Tree: 200c43b58e practical-recon-levelup0x02 / practical_recon.md Find file Copy path

0xbharath Updated twitter handle 200c43b on Jan 24

1 contributor

782 lines (463 sloc) 21.7 KB Raw Blame History

Practical recon techniques for bug hunters & pen testers

![](imgs/appsecco_logo.png) #### Bharath Kumar

About me

- Bharath Kumar
- Live from Bangalore, India
- Security Engineer @Appsecco
- **O**ffensive **S**ecurity **C**ertified **P**rofessional(OSCP)

Demo environment

- Feel free to run the DNS & DNSSEC attacks **mentioned in this talk** against the following nameservers and domain names

Nameservers

- **ns1.insecuredns.com**
- **ns2.insecuredns.com**

Domains - **totallylegit.in** - **insecuredns.com**

What is this talk about?

- This talk is about practical recon techniques that are useful for bug bounty hunters and penetration testers
- The objective of this talk is to cover exhaustive number of practical recon techniques, tools of trade and tips/tricks

Note: By practical I mean that these techniques covered can actually be used during a security assessment. The talk will be crisp and concise. We demonstrate quick and effective ways to apply a technique in such a way that the audience can use them in their assessments right away

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.

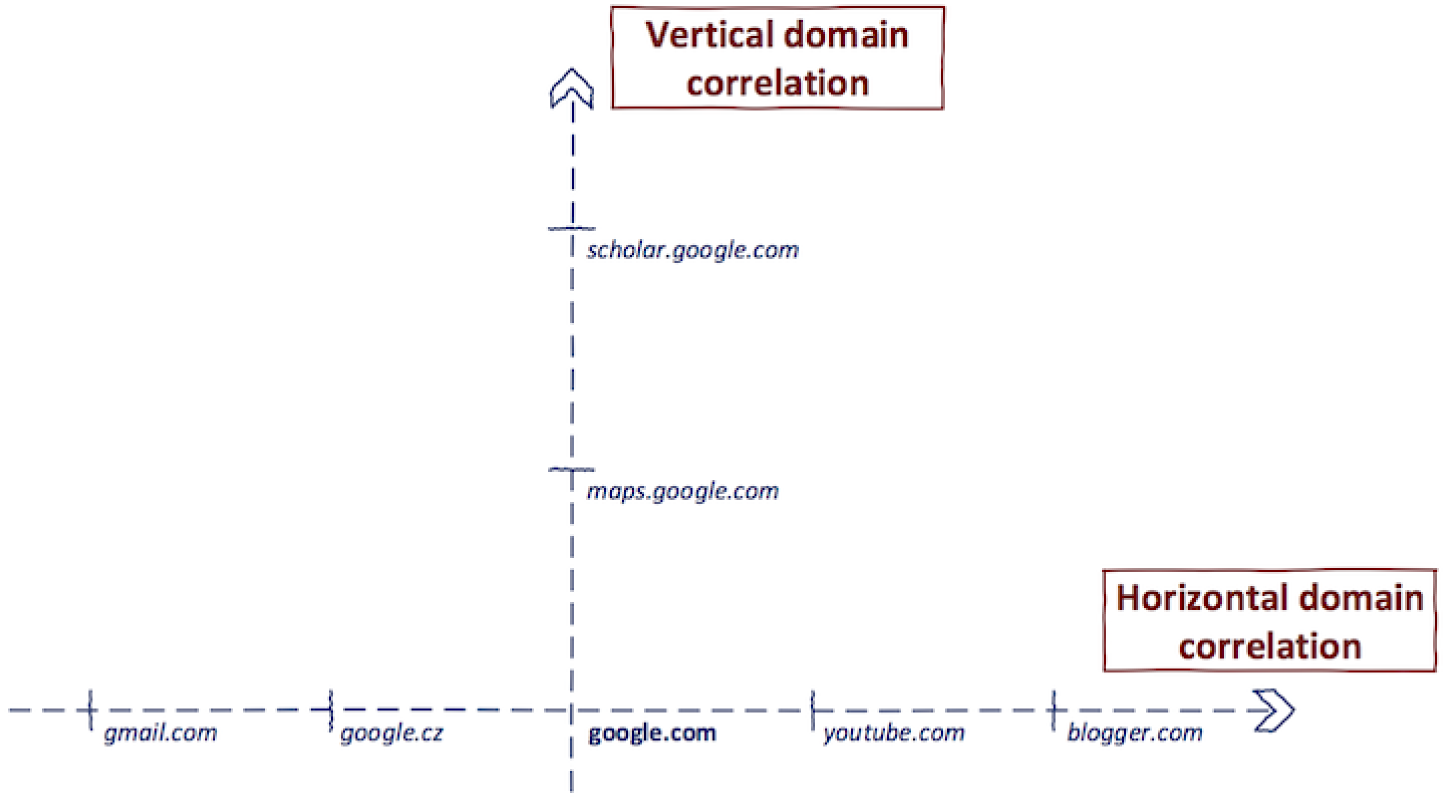
What do we look for during recon?

1. Info to increase attack surface(domains, net blocks)
2. Credentials(email, passwords, API keys)
3. Sensitive information
4. Infrastructure details

Enumerating domains

> The objective is to find/correlate all domain names owned by a single entity of our interest.

Types of domain correlation



https://0xpatrik.com/asset-discovery/

What is sub-domain enumeration?

> Sub-domain enumeration is the process of finding subdomains for one or more domain(s).

Using popular search engines

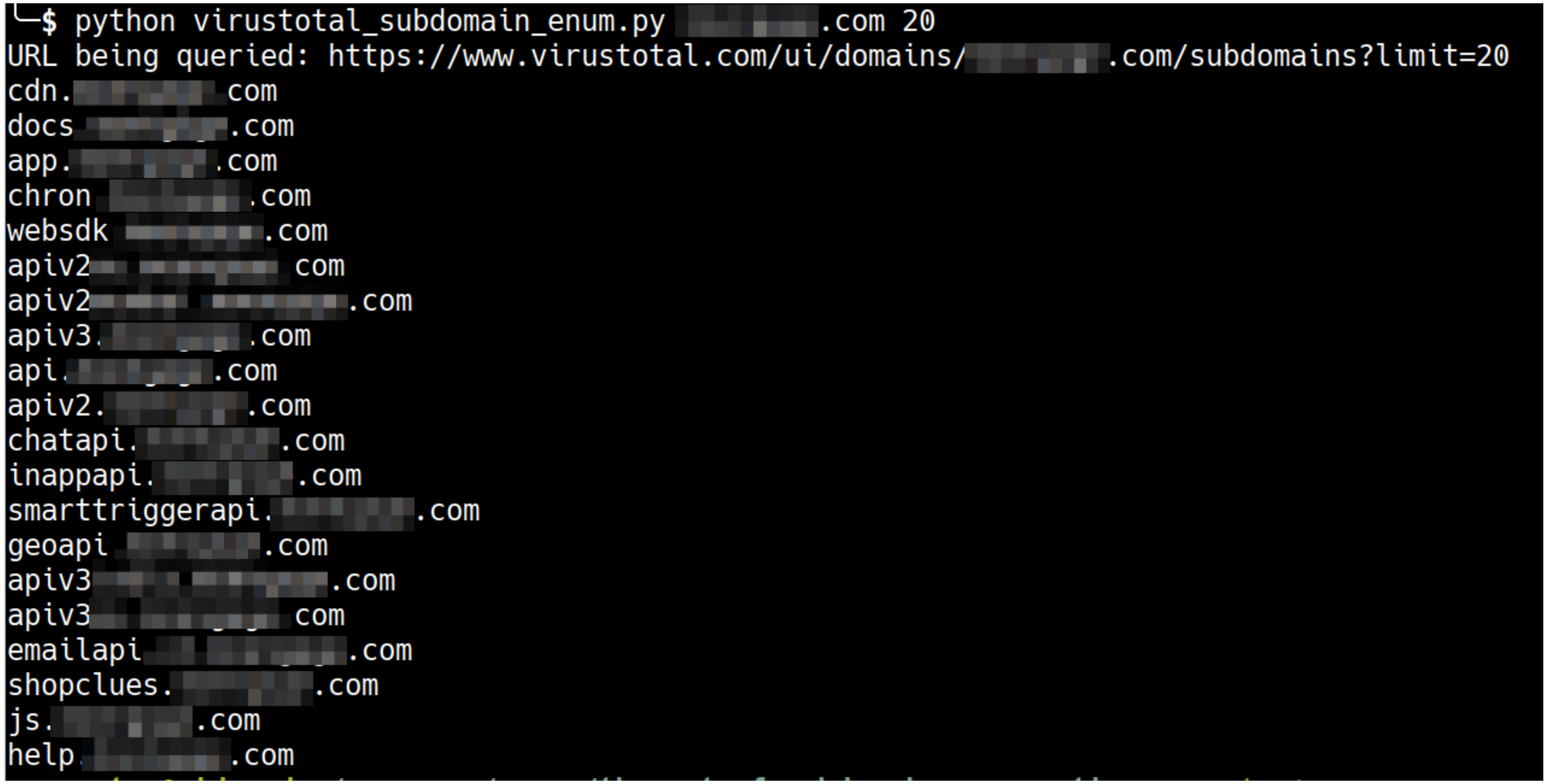
- Search engines like Google and Bing supports various advanced search operators to refine search queries.
- `site:` is helpful in doing vertical domain correlation(sub-domains)
- `ip:` is helpful in doing horizontal domain correlation

Using 3rd party information aggregators

- **VirusTotal** runs its own passive DNS replication service, built by storing DNS resolutions performed when visiting URLs submitted by users.

https://www.virustotal.com/#/home/search

A script that uses VirusTotal to find sub-domains



```
└─$ python virustotal_subdomain_enum.py [redacted].com 20
URL being queried: https://www.virustotal.com/ui/domains/[redacted].com/subdomains?limit=20
cdn.[redacted]com
docs[redacted].com
app.[redacted].com
chron[redacted].com
websdk[redacted].com
apiv2[redacted] com
apiv2[redacted].com
apiv3.[redacted].com
api.[redacted].com
apiv2.[redacted].com
chatapi.[redacted].com
inappapi.[redacted].com
smarttriggerapi.[redacted].com
geoapi[redacted].com
apiv3[redacted].com
apiv3[redacted] com
emailapi[redacted].com
shopclues.[redacted].com
js.[redacted].com
help[redacted].com
```

Script - https://git.io/vhqBF

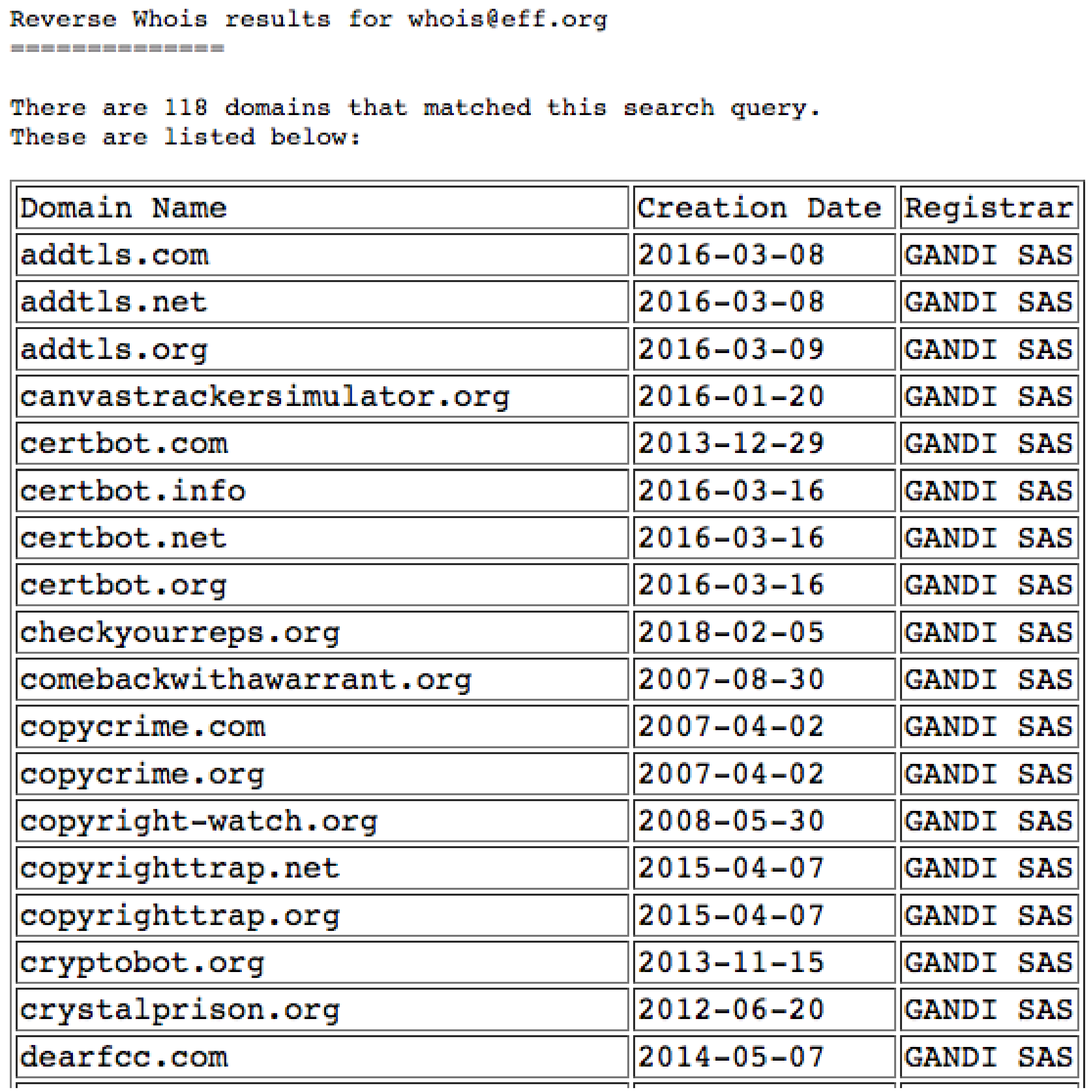
Quick tip

- I like using shell functions to quickly perform some recon tasks

```
find-subdomains-vt()
{ curl -s https://www.virustotal.com/ui/domains/$1/subdomains\?limit\=$2 | jq .data[].id; }
```

Using 3rd party information aggregators

- viewdns.info is a handy service for all the DNS and WHOIS related recon



```
Reverse Whois results for whois@eff.org
==============

There are 118 domains that matched this search query.
These are listed below:
```

Domain Name	Creation Date	Registrar
addtls.com	2016-03-08	GANDI SAS
addtls.net	2016-03-08	GANDI SAS
addtls.org	2016-03-09	GANDI SAS
canvastrackersimulator.org	2016-01-20	GANDI SAS
certbot.com	2013-12-29	GANDI SAS
certbot.info	2016-03-16	GANDI SAS
certbot.net	2016-03-16	GANDI SAS
certbot.org	2016-03-16	GANDI SAS
checkyourreps.org	2018-02-05	GANDI SAS
comebackwithawarrant.org	2007-08-30	GANDI SAS
copycrime.com	2007-04-02	GANDI SAS
copycrime.org	2007-04-02	GANDI SAS
copyright-watch.org	2008-05-30	GANDI SAS
copyrighttrap.net	2015-04-07	GANDI SAS
copyrighttrap.org	2015-04-07	GANDI SAS
cryptobot.org	2013-11-15	GANDI SAS
crystalprison.org	2012-06-20	GANDI SAS
dearfcc.com	2014-05-07	GANDI SAS

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

https://blog.appsecco.com/certificate-transparency-part-2-the-bright-side-c0b99ebf31a8

Certificate Transparency - side effect

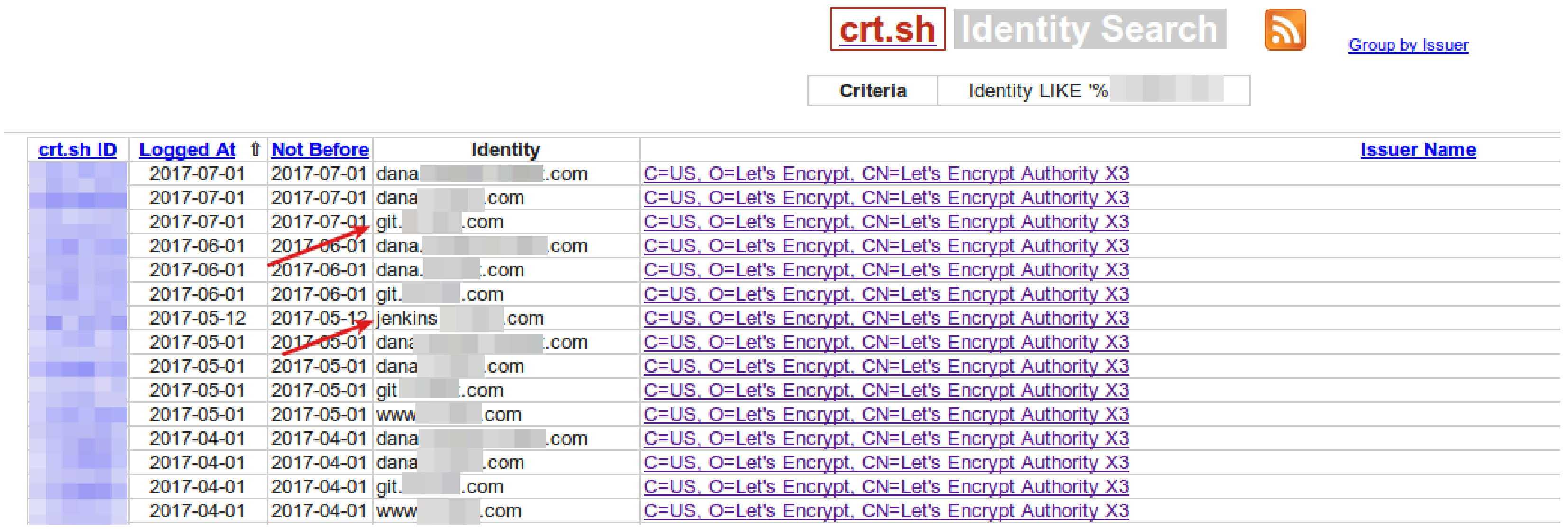
- CT logs by design contain all the certificates issued by a participating CA for any given domain
- By looking through the logs, **an attacker can gather a lot of information** about an organization's infrastructure i.e. internal domains, email addresses in a **completely passive manner**

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
    i. https://crt.sh/
    ii. https://censys.io/
    iii. https://developers.facebook.com/tools/ct/
    iv. https://google.com/transparencyreport/https/ct/

A script that searches SSL/TLS certificates issued for a domain using crt.sh

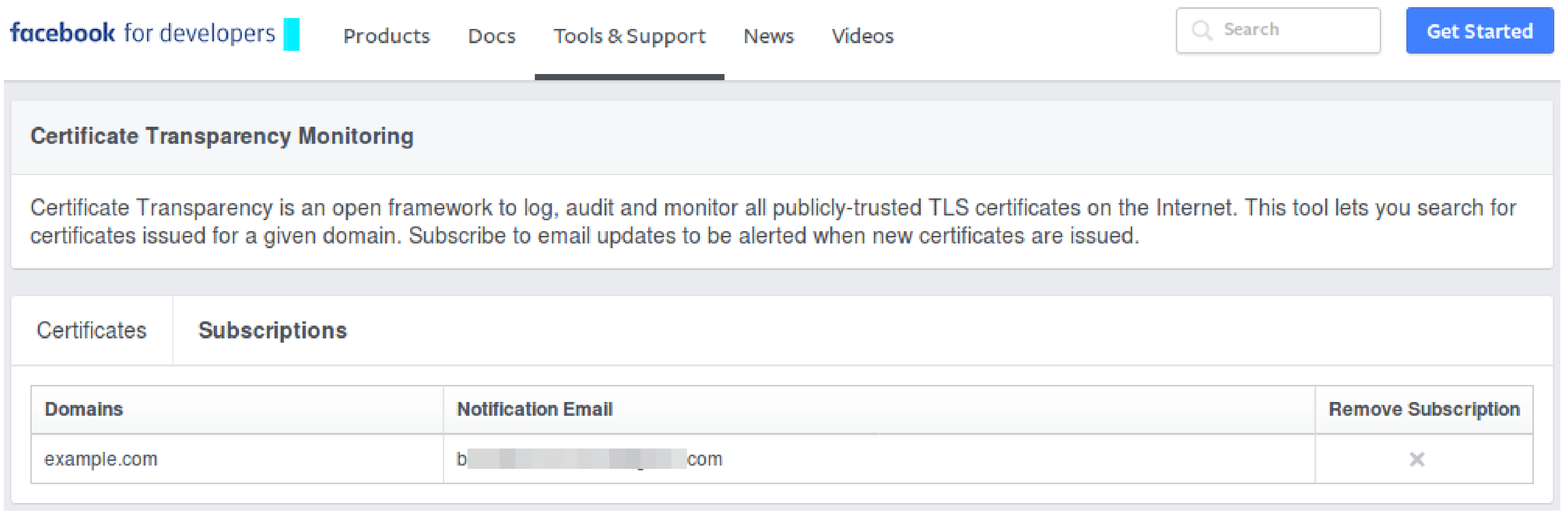


Criteria	Identity LIKE '%[redacted]

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

Script - https://git.io/vhqRd

Keeping track of an organisation's sub-domains



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

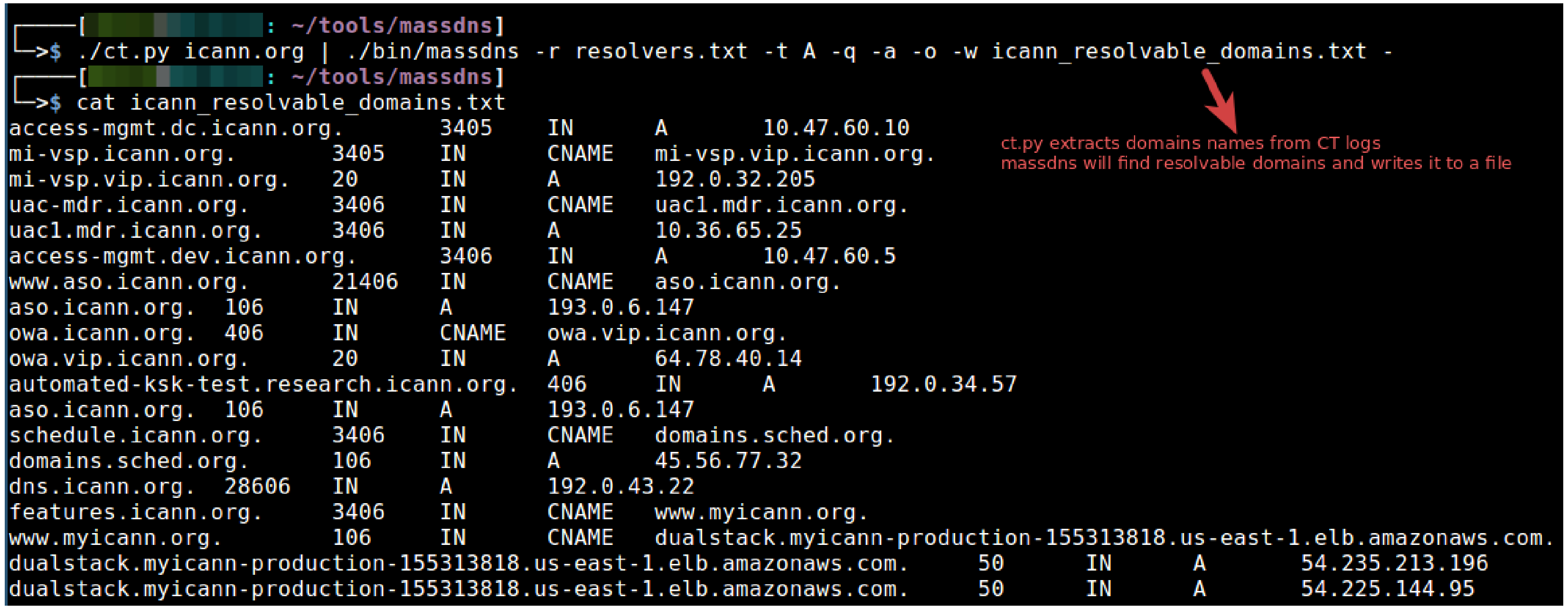
- CT logs are append-only. There is no way to delete an existing entry
- The domain names found in the CT logs may not exist anymore and thus they can’t be resolved to an IP address

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 results.txt -
```



```
┌──[[redacted]: ~/tools/massdns]
└─>$ ./ct.py icann.org | ./bin/massdns -r resolvers.txt -t A -q -a -o -w icann_resolvable_domains.txt -
┌──[[redacted]: ~/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

Using certspotter

```
find-cert()
{ curl -s https://certspotter.com/api/v0/certs?domain=$1 | jq -c '.[].dns_names' | grep -o '"[^"]\-
```

Using certdb.com

- While `crt.sh` gets the data from CT logs only where "legit" CA submit the certs to a log; CertDB is based on the scanning the IPv4 segment, domains and "finding & analyzing" all the certificates

```
curl -L -sd "api_key=API-KEY&q=Organization:\"tesla\"&response_type=3" -X POST https://certdb.com/a
```

https://certdb.com

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

Censys.io

- Censys aggregates SSL certificates that are a result of SSL scans on IPv4 address space and also from Certificate Transparency (CT) logs
- This is a good source of domains and also email addresses

https://0xpatrik.com/censys-guide/

Extracting domains/emails from SSL/TLS certs using censys

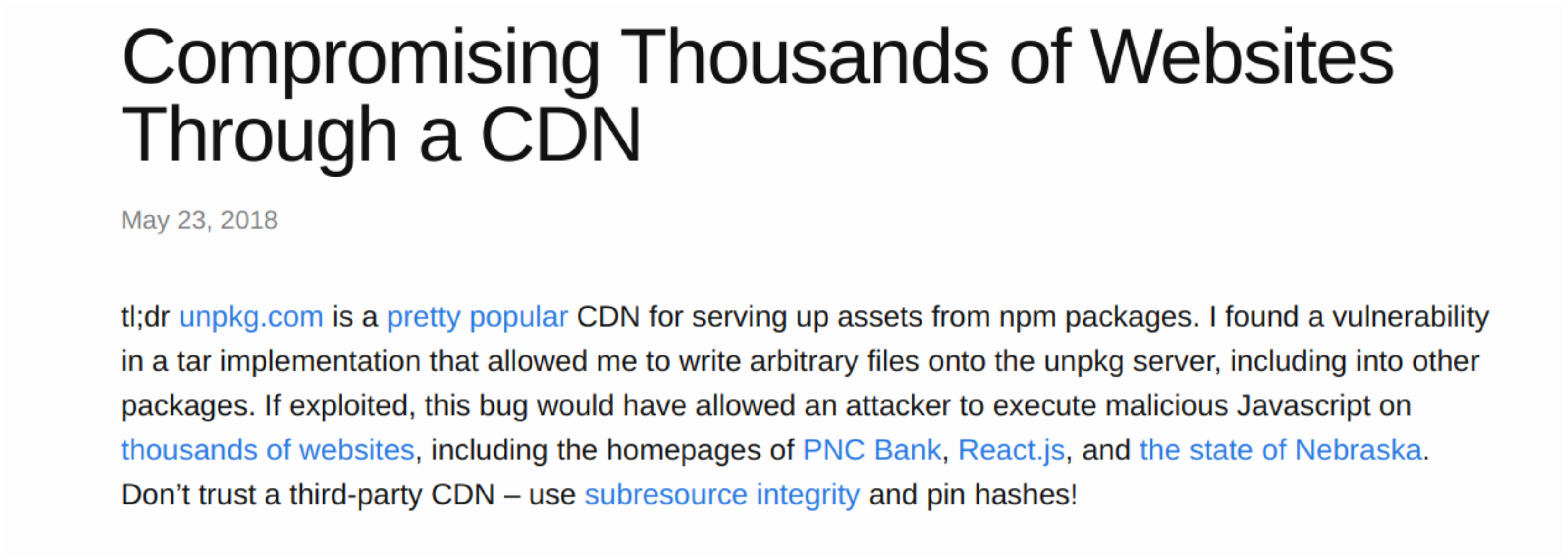
https://github.com/0xbharath/censys-enumeration

Content Security Policy(CSP)

- Content Security Policy(CSP) defines the `Content-Security-Policy` HTTP header, which allows us to create a whitelist of sources of trusted content, and instructs the browser to only execute or render resources from those sources
- So basically, Content-Security-Policy header will list a bunch of sources(domains) that might be of interest to us as an attackers.

Extract domains from CSP headers

https://github.com/0xbharath/domains-from-csp



Compromising Thousands of Websites Through a CDN

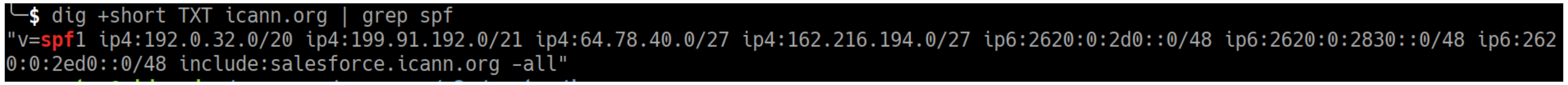
May 23, 2018

tl;dr unpkg.com is a pretty popular CDN for serving up assets from npm packages. I found a vulnerability in a tar implementation that allowed me to write arbitrary files onto the unpkg server, including into other packages. If exploited, this bug would have allowed an attacker to execute malicious Javascript on thousands of websites, including the homepages of PNC Bank, React.js, and the state of Nebraska. Don't trust a third-party CDN – use subresource integrity and pin hashes!

https://justi.cz/security/2018/05/23/cdn-tar-oops.html

Sender Policy Framework

- A Sender Policy Framework(SPF) record and is used to indicate to recieving mail exchanges which hosts are authorized to send mail for a given domain
- Simply put, an SPF record lists all the hosts that are authorised send emails on behalf of a domain



```
└─$ dig +short TXT icann.org | grep spf
"v=spf1 ip4:192.0.32.0/20 ip4:199.91.192.0/21 ip4:64.78.40.0/27 ip4:162.216.194.0/27 ip6:2620:0:2d0::0/48 ip6:2620:0:2830::0/48 ip6:262
0:0:2ed0::0/48 include:salesforce.icann.org -all"
```

Extract net blocks/domains from SPF record

https://github.com/0xbharath/assets-from-spf

Domain enumeration in 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.

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
```

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
```

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 ABCDEF
NKDO8UKT2STOL6EJRD1EKVD1BQ2688DM A NS SOA TXT AAAA RRSIG DNSKEY NSEC3PARAM
```

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

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 hases 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 sucessfully 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.
stream.icann.org.
tally.icann.org.
video.icann.org.
mm.icann.org.
ns.icann.org.
qa.icann.org.
ist.icann.org.
aso.icann.org.
cai.icann.org.
dev.icann.org.
exc.icann.org.
jss.icann.org.
mex.icann.org.
rrs.icann.org.
```

```
syd.icann.org.
upk.icann.org.
vip.icann.org.
crm.icann.org.
dns.icann.org.
liao.icann.org.
redis.icann.org.
svn.icann.org.
admin.icann.org.
orbis.icann.org.
jira.icann.org.
omblog.icann.org.
pptr.icann.org.
splunk.icann.org.
nomcom.icann.org.
rssac.icann.org.
sftp.icann.org.
netscan.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 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.
  i. AWS S3 buckets
  ii. Digital Ocean Spaces

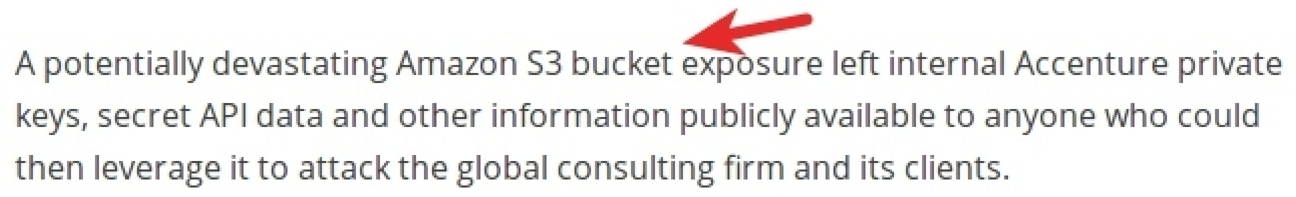
What's the catch with object storage?

- 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.

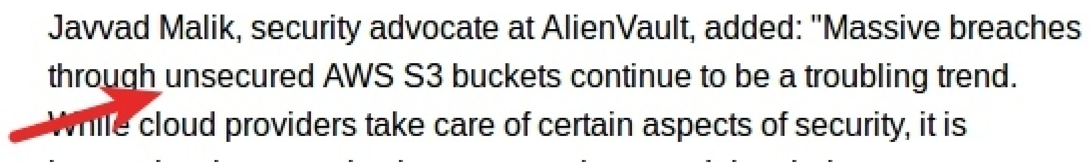
Attack on Accenture(Sep, 2017)- AWS S3 buckets as attack surface



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.

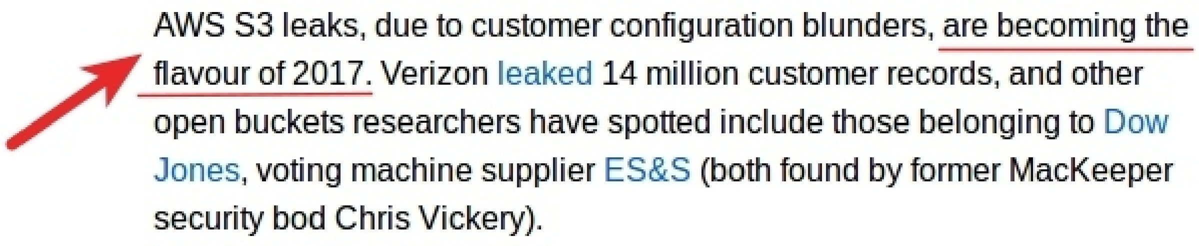
https://www.upguard.com/breaches/cloud-leak-accenture

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

  i. AWSBucketDump
  ii. Slurp

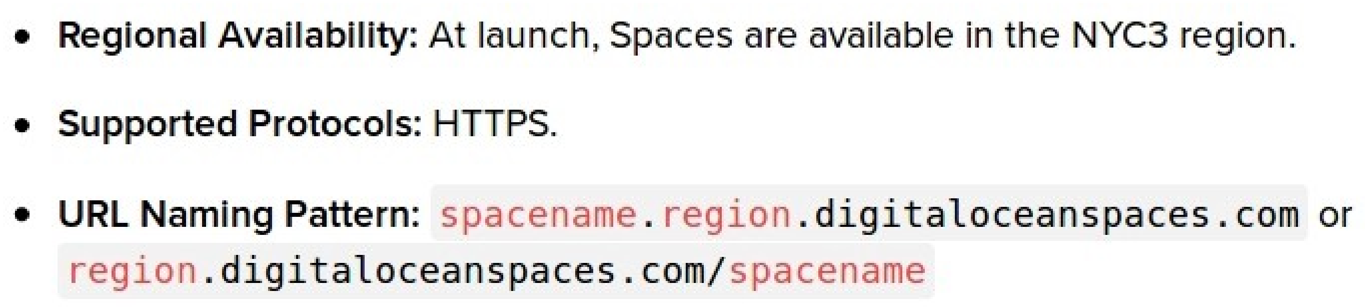
Digital Ocean Spaces

- 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

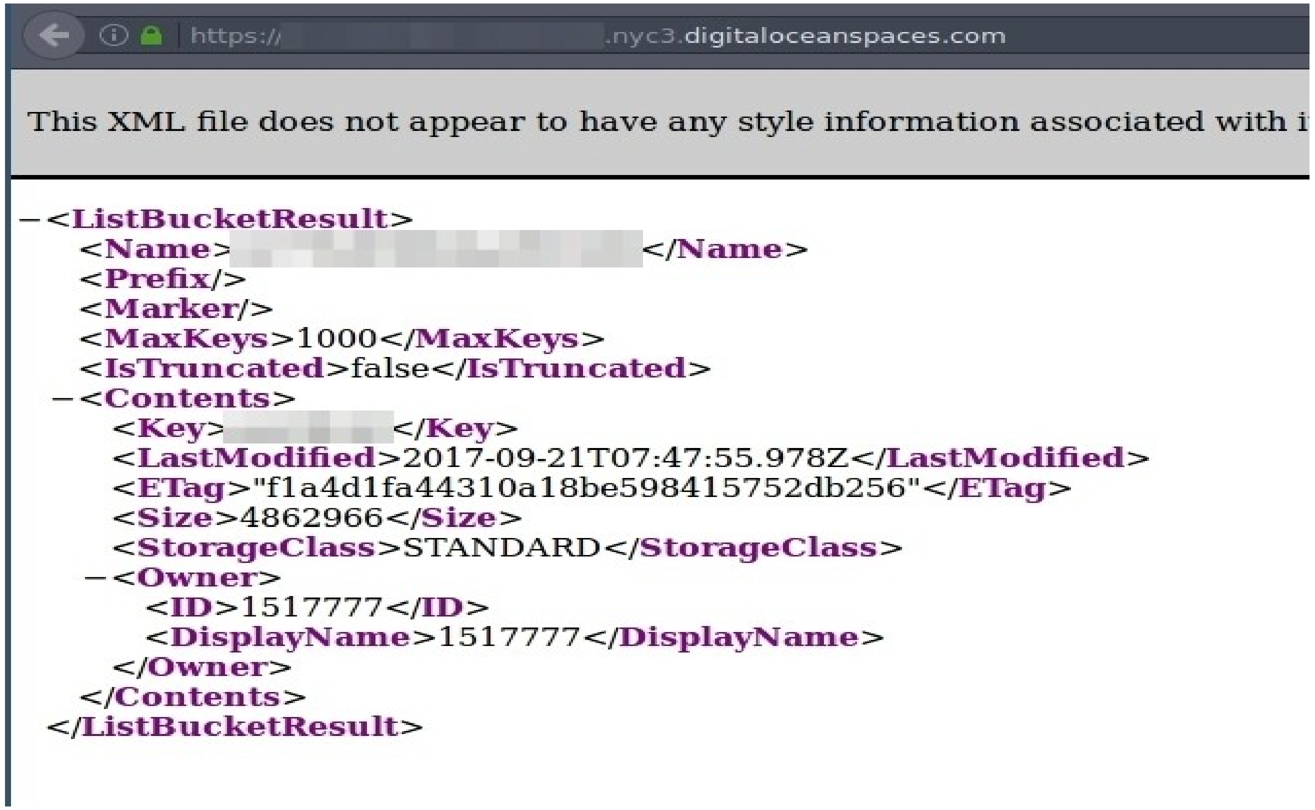
- Users can store Files in a "Space"
- Each Space will get an unique, predictable URL
- Each file in a Space will get an unique URL as well.
- Access controls mechanisms are available at Space and file level.



- **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

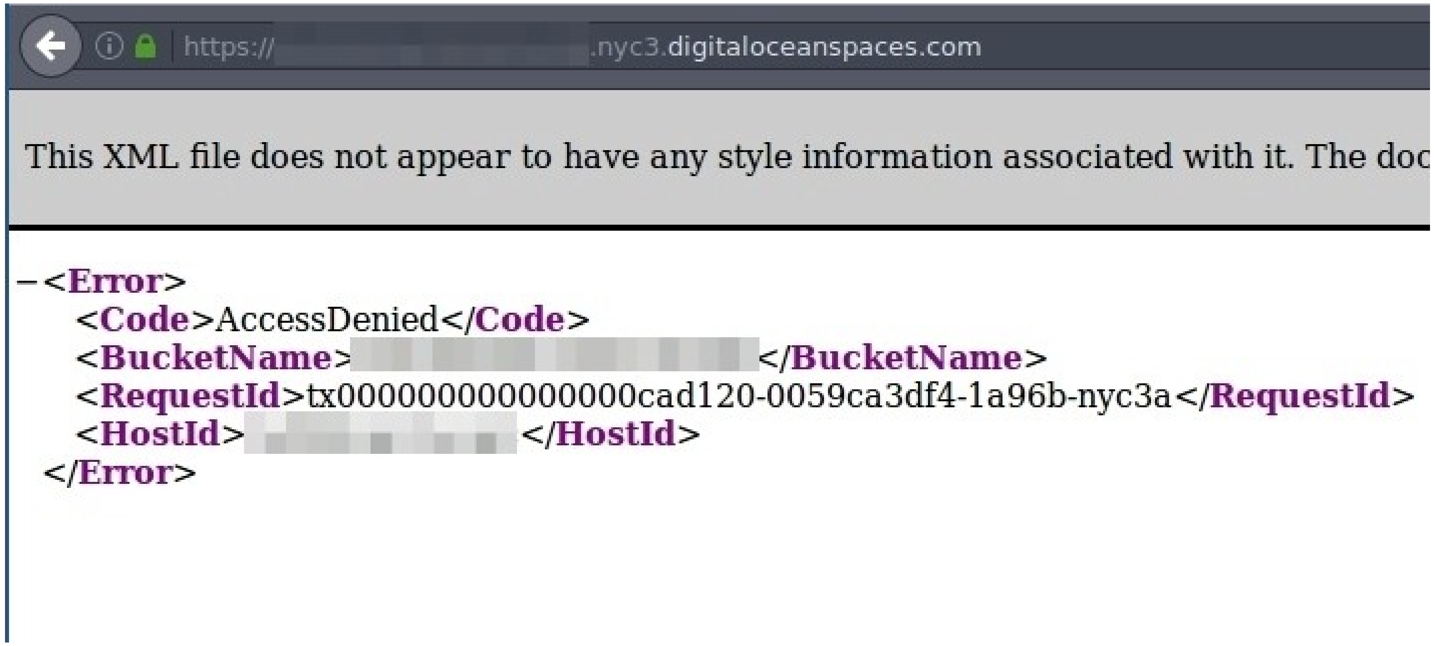


https://[redacted].nyc3.digitaloceanspaces.com

This XML file does not appear to have any style information associated with i

```
–<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


https://[redacted].nyc3.digitaloceanspaces.com

This XML file does not appear to have any style information associated with it. The doc

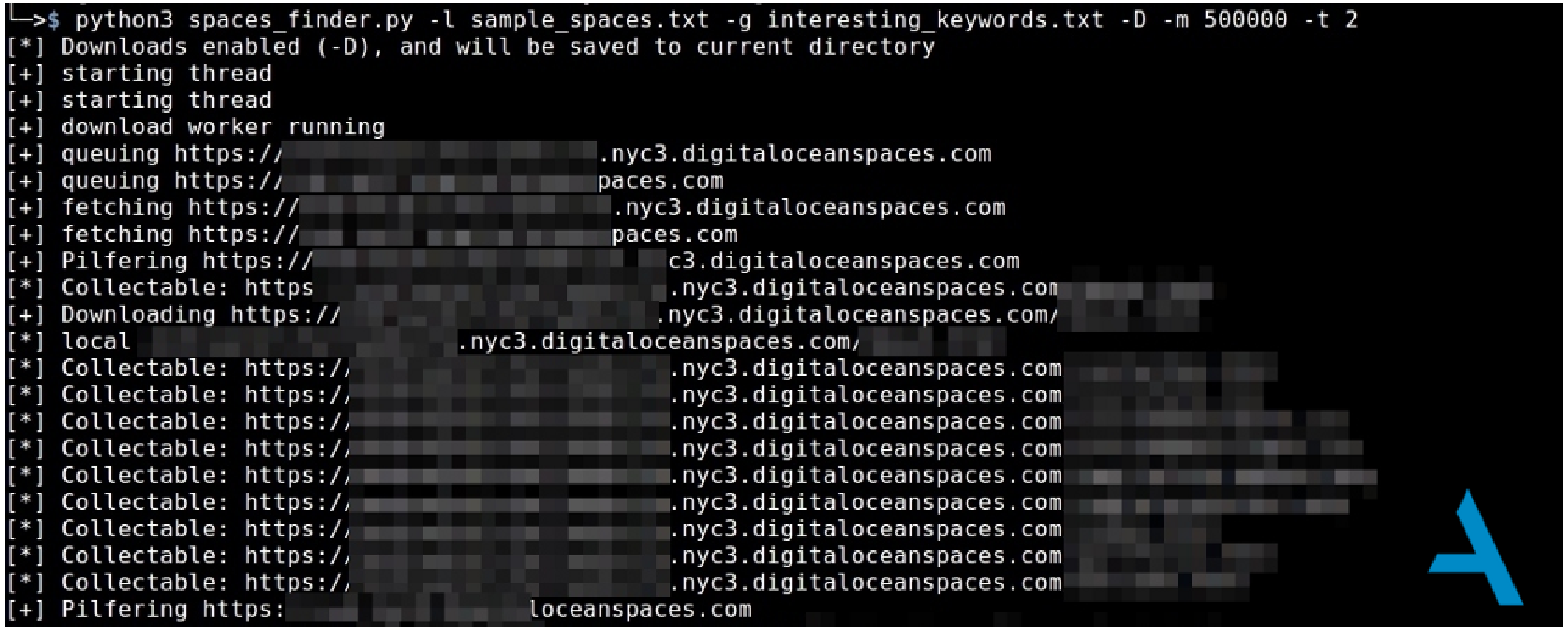
```
-<Error>
   <Code>AccessDenied</Code>
   <BucketName>[redacted]</BucketName>
   <RequestId>tx00000000000000cad120-0059ca3df4-1a96b-nyc3a</RequestId>
   <HostId>[redacted]</HostId>
 </Error>
```

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[redacted]
[+] Downloading https://[redacted].nyc3.digitaloceanspaces.com/[redacted]
[*] local [redacted].nyc3.digitaloceanspaces.com/[redacted]
[*] Collectable: https:/[redacted].nyc3.digitaloceanspaces.com[redacted]
[*] Collectable: https:/[redacted].nyc3.digitaloceanspaces.com[redacted]
[*] Collectable: https:/[redacted].nyc3.digitaloceanspaces.com[redacted]
[*] Collectable: https:/[redacted].nyc3.digitaloceanspaces.com[redacted]
[*] Collectable: https:/[redacted].nyc3.digitaloceanspaces.com[redacted]
[*] Collectable: https:/[redacted].nyc3.digitaloceanspaces.com[redacted]
[*] Collectable: https:/[redacted].nyc3.digitaloceanspaces.com[redacted]
[*] Collectable: https:/[redacted].nyc3.digitaloceanspaces.com[redacted]
[*] Collectable: https:/[redacted].nyc3.digitaloceanspaces.com[redacted]
[+] Pilfering https:[redacted]loceanspaces.com
```

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

Authentication

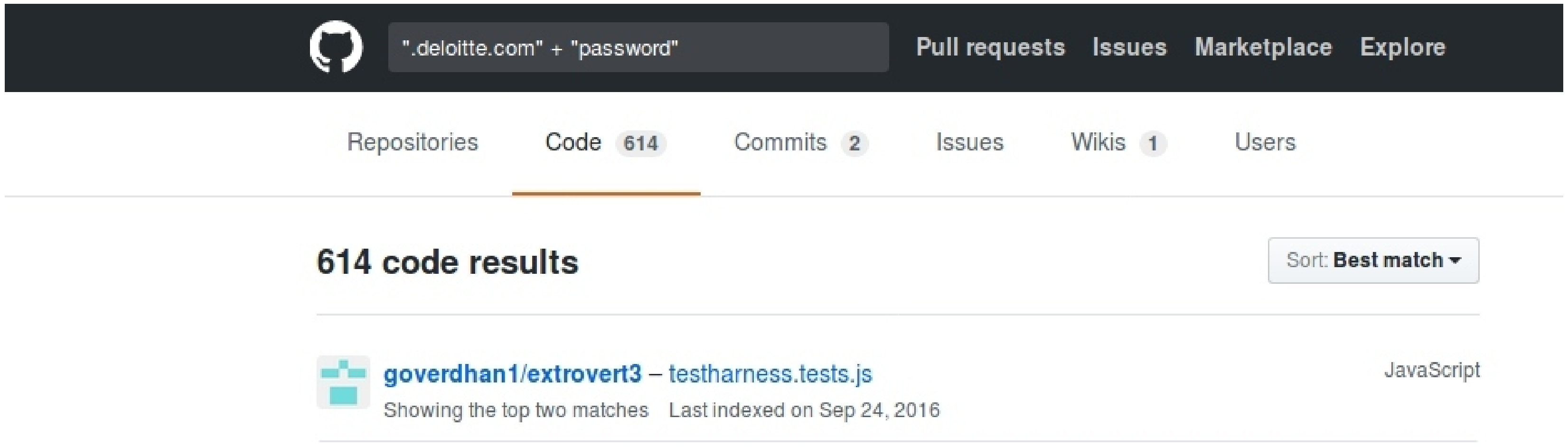
- With almost every service exposing an API, keys have become critical in authenticating
- API keys are treated as keys to the kingdom
- For applications, API keys tend to be achilles heel

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


".deloitte.com" + "password"

Repositories | Code 614 | Commits 2 | Issues | Wikis 1 | Users

614 code results

Sort: Best match

goverdhan1/extrovert3 – testharness.tests.js
Showing the top two matches   Last indexed on Sep 24, 2016
JavaScript

Github for recon

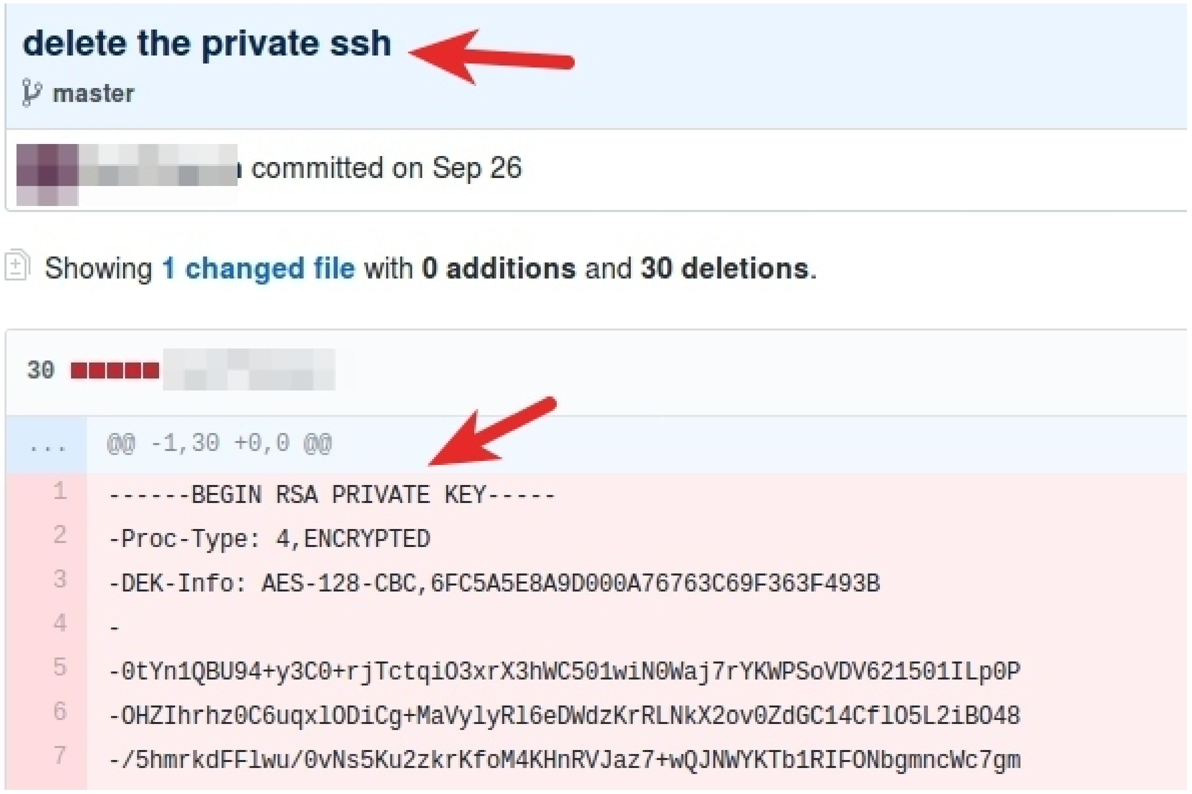
- GitHub is an extremely popular version control and collaboration platform
- Code repos on github tend to have all sorts of sensitive information
- Github also has a powerful search feature with advanced operators
- Github has a very well designed REST API
- edoverflow has a neat little guide on GitHub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

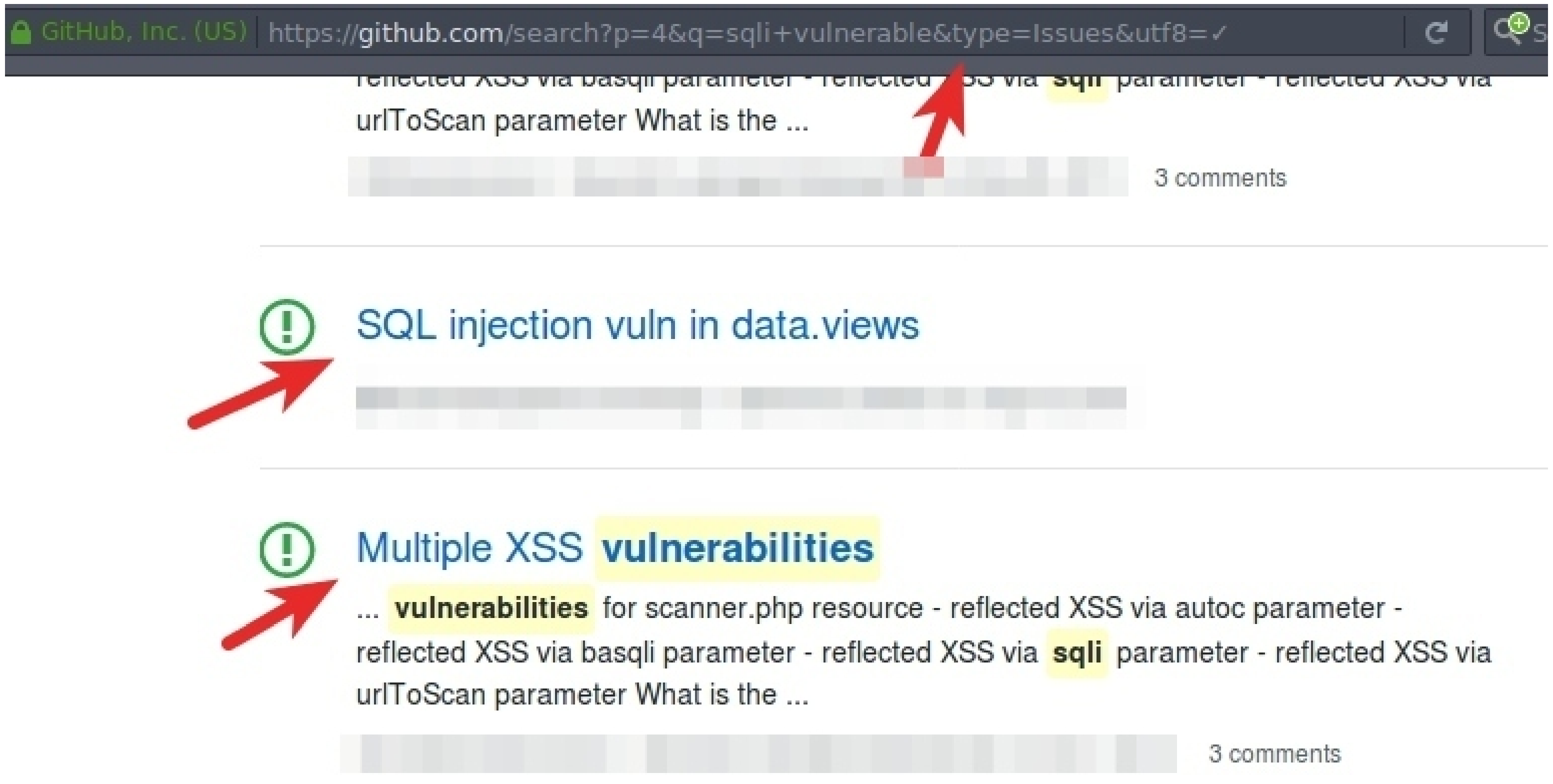
master

[redacted] committed on Sep 26

Showing 1 changed file with 0 additions and 30 deletions.

30 [redacted]

```
...  @@ -1,30 +0,0 @@
------BEGIN RSA PRIVATE KEY-----
-Proc-Type: 4,ENCRYPTED
-DEK-Info: AES-128-CBC,6FC5A5E8A9D000A76763C69F363F493B
-
-0tYn1QBU94+y3C0+rjTctqiO3xrX3hWC501wiN0Waj7rYKWPSoVDV621501ILp0P
-OHZIhrhz0C6uqxlODiCg+MaVylyRl6eDWdzKrRLNkX2ov0ZdGC14CflO5L2iBO48
-/5hmrkdFFlwu/0vNs5Ku2zkrKfoM4KHnRVJaz7+wQJNWYKTb1RIFONbgmncWc7gm
```


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 ...

[redacted] 3 comments

SQL injection vuln in data.views

[redacted]

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 ...

[redacted] 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

   i. Brakeman for Ruby
   ii. Bandit for Python

Finding secrets in code manually

- 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://
```

Finding secrets in code in automated fashion

There are various tools available to find juicy information in source code.

1. Truffle Hog
2. git-all-secrets

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

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

https://github.com/fathom6/inetdata

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

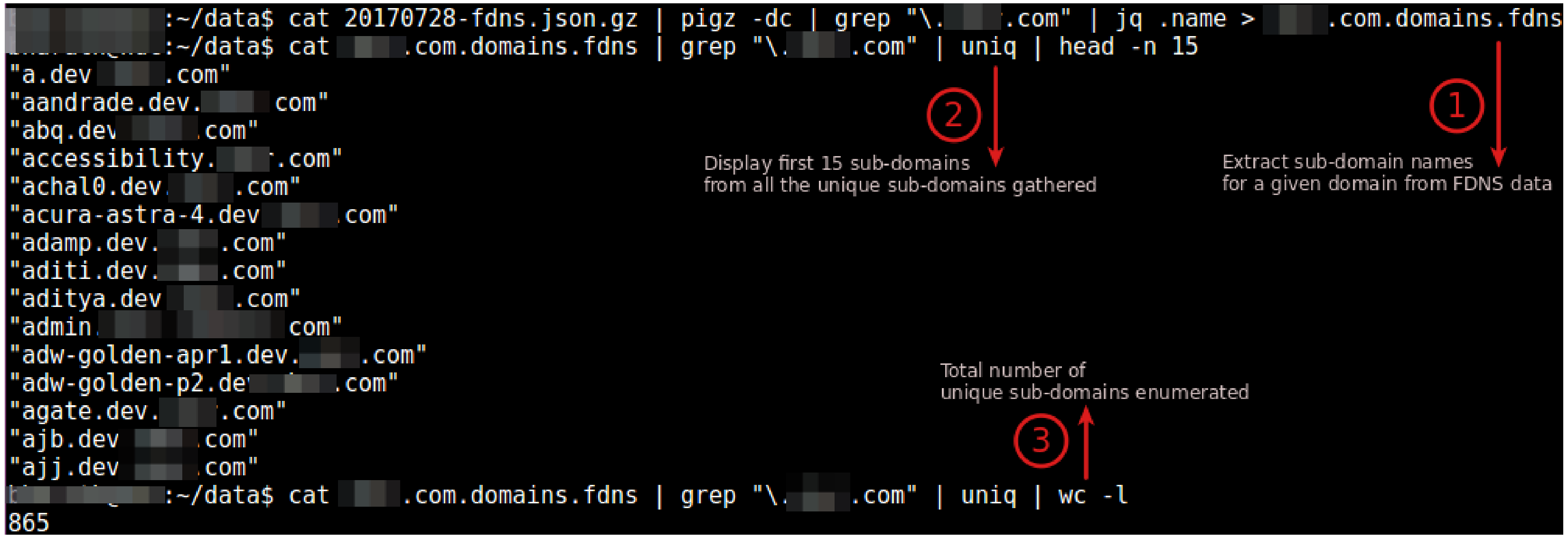
- The data format is a gzip-compressed JSON file so we can use `jq` utility to extract sub-domains of a specific domain:

```
curl --silent -L https://opendata.rapid7.com/sonar.fdns_v2/2018-04-21-1524297601-fdns_any.json.gz
```

```
cat 2018-04-21-1524297601-fdns_any.json.gz | pigz -dc | grep "\.example\.com" | jq .name
```

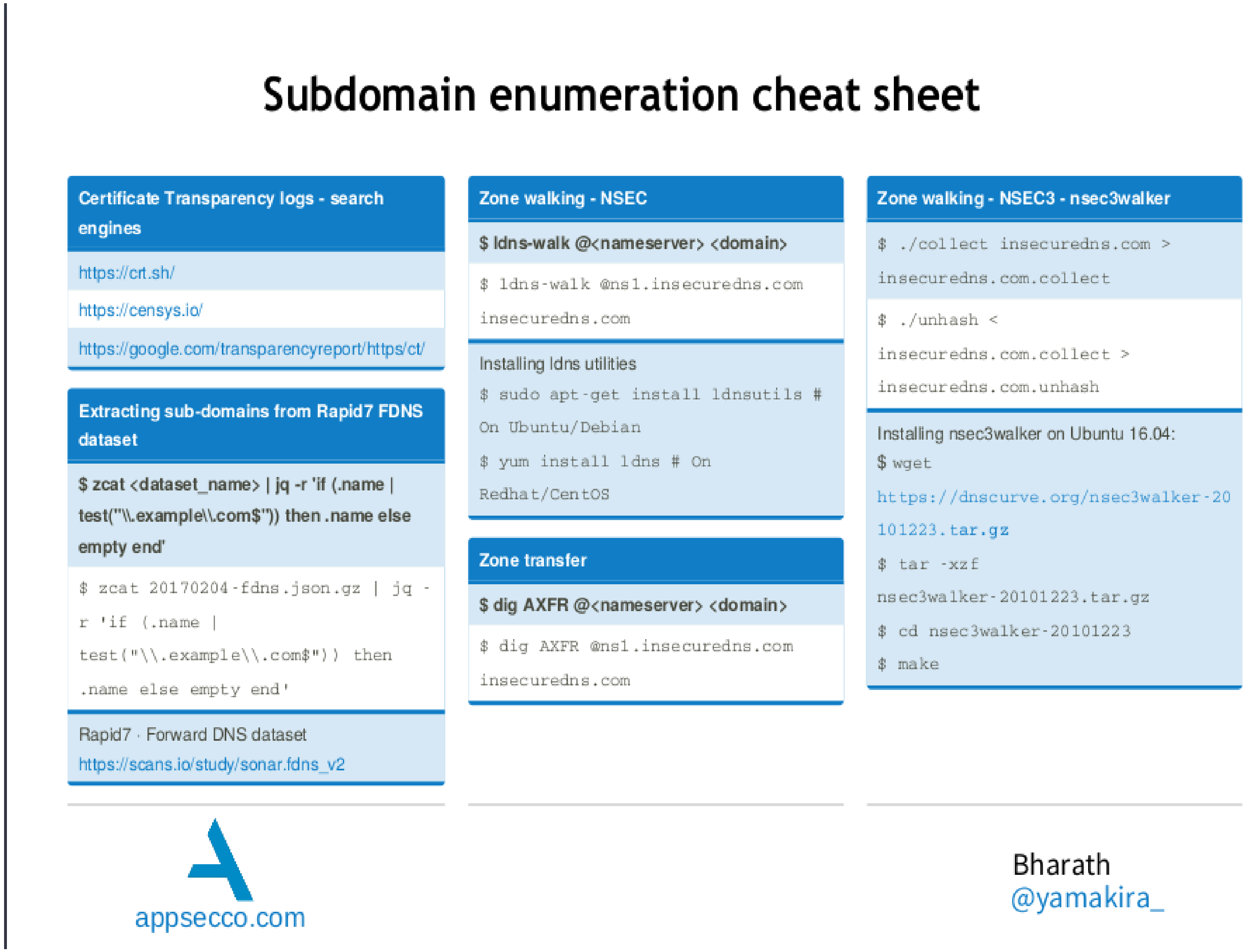
https://opendata.rapid7.com/about/

Hunting sub-domain in FDNS dataset



```
[redacted]:~/data$ cat 20170728-fdns.json.gz | pigz -dc | grep "\.[redacted].com" | jq .name > [redacted].com.domains.fdns
[redacted]:~/data$ cat [redacted].com.domains.fdns | grep "\.[redacted].com" | uniq | head -n 15
"a.dev[redacted].com"
"aandrade.dev.[redacted]com"
"abq.dev[redacted]com"
"accessibility.[redacted].com"
"achal0.dev[redacted].com"
"acura-astra-4.dev[redacted]com"
"adamp.dev.[redacted].com"
"aditi.dev.[redacted].com"
"aditya.dev[redacted].com"
"admin[redacted]com"
"adw-golden-apr1.dev.[redacted].com"
"adw-golden-p2.de[redacted].com"
"agate.dev.[redacted].com"
"ajb.dev[redacted]com"
"ajj.dev[redacted]com"
[redacted]:~/data$ cat [redacted].com.domains.fdns | grep "\.[redacted].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
Total number of unique sub-domains enumerated
3



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
```

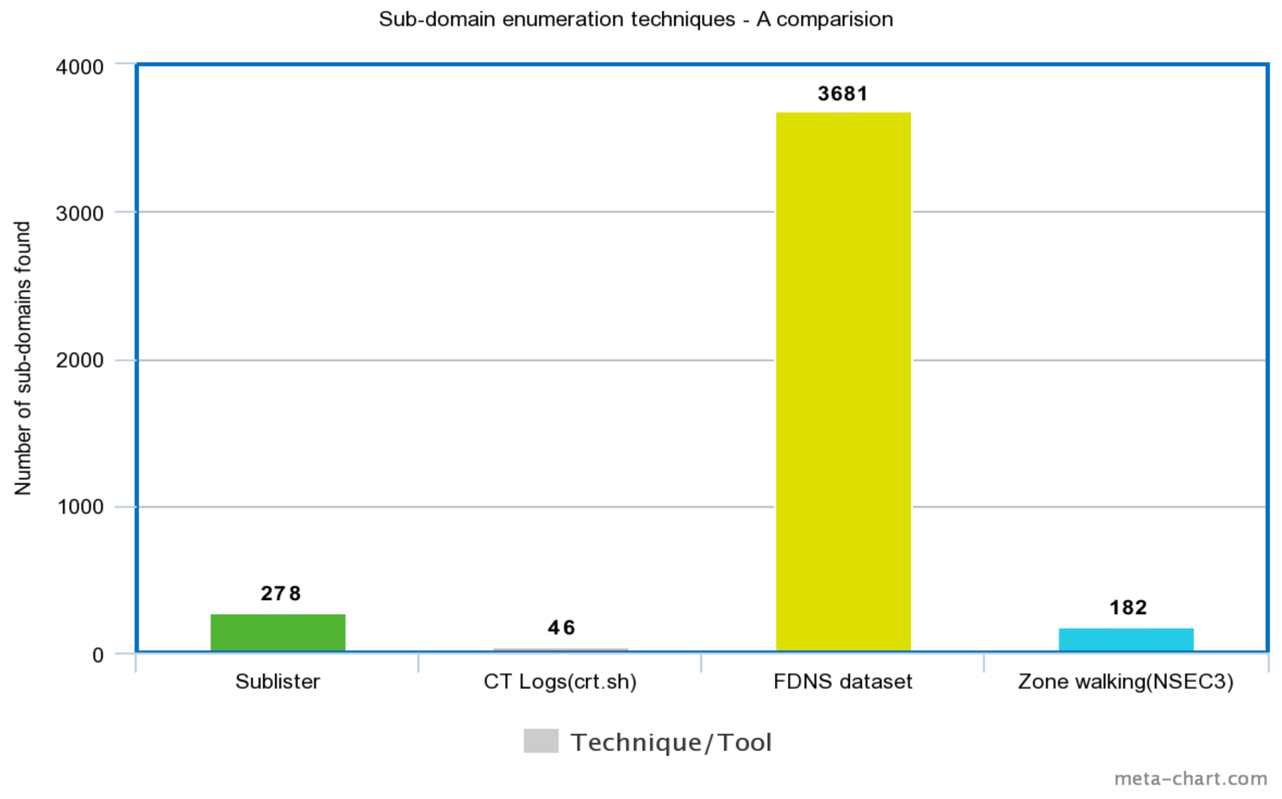
appsecco.com

Bharath
@yamakira_

https://github.com/appsecco/bugcrowd-levelup-subdomain-enumeration

ICANN.ORG subdomains

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



TALK MATERIAL

https://github.com/appsecco/practical-recon-levelup0x02

Take away

A gitbook on sub-domain enumeration

https://appsecco.com/books/subdomain-enumeration/

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

Thanks

@0xbharath