -xzf nsec3walker-20101223.tar.gz
$ cd nsec3walker-20101223
$ make
```

FEW THINGS THAT CHANGED WITH THE ADVENT OF APIS/DEVOPS

1. Storage
2. Authentication
3. More and more code
4. CI/CD pipelines

CLOUD STORAGE

- Cloud storage has gotten inexpensive, easy to setup and gained popularity
- Especially object/block storage
- Object storage is ideal for storing static, unstructured data like audio, video, documents, images and logs as well as large amounts of text.
 1. AWS S3 buckets
 2. Digital Ocean Spaces

WHAT'S THE CATCH WITH OBJECT STORAGE?

- Due to the nature of object storage, it is a treasure trove of information from an attacker/penetration tester perspective.
- In our experience, given an chance, users will store anything on third-party services, from their passwords in plain text files to pictures of their pets.

AMAZON S3 BUCKETS

- AWS S3 is an object storage service by Amazon
- Buckets allow users to store and serve large amounts of data.


Attack on Accenture(Sep, 2017)- AWS S3 buckets as attack surface

A potentially devastating Amazon S3 bucket exposure left internal Accenture private keys, secret API data and other information publicly available to anyone who could then leverage it to attack the global consulting firm and its clients.


<https://www.upguard.com/breaches/cloud-leak-accenture>

AWS S3 buckets as attack surface - The trend

Javvad Malik, security advocate at AlienVault, added: "Massive breaches through unsecured AWS S3 buckets continue to be a troubling trend.

 While cloud providers take care of certain aspects of security, it is

AWS S3 buckets as attack surface - The trend



AWS S3 leaks, due to customer configuration blunders, are becoming the flavour of 2017. Verizon [leaked](#) 14 million customer records, and other open buckets researchers have spotted include those belonging to [Dow Jones](#), voting machine supplier [ES&S](#) (both found by former MacKeeper security bod Chris Vickery).

HUNTING FOR PUBLICLY ACCESSIBLE S3 BUCKETS

- Users can store Files(Objects) in a Bucket
- Each Bucket will get an unique, predictable URL and each file in a Bucket will get an unique URL as well
- There are Access controls mechanisms available at both Bucket and Object level.

HUNTING FOR PUBLICLY ACCESSIBLE S3 BUCKETS

- Good old Google dorks

```
site:s3.amazonaws.com file:pdf
```

```
site:s3.amazonaws.com password
```

HUNTING FOR PUBLICLY ACCESSIBLE S3 BUCKETS

- As buckets have predictable URL it is trivial to do a dictionary based attack
- Following tools help run a dictionary attack to identify S3 buckets
 1. [AWSBucketDump](#)
 2. [Bucket finder](#)

DIGITAL OCEAN SPACES

- Spaces is an object storage service by DigitalOcean
- It is similar to AWS S3 buckets
- Spaces API aims to be interoperable with Amazon's AWS S3 API.

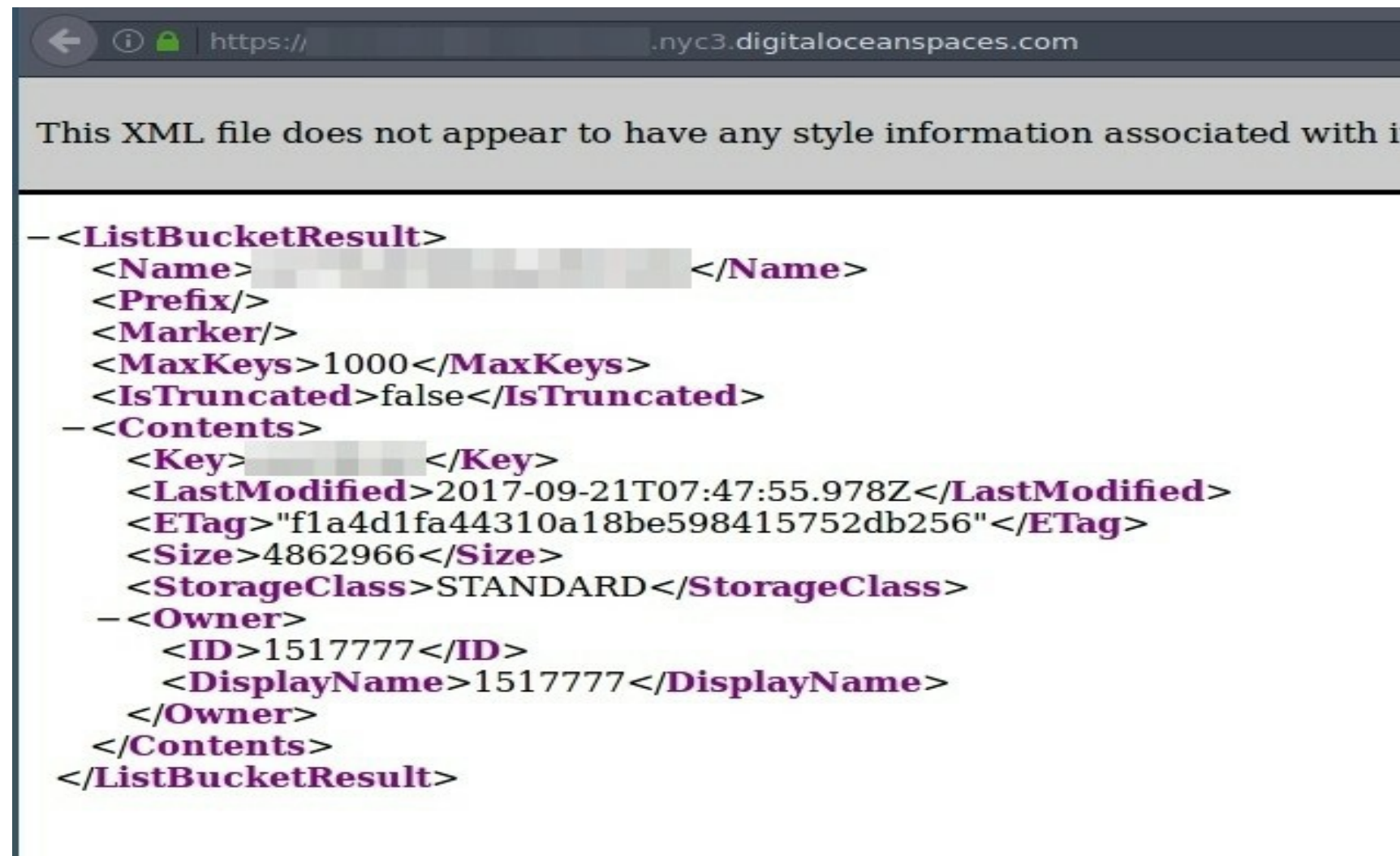
SPACES URL PATTERN

- Users can store Files in a “Space”
- Each Space will get an unique, predictable URL
- Each file in a Space will get an unique URL as well.
- Access controls mechanisms are available at Space and file level.

- **Regional Availability:** At launch, Spaces are available in the NYC3 region.
- **Supported Protocols:** HTTPS.
- **URL Naming Pattern:** `spacename.region.digitaloceanspaces.com` or `region.digitaloceanspaces.com/spacename`

HUNTING FOR PUBLICLY ACCESSIBLE S3 BUCKETS

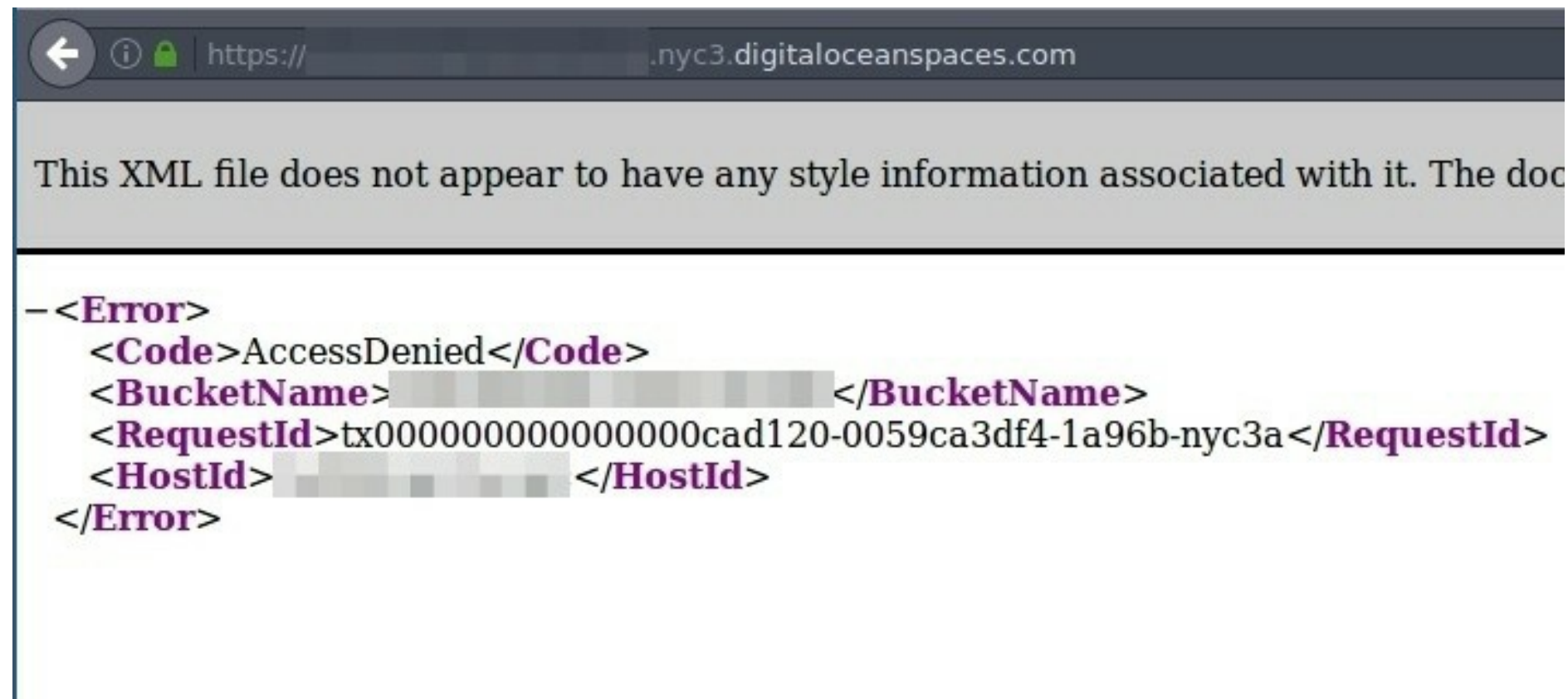
A Space is typically considered “public” if any user can list the contents of the Space



The screenshot shows a web browser window with the address bar displaying `https://.nyc3.digitaloceanspaces.com`. Below the address bar, a message states: "This XML file does not appear to have any style information associated with it". The main content area displays an XML document representing a `ListBucketResult`. The XML is color-coded with purple for tags and black for text values. Some values are redacted with grey boxes.

```
-<ListBucketResult>
  <Name>[REDACTED]</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>>false</IsTruncated>
  -<Contents>
    <Key>[REDACTED]</Key>
    <LastModified>2017-09-21T07:47:55.978Z</LastModified>
    <ETag>"f1a4d1fa44310a18be598415752db256"</ETag>
    <Size>4862966</Size>
    <StorageClass>STANDARD</StorageClass>
  -<Owner>
    <ID>1517777</ID>
    <DisplayName>1517777</DisplayName>
  </Owner>
</Contents>
</ListBucketResult>
```

A Space is typically considered “private” if the Space’s contents can only be listed or written by certain users



The screenshot shows a web browser window with the address bar displaying `https://[redacted].nyc3.digitaloceanspaces.com`. Below the address bar, a message states: "This XML file does not appear to have any style information associated with it. The document root element, 'Error', is not recognized." Below this message, the XML content is displayed:

```
-<Error>
  <Code>AccessDenied</Code>
  <BucketName>[redacted]</BucketName>
  <RequestId>tx0000000000000000cad120-0059ca3df4-1a96b-nyc3a</RequestId>
  <HostId>[redacted]</HostId>
</Error>
```

SPACES FINDER

- Spaces API is interoperable with Amazon's S3 API, we tweaked [AWSBucketDump](#) to work with DO Spaces
- *Spaces finder* is a tool that can look for publicly accessible DO Spaces using a wordlist, list all the accessible files on a public Space and download the files.

<https://github.com/appsecco/spaces-finder>

SPACES FINDER IN ACTION

```
└─>$ python3 spaces_finder.py -l sample_spaces.txt -g interesting_keywords.txt -D -m 500000 -t 2
[*] Downloads enabled (-D), and will be saved to current directory
[+] starting thread
[+] starting thread
[+] download worker running
[+] queuing https://[REDACTED].nyc3.digitaloceanspaces.com
[+] queuing https://[REDACTED]paces.com
[+] fetching https://[REDACTED].nyc3.digitaloceanspaces.com
[+] fetching https://[REDACTED]paces.com
[+] Pilfering https://[REDACTED]c3.digitaloceanspaces.com
[*] Collectable: https://[REDACTED].nyc3.digitaloceanspaces.com
[+] Downloading https://[REDACTED].nyc3.digitaloceanspaces.com/
[*] local [REDACTED].nyc3.digitaloceanspaces.com/
[*] Collectable: https://[REDACTED].nyc3.digitaloceanspaces.com
[*] Collectable: https://[REDACTED].nyc3.digitaloceanspaces.com
[*] Collectable: https://[REDACTED].nyc3.digitaloceanspaces.com
[*] Collectable: https://[REDACTED].nyc3.digitaloceanspaces.com
[*] Collectable: https://[REDACTED].nyc3.digitaloceanspaces.com
[*] Collectable: https://[REDACTED].nyc3.digitaloceanspaces.com
[*] Collectable: https://[REDACTED].nyc3.digitaloceanspaces.com
[*] Collectable: https://[REDACTED].nyc3.digitaloceanspaces.com
[*] Collectable: https://[REDACTED].nyc3.digitaloceanspaces.com
[+] Pilfering https://[REDACTED]loceanspaces.com
```



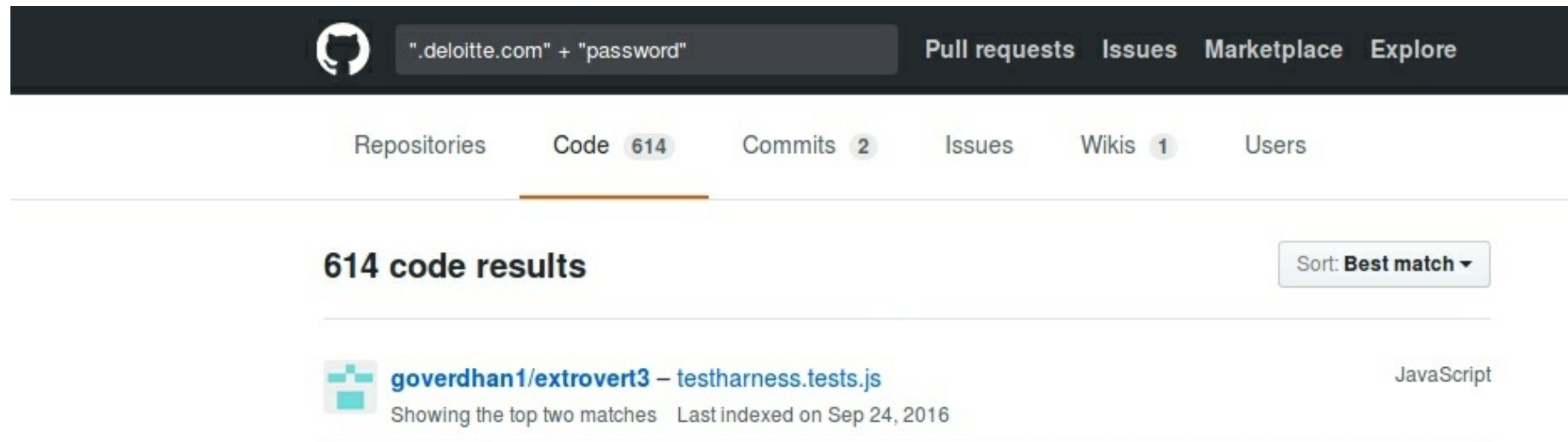
AUTHENTICATION

- With almost every service exposing an API, keys have become critical in authenticating
- API keys are treated as keys to the kingdom
- For applications, API keys tend to be achilles heel

<https://danielmiessler.com/blog/apis-2fas-achilles-heel/>

CODE REPOS FOR RECON

- Code repos are a treasure trove during recon
- Code repos can reveal a lot from credentials, potential vulnerabilities to infrastructure details



The screenshot shows the GitHub search interface. At the top, the search bar contains the query `".deloitte.com" + "password"`. Navigation links include "Pull requests", "Issues", "Marketplace", and "Explore". Below the search bar, a horizontal bar displays search statistics: "Repositories", "Code 614", "Commits 2", "Issues", "Wikis 1", and "Users". The "Code" section is highlighted with an orange underline. Below this, the text "614 code results" is displayed on the left, and a "Sort: Best match" dropdown menu is on the right. The first search result is for the repository `goverdhan1/extrovert3` with the file `testharness.tests.js`, identified as a JavaScript file. Below the repository name, it says "Showing the top two matches" and "Last indexed on Sep 24, 2016".

GITHUB FOR RECON


- GitHub is an extremely popular version control and collaboration platform
- Code repos on github tend to have all sorts of sensitive information
- Github also has a powerful search feature with advanced operators
- Github has a very well designed REST API
- [edoverflow](#) has a neat little guide on [GitHub for Bug Bounty Hunters](#)

THINGS TO FOCUS ON IN GITHUB

There are 4 main sections to look out for here.

- Repositories
- Code
- Commits(My fav!)
- Issues


delete the private ssh 

 master

 committed on Sep 26

 Showing **1 changed file** with **0 additions** and **30 deletions**.

30  

... @@ -1,30 +0,0 @@ 

```
1 -----BEGIN RSA PRIVATE KEY-----
2 -Proc-Type: 4,ENCRYPTED
3 -DEK-Info: AES-128-CBC,6FC5A5E8A9D000A76763C69F363F493B
4 -
5 -0tYn1QBU94+y3C0+rjTctqi03xrX3hWC501wiN0Waj7rYKWPSoVDV621501ILp0P
6 -0HZIhrhz0C6uqx10DiCg+MaVylyR16eDWdzKrRLNkX2ov0ZdGC14Cf105L2iB048
7 -/5hmrkdFFlwu/0vNs5Ku2zkrKfoM4KHnRVJaz7+wQJNWYKTb1RIF0NbgmncWc7gm
```

GitHub, Inc. (US) <https://github.com/search?p=4&q=sqli+vulnerable&type=Issues&utf8=✓>

reflected XSS via basqli parameter - reflected XSS via **sqli** parameter - reflected XSS via
urlToScan parameter What is the ...

3 comments



SQL injection vuln in data.views



Multiple XSS vulnerabilities

... **vulnerabilities** for scanner.php resource - reflected XSS via autoc parameter -
reflected XSS via basqli parameter - reflected XSS via **sqli** parameter - reflected XSS via
urlToScan parameter What is the ...

3 comments

MASS CLONING ON GITHUB

- You can ideally clone all the target organization's repos and analyze them locally
- [GitHubCloner](#) by @mazen160 comes very handy to automate the process

```
$ python githubcloner.py --org organization -o /tmp/output
```

<https://gist.github.com/EdOverflow/922549f610b258f459b219a32f92d10b>

STATIC CODE ANALYSIS

- Once the repos are cloned, you can do a static code analysis
- There are language specific tools to speed up and automate the process
 1. [Brakeman](#) for Ruby
 2. [Bandit](#) for Python

MANUAL SEARCH

- Once you have the repos cloned. You can understand the code, language used and architecture
- Start looking for keywords or patterns

- API **and** key. (Get some more endpoints **and** find API keys.)
- token
- secret
- vulnerable
- **http://**

GITHUB DORKS

- Github dorks are the new Google dorks
- Github search is quite powerful feature & can be used to find sensitive data on the repos
- **A collection of Github dorks**
<https://github.com/techgaun/github-dorks/blob/master/github-dorks.txt>
- **Tool to run Github dorks against a repo**
<https://github.com/techgaun/github-dorks>

PASSIVE RECON USING PUBLIC DATASETS

- There are various projects that gather Internet wide scan data and make it available to researchers and the security community.
- This data includes port scans, DNS data, SSL/TLS cert data and even data breach dumps that they can find.
- Find your needle in the haystack.

WHY USE PUBLIC DATA SETS FOR RECON?

- To reduce dependency on 3rd party APIs and services
- To reduce active probing of target infrastructure
- More the sources better the coverage
- Build your own recon platforms

LET'S LOOK AT SOME PUBLIC DATASETS

Name	Description	Price
Sonar	FDNS, RDNS, UDP, TCP, TLS, HTTP, HTTPS scan data	FREE
Censys.io	TCP, TLS, HTTP, HTTPS scan data	FREE
CT	TLS	FREE

<https://github.com/fathom6/inetdata>

LET'S LOOK AT SOME PUBLIC DATASETS

Name	Description	Price
CZDS	zone files for "new" global TLDs	FREE
ARIN	American IP registry information	FREE
CAIDA PFX2AS IPv4	Daily snapshots of ASN to IPv4 mappings	FREE

LET'S LOOK AT SOME PUBLIC DATASETS

Name	Description	Price
US Gov	US government domain names	FREE
UK Gov	UK government domain names	FREE
RIR Delegations	Regional IP allocations	FREE

LET'S LOOK AT SOME PUBLIC DATASETS

Name	Description	Price
PremiumDrops	DNS zone files for com/net/info/org/biz/xxx/sk/us TLDs	\$24.95/mo
WWWS.io	Domains across many TLDs (~198m)	\$9/mo
WhoisXMLAPI.com	New domain whois data	\$109/mo

<https://github.com/fathom6/inetdata>

RAPID7 FORWARD DNS DATASET

- Rapid7 publishes its Forward DNS study/dataset on scans.io project(it's a massive dataset, 20+ GB compressed & 300+ GB uncompressed)
- This dataset aims to discover all domains found on the Internet

HUNTING SUB-DOMAIN IN FDNS DATASET

- The data format is a gzip-compressed JSON file so we can use jq utility to extract sub-domains of a specific domain:

```
curl --silent https://scans.io/data/rapid7/sonar.fdns_v2/20170417-fdns.json
```

```
cat 20170417-fdns.json.gz | pigz -dc | grep "\.example\.com" | jq .name
```

<https://sonar.labs.rapid7.com/>

HUNTING SUB-DOMAIN IN FDNS DATASET

```
~/data$ cat 20170728-fdns.json.gz | pigz -dc | grep "\.com" | jq .name > .com.domains.fdns
~/data$ cat .com.domains.fdns | grep "\.com" | uniq | head -n 15
"a.dev.com"
"aandrade.dev.com"
"abq.dev.com"
"accessibility.com"
"achal0.dev.com"
"acura-astra-4.dev.com"
"adamp.dev.com"
"aditi.dev.com"
"aditya.dev.com"
"admin.com"
"adw-golden-apr1.dev.com"
"adw-golden-p2.dev.com"
"agate.dev.com"
"ajb.dev.com"
"ajj.dev.com"
~/data$ cat .com.domains.fdns | grep "\.com" | uniq | wc -l
865
```

② ↓
Display first 15 sub-domains
from all the unique sub-domains gathered

① ↓
Extract sub-domain names
for a given domain from FDNS data

③ ↑
Total number of
unique sub-domains enumerated

Subdomain enumeration cheat sheet

Certificate Transparency logs - search engines

<https://crt.sh/>

<https://censys.io/>

<https://google.com/transparencyreport/https/ct/>

Extracting sub-domains from Rapid7 FDNS dataset

```
$ zcat <dataset_name> | jq -r 'if (.name | test("\\.example\\.com$")) then .name else empty end'
```

```
$ zcat 20170204-fdns.json.gz | jq -r 'if (.name | test("\\.example\\.com$")) then .name else empty end'
```

Rapid7 · Forward DNS dataset

https://scans.io/study/sonar.fdns_v2



Zone walking - NSEC

```
$ ldns-walk @<nameserver> <domain>
```

```
$ ldns-walk @ns1.insecuredns.com  
insecuredns.com
```

Installing ldns utilities

```
$ sudo apt-get install ldnsutils #
```

On Ubuntu/Debian

```
$ yum install ldns # On
```

Redhat/CentOS

Zone transfer

```
$ dig AXFR @<nameserver> <domain>
```

```
$ dig AXFR @ns1.insecuredns.com  
insecuredns.com
```

Zone walking - NSEC3 - nsec3walker

```
$ ./collect insecuredns.com >  
insecuredns.com.collect
```

```
$ ./unhash <
```

```
insecuredns.com.collect >  
insecuredns.com.unhash
```

Installing nsec3walker on Ubuntu 16.04:

```
$ wget
```

<https://dnscurve.org/nsec3walker-20101223.tar.gz>

```
$ tar -xzf
```

```
nsec3walker-20101223.tar.gz
```

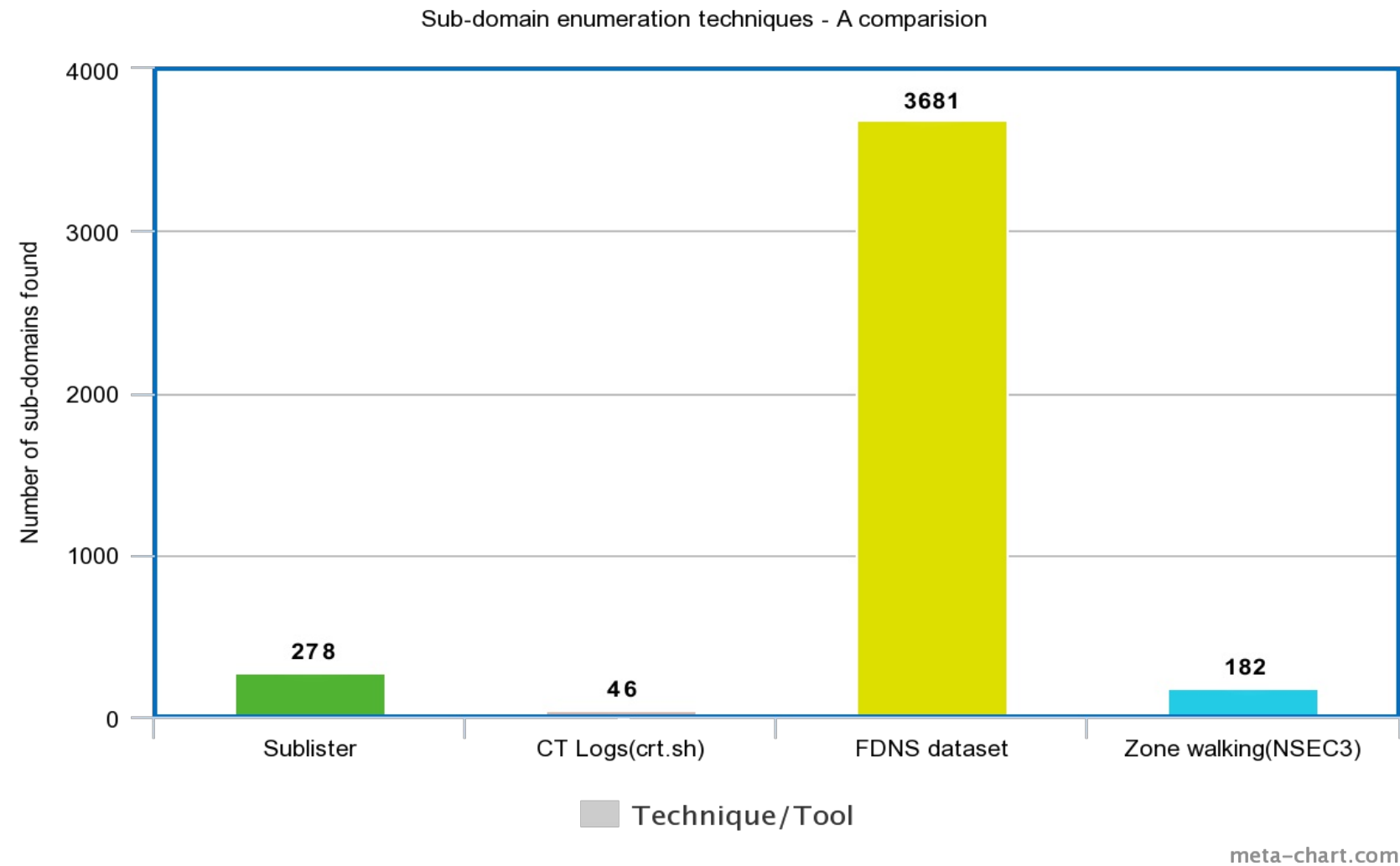
```
$ cd nsec3walker-20101223
```

```
$ make
```

Bharath
@yamakira_

ICANN.ORG SUBDOMAINS

Number of **unique, resolvable sub-domains** each enumeration technique found independently against icann.org



REFERENCES

- <https://www.certificate-transparency.org/>
- <https://www.cloudflare.com/dns/dnssec/how-dnssec-works/>
- <https://www.cloudflare.com/dns/dnssec/dnssec-complexities-and-considerations/>
- <http://info.menandmice.com/blog/bid/73645/Take-your-DNSSEC-with-a-grain-of-salt>
- <https://github.com/rapid7/sonar/wiki/Forward-DNS>

About Appsecco

- Pragmatic, holistic, business-focused approach
- Specialist Application Security company
- Highly experienced and diverse team
 - Commercial
 - Security; Gold Standards



OWASP chapter
leads



Certified
Hackers



Assigned
multiple CVEs



Def Con
speakers

THANKS

@yamakira_