Bug hunting by NoobsheiK

BASIC APPROACHING

1. Recon on the scope

2. Trying to find exploits for possible vulnerabilities found during recon and reporting it.(For eg sensitive data exposure or open ports)

3. Low hanger bugs(For eg No rate limiting, failure to invalidate session)

4. Technology/Service Bugs (For eg: Jira Vulns)

5. Authentication and session management and Bugs.

6. From the urls you have gathered try bugs like Reflected XSS, Open Redirect, etc param based bugs.

7. Hunt for Stored XSS, SSRF, Command Injection kind of bugs which require manual testing.

8. Business Logic Bugs, like IDOR, CSRF, Parameter Tampering, Interesting account takeover

9. Github Recon

=======================================================================================SUBDOMAIN ENUMERATION================================================================================================================================

Step 1: subdomain enum

subscraper ryncoorchids.com --censys-api 41dfdb28-3929-42d5-9d48-201a8539a460 --censys-secret TaPbSuRSSo42QnlRdaUj4m1Swy8QMIWt -o subs.txt

Step 2: Github subdomain

go

github-subdomains -d target.com –t ghp\_1ydX99NxsLqa68YgEivtAlurSbNInL2Zjcoo -o subs2.txt

Step 3: HTTPROBE

cat subdomains.txt | httprobe -p http:81 -p http:8008 -p https:8008 -p https:3001 -p http:8000 -p https:8000 -p http:8080 -p https:8080 -p https:8443 -p https:10000 -p -s -c 100 | tee uncommon\_probe.txt

cat subdomains.txt | httprobe | tee -a host.txt

Step 4: subfinder:-

subfinder -d example.com -t 100 -v -o allsub.txt

Step 4: amaas:-

amass enum -src -ip -brute -d example.com -o allsub1.txt

Step 5: Crt.sh:-

./cert.sh example.com > /home/noobsheik/Desktop/allsub3.txt

Step 6: Sorting:-

sort allsub.txt allsub1.txt allsub3.txt | uniq -u > finalsubdomain.txt

Step 7: Httpx:-

cat finalsubdomain.txt | httpx 200 | tee -a subdomain.txt

============================================================================================URL ENUMARATION=================================================================================================================================

Step 8: GAUPLUS

cat target.txt | gauplus -o urls3.txt

Step 8.1: Crawler

cat host.txt | crawler | tee -a endpoint.txt

Step 9: Arjun params

arjun -i finalSUBS.txt -t 100 --passive -oT arjun\_params.txt

Step 10: Waybackurls:-

cat subdomain.txt | waybackurls | tee -a waybackurls.txt

cat host.txt | waybackurl | tee -a endpoint.txt

Step 11: Param spider:-

python3 paramspider.py --domain example.com -o paramslist.txt

Step 12: sort allurls.txt allurls1.txt allurl3.txt | uniq -u > FinalURLS.txt

=========================================================Directory Enumaration============================================================================================================

Step 13: Dirsearch

dirsearch -u https://example.com/ -e php,asp,aspx,jsp,py,txt,conf,config,bak,backup,swp,old,db,sqlasp,aspx,aspx~,asp~,py,py~,rb,rb~,php,php~,bak,bkp,cache,cgi,conf,csv,html,inc,jar,js,json,jsp,jsp~,lock,log,rar,old,sql,sql.gz,sql.zip,sql.tar.gz,sql~,swp,swp~,tar,tar.bz2,tar.gz,txt,wadl,zip,json,html,css,java,xml -i 200 --full-url

Step 14: dibuster

==========================================================Parameter sorting according to your Need (Bug)================================================================================================

Step 15: GF tool

cat FinalURLlist.txt | gf sqli > sqli.txt

Step 16:Find xss using nuclei:

nuclei -u http://example.com/param?id=1 -tags xss

Step 17: Dalfox: to find XSS

dalfor url [url]

Step 17.1: Crawler

cat host.txt | crawler | tee -a endpoint.txt

cat endpoint.txt | qsreplace ‘“><img src=x onerror=alert(1)> | tee -a xss\_fuzz.txt

cat xss\_fuzz.txt | freq | tee -a possible\_xss.txt