Assets-from-spf

- Description
    - Parse net blocks & domain names from SPF records
    - https://github.com/yamakira/assets-from-spf
- Installation
- git clone https://github.com/yamakira/assets-from-spf.git
- pip install click ipwhois
- Usage
    - cd the-art-of-subdomain-enumeration; python assets_from_spf.py target.com
    - Options
        - --asn: Enable ASN enumeration

BiLE-suite

- Description
    - HTML parsing, reverse DNS, TLD expansion, horizontal domain correlation
    - https://github.com/sensepost/BiLE-suite
- Installation
- aptitude install httrack
- git clone https://github.com/sensepost/BiLE-suite.git
- Usage
    - List links related to a site: cd BiLE-suite; perl BiLE.pl target.com target
    -

Extract subdomains from the results of BiLe.pl: ` cat target.mine	grep -v “Link from”	cut -d’:’ -f2	grep target.co

Bing

- Search engine
- Usage
    - Find subsomains: site:target.com
    - Find subdomains & exclude specific ones: site:target.com -site:www.target.com

Censys_subdomain_enum.py

- Description
    - Extract domains & emails from SSL/TLS certs collected by Censys

    - https://github.com/appsecco/the-art-of-subdomain-enumeration/blob/master/censys_subdomain_enum.py

- Installation
- pip install censys
- git clone https://github.com/appsecco/the-art-of-subdomain-enumeration.git
    - Add your CENSYS API ID & SECRET to the-art-of-subdomain-enumeration/censys_subdomain_enum.py
- Usage
    - cd the-art-of-subdomain-enumeration; python censys_enumeration.py target.com

Cloudflare_enum.py

- Description
    - Extract subdomains from Cloudflare
    - DNS aggregator
    - https://github.com/appsecco/the-art-of-subdomain-enumeration/blob/master/cloudflare_subdomain_enum.py
- Installation
- pip install censys
- git clone https://github.com/appsecco/the-art-of-subdomain-enumeration.git
- Usage
    - the-art-of-subdomain-enumeration; python cloudflare_subdomain_enum.py your@cloudflare.email target.com

Crt_enum_psql.py

- Description
    - Query crt.sh postgres interface for subdomains
    - https://github.com/appsecco/the-art-of-subdomain-enumeration/blob/master/crt_enum_psql.py
- Installation
- pip install psycopg2
- git clone https://github.com/appsecco/the-art-of-subdomain-enumeration.git
- Usage
    - cd python the-art-of-subdomain-enumeration; python crtsh_enum_psql.py target.com

Crt_enum_web.py

- Description
    - Parse crt.sh web page for subdomains
    - https://github.com/appsecco/the-art-of-subdomain-enumeration/blob/master/crt_enum_web.py
- Installation
- pip install psycopg2
- git clone https://github.com/appsecco/the-art-of-subdomain-enumeration.git
- Usage
    - cd python the-art-of-subdomain-enumeration; python3 crtsh_enum_web.py target.com

CTFR

- Description
    - Enumerate subdomains using CT logs (crt.sh)
    - https://github.com/UnaPibaGeek/ctfr
- Installation
- git clone https://github.com/UnaPibaGeek/ctfr.git
- cd ctfr
- pip3 install -r requirements.txt
- Usage
    - cd ctfr; python3 ctfr.py -d target.com -o $outfile

Dig

- Description
    - Zone transfer, DNS lookups & reverse lookups
- Installation
    - Installed by default in Kali, otherwise:
    - aptitude instal dnsutils
- Usage dig +multi AXFR target.com dig +multi AXFR $ns_server target.com

Domains-from-csp

- Description
    - Extract domain names from Content Security Policy(CSP) headers
    - https://github.com/yamakira/domains-from-csp
- Installation

- git clone https://github.com/yamakira/domains-from-csp.git
- pip install click
- Usage
    - Parse CSP header for domains: cd domains-from-csp; python csp_parser.py $URL
    - Parse CSP header & resolve the domains: cd domains-from-csp; python csp_parser.py $URL -r

Dnscan

- Description
    - AXFR, brute force
    - https://github.com/rbsec/dnscan
- Install
- git clone https://github.com/rbsec/dnscan.git
- cd dnscan
- pip install -r requirements.txt
- Usage
    - Subdomain brute-force of a domain: dnscan.py -d target.com -o outfile -w $wordlist
    - Subdomain brute-force of domains listed in a file (one by line): dnscan.py -l $domains_file -o outfile -w $wordlist
    - Other options:
        - -i $file: Output discovered IP addresses to a text file
        - -r: Recursively scan subdomains
        - -T: TLD expansion

Dnsrecon

- Description
    - DNS zone transfer, DNS cache snooping, TLD expansion, SRV enumeration, DNS records enumeration, brute-force, check for Wildcard resolution, subdomain scraping, PTR record lookup, check DNS server cached records, mDNS records enumeration…
    - https://github.com/darkoperator/dnsrecon
- Installation
    - aptitude install dnsrecon on Kali, or:
    - git clone https://github.com/darkoperator/dnsrecon.git

    - cd dnsrecon
    - pip install -r requirements.txt
- Usage
    - Brute-force: dnsrecon -d target.com -D wordlist.txt -t brt
    - DNS cache snooping: dnsrecon -t snoop -D wordlist.txt -n 2.2.2.2 where 2.2.2.2 is the IP of the target's NS server
    - Options
        - --threads 8: Number of threads
        - -n nsserver.com: Use a custom name server
        - Output options
            - --db: SQLite 3 file
            - --xml: XML file
            - --json: JSON file
            - --csv: CSV file

Dnssearch

- Description
    - Subdomain brute-force
    - https://github.com/evilsocket/dnssearch
- Installation
- go get github.com/evilsocket/dnssearch
    - Add ~/go/bin/ to PATH by adding this line to ~/.profile: export PATH=$PATH:/home/mima/go/bin/
- Usage
    - dnssearch -domain target.com -wordlist $wordlist
    - Other options
        - -a bool: Lookup A records (default true)
        - -txt bool: Lookup TXT records (default false)
        - -cname bool: Show CNAME records (default false)
        - -consumers 10: Number of threads (default 8)

Domained

- Description

    - Wrapper for Sublist3r, Knock, Subbrute, Massdns, Recon-ng, Amass & SubFinder
    - https://github.com/cakinney/domained
- Installation
- git clone https://github.com/cakinney/domained.git
- cd domained
- pip install -r ./ext/requirements.txt
- python domained.py --install
- Usage
    - Run Sublist3r (+subbrute), enumall, Knock, Amass & SubFinder: python domained.py -d target.com
    - Run only Amass & Subfinder: python domained.py -d target.com --quick
    - Brute-force with massdns & subbrute with Seclist wordlist, plus Sublist3r, Amass, enumall & SubFinder: python domained.py -d target.com --b
    - Bruteforce with Jason Haddix's All.txt wordlist, plus Sublist3r, Amass, enumall & SubFinder: python domained.py -d target.com -b --bruteall
    - Other options
        - --notify: Send Pushover or Gmail notifications
        - --noeyewitness: No Eyewitness
        - --fresh: Delete old data from output folder

Fierce

- Description
    - AXFR, brute force, reverse DNS
    - https://github.com/bbhunter/fierce-domain-scanner (original link not available anymore)
- Installation
    - Installed by default on Kali
- Usage fierce -dns target.com

Gobuster

- Description
    - todo
    - https://github.com/OJ/gobuster
- Installation

- git clone https://github.com/OJ/gobuster.git
- cd gobuster/
- go get && go build
- go install
- Usage
    - gobuster -m dns -u target.com -w $wordlist
    - Other options:
        - -i: Show IP addresses
        - -t 50: Number of threads (default 10)

Google

- Search engine
- Usage
    - Find subsomains: site:*.target.com
    - Find subdomains & exclude specific ones: site:*.target.com -site:www.target.com -site:help.target.com

Knock

- Description
    - AXFR, virustotal, brute-force
    - https://github.com/guelfoweb/knock
- Install
- apt-get install python-dnspython
- git clone https://github.com/guelfoweb/knock.git
- cd knock
- nano knockpy/config.json # <- set your virustotal API_KEY
- python setup.py install
- Usage
    - Use default wordlist: knockpy target.com
    - Use custom wordlist: knockpy target.com -w $wordlist
    - Resolve domain name & get response headers: knockpy -r target.com or knockpy -r $ip
    - Save scan output in CSV: knockpy -c target.com
    - Export full report in JSON: knockpy -j target.com

Ldns-walk

- Description
  - DNSSEC zone walking
- Installation
  - aptitude install ldnsutils
- Usage
  - Detect if DNSSEC NSEC or NSEC3 is used:
    - ldns-walk target.com
    - ldns-walk @nsserver.com target.com
  - If DNSSEC NSEC is enabled, you'll get all the domains
  - If DNSSEC NSEC3 is enabled, use Nsec3walker

Massdns

- Description
  - DNS resolver
  - https://github.com/blechschmidt/massdns
- Installation
- git clone https://github.com/blechschmidt/massdns.git
- cd massdns/
- make
- Usage
  - Resolve domains: cd massdns; ./bin/massdns -r lists/resolvers.txt -t AAAA -w results.txt domains.txt -o S -w output.txt
  - Subdomain brute-force: ./scripts/subbrute.py wordlist.txt target.com | ./bin/massdns -r lists/resolvers.txt -t A -o S -w output.txt
  - Get subdomains with CT logs parser & resolve them with Massdns: ./scripts/ct.py target.com | ./bin/massdns -r lists/resolvers.txt -t A -o S -w output.txt
  - Other options:
    - -s 5000: Number of concurrent lookups (default 10000)
    - -t A (default), -t AAAA, -t PTR…: Type of DNS records to retrieve
    - Output options
      - -o S -w output.txt: Save output as simple text

        - -o F: Save output as full text
        - -o J: Save output as ndjson

Nsec3walker

- Description
    - DNSSEC NSEC3 zone walking
    - https://dnscurve.org/nsec3walker.html
- Installation
- wget https://dnscurve.org/nsec3walker-20101223.tar.gz
- tar -xzf nsec3walker-20101223.tar.gz
- cd nsec3walker-20101223
- make
- Usage
- ./collect target.com > target.com.collect
- ./unhash target.com.collect > target.com.unhash
- cat target.com.unhash | grep "target" | wc -l
- cat target.com.unhash | grep "target" | awk '{print $2;}'

Rapid7 Forward DNS dataset (Project Sonar)

- Description
    - Public dataset containing the responses to DNS requests for all forward DNS names known by Rapid7's Project Sonar
    - https://opendata.rapid7.com/sonar.fdns_v2/
- Installation
    - aptitude install jq pigz
- Usage
- wget https://scans.io/data/rapid7/sonar.fdns_v2/20170417-fdns.json.gz
- cat 20170417-fdns.json.gz | pigz -dc | grep ".target.org" | jq`

San_subdomain_enum.py

- Description
    - Extract subdomains listed in Subject Alternate Name(SAN) of SSL/TLS certificates
    - https://github.com/appsecco/the-art-of-subdomain-enumeration/blob/master/san_subdomain_enum.py

- Installation
    - git clone https://github.com/appsecco/the-art-of-subdomain-enumeration.git
- Usage
    - cd python the-art-of-subdomain-enumeration; ./san_subdomain_enum.py target.com

Second Order

- Description
    - Second-order subdomain takeover scanner
    - Can also be leveraged as an HTML parser to enumerate subdomains
    - https://github.com/mhmdiaa/second-order
- Installation
    - go get github.com/mhmdiaa/second-order
- Usage
    - Create a new copy of the default config.json file: cp ~/go/src/github.com/mhmdiaa/second-order/config.json ~/go/src/github.com/mhmdiaa/second-order/config-subs-enum.json
    - And edit ` ~/go/src/github.com/mhmdiaa/second-order/config-subs-enum.json to replace “LogCrawledURLs”: false with “LogCrawledURLs”: true`
    - second-order -base https://target.com -config config.json -output target.com
    - Look for new subdomains in the resulting folder (./target.com)

Subbrute

- Description
    - Brute-force
    - https://github.com/TheRook/subbrute
- Installation
- aptitude install python-dnspython
- git clone https://github.com/TheRook/subbrute.git
- Usage
    - Test a single domain: ./subbrute.py target.com
    - Test multiple domains: ./subbrute.py target1.com target2.com
    - Test a list of domains: ./subbrute.py -t domains.txt
    - Enumerate subdomains, then their own subdomains:

    - ./subbrute.py target.com > target.out
    - ./subbrute.py -t target.out
    - Other options
        - -s wordlist.txt: Use a custom subdomains wordlist
        - -p: Print data from DNS records
        - -o outfile.txt: Save output in Greppable format
        - -j JSON: Save output to JSON file
        - -c 10: Number of threads (default 8)
        - -r resolvers.txt: Use a custom list of DNS resolvers

Subfinder

- Description
    - VirusTotal, PassiveTotal, SecurityTrails, Censys, Riddler, Shodan, Bruteforce
    - https://github.com/subfinder/subfinder
- Installation:
    - go get github.com/subfinder/subfinder
    - Configure API keys: ./subfinder --set-config VirustotalAPIKey=0x41414141
- Usage
    - Scraping: ./subfinder -d target.com -o $outfile
    - Scraping & brute-force: subfinder -b -d target.com -w $wordlist -o $outfile
    - Brute-force only: ./subfinder --no-passive -d target.com -b -w $wordlist -o $outfie
    - Other options:
        - -t 100: Number of threads (default 10)
        - -r 8.8.8.8,1.1.1.1 or -rL resolvers.txt: Use custom resolvers
        - -nW: Exclude wildcard subdomains
        - -recursive: Use recursion
        - -o $outfile -oJ: JSON output

Sublist3r

- Description
    - Baidu, Yahoo, Google, Bing, Ask, Netcraft, DNSdumpster, VirusTotal, Threat Crowd, SSL Certificates, PassiveDNS
    - https://github.com/aboul3la/Sublist3r

- Installation
- git clone https://github.com/aboul3la/Sublist3r.git
- cd Sublist3r
- pip install -r requirements.txt
- Usage
    - Scraping: ./sublist3r.py -d target.com -o $outfile
    - Bruteforce: ./sublist3r.py -b -d target.com -o $outfile
    - Other options:
        - -p 80,443: Show only subdomains which have open ports 80 and 443

Theharvester

- Description
    - Tool for gathering subdomain names, e-mail addresses, virtual hosts, open ports/ banners, and employee names from different public sources
    - Scraping, Brute-force, Reverse DNS, TLD expansion
    - Scraping sources: Threatcrowd, Crtsh, Google, googleCSE, google-profiles, Bing, Bingapi, Dogpile, PGP, LinkedIn, vhost, Twitter, GooglePlus, Yahoo, Baidu, Shodan, Hunter
    - https://github.com/laramies/theHarvester
- Installation
    - aptitude install theharvester
- Usage
    - Scraping: theharvester -d target.com -b all
    - Other options:
        - -h output.html: Save output to HTML file
        - -f output.html: Save output to HTML & XML files
        - -t: Also do TLD expansion discovery
        - -c: Also do subdomain bruteforce
        - -n: Also do a DNS reverse query on all ranges discovered

vhost-brute

- Description
    - vhosts brute-force
    - https://github.com/gwen001/vhost-brute

- Installation
- aptitude install php-curl
- git clone https://github.com/gwen001/vhost-brute.git
- Usage
  - php vhost-brute.php --ip=$ip --domain=target.com --wordlist=$outfile
  - Other options:
    - --threads=5: Maximum threads (default 1)
    - --port: Set port
    - --ssl: Force SSL

Virtual-host-discovery

- Description
  - vhosts brute-force
  - https://github.com/jobertabma/virtual-host-discovery
- Installation
  - git clone https://github.com/jobertabma/virtual-host-discovery.git
- Usage
  - cd virtual-host-discover; ruby scan.rb --ip=1.1.1.1 --host=target.com --output output.txt
  - Other options
    - --ssl=on: Enable SSL
    - --port 8080: Use a custom port
    - --wordlist wordlist.txt: Use a custom wordlist

Virustotal_subdomain_enum.py

- Description
  - Query VirusTotal API for subdomains
  - DNS aggregator
  - https://github.com/appsecco/the-art-of-subdomain-enumeration/blob/master/virustotal_subdomain_enum.py
- Installation
  - git clone https://github.com/appsecco/the-art-of-subdomain-enumeration.git
- Usage
  - python virustotal_subdomain_enum.py target.com 40