

PTN 9th Anniversary 🎉

30 July 2022, KTM



Automating Low hanging Bugs

By: Veshraj Ghimire

Little about the Presenter:

- Community Leader at Pentester Nepal
- Interested in offensive Security
- Part time bug bounty hunter
- #2 on Bugv All Time Leaderboard as of now
- Life long learner





List of tools to be discussed today

- Nuclei
- NtHiM
- Wpscan
- Gf with GAU
- Dalfox
- reNgine

Nuclei

- By Project Discovery
- Fast and customizable vulnerability scanner based on simple YAML based DSL.
- <https://github.com/projectdiscovery/nuclei>

1. Create Your YAML template

```
id: amazon-mws-secret-token-value
info:
  author: puzzlepeaches
  name: Amazon MWS Secret Token
  severity: medium
requests:
  - method: GET
    path:
      - "{{BaseURL}}"
    extractors:
      type: regex
      part: body
      regex:
        - "amzn\\.mws\\.\\.([0-9a-f]{8})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{4})-([0-9a-f]{12})"
```

2. Run on your targets

```
cat staging-apps.txt
https://staging.example.com
https://staging.admin.example.com
https://staging.cra.example.com
https://api-staging.example.com
https://internal.example.com
https://build-app.example.com
https://demo.example.com
https://preprod.backend-api.example.com

nuclei -t amazon-mws-secret-leak.yaml -l staging-apps.txt

projectdiscovery.io v2.2.0

[WRN] Use with caution. You are responsible for your actions
[WRN] Developers assume no liability and are not responsible for any misuse or damage.
[INF] Loading templates...
[INF] [amazon-mws-secret-leak] Amazon MWS Auth Token leak (@puzzlepeaches) [medium]
[INF] Using 1 rules (1 templates, 0 workflows)
[amazon-mws-secret-leak] [http] [medium] https://internal.example.com
[amazon-mws-secret-leak] [http] [medium] https://build-app.example.com
[amazon-mws-secret-leak] [http] [medium] https://staging.admin.example.com
```


NtHiM

- By late Binit Ghimire
- Now, the Host is Mine! - Super Fast Sub-domain Takeover Detection!
- <https://github.com/TheBinitGhimire/NtHiM>

```
root@WHOSbinit:~# cat hostnames.txt
https://checkingifitexists.github.io
https://whoisbinit.me
https://letscheckthis.github.io
http://letscheckthisone.s3.amazonaws.com
https://thebinitghimire.github.io
http://isitpossibleornot.wordpress.com

root@WHOSbinit:~# NtHiM -f hostnames.txt --threads 50
[Amazon S3] Possible Sub-domain Takeover at http://letscheckthisone.s3.amazonaws.com!
[GitHub Pages] Possible Sub-domain Takeover at https://letscheckthis.github.io!
[GitHub Pages] Possible Sub-domain Takeover at https://checkingifitexists.github.io!
[WordPress.com] Possible Sub-domain Takeover at http://isitpossibleornot.wordpress.com!

root@WHOSbinit:~# |
```

WpScan

- By wpscan Team
- WordPress security scanner. Written for security professionals and blog maintainers to test the security of their WordPress websites.
- <https://github.com/wpscanteam/wpscan>



GAU

- By Corben Leo
- Fetch known URLs from AlienVault's Open Threat Exchange, the Way back Machine, and Common Crawl.
- <https://github.com/lc/gau>

```
root@veshraj: ~  
root@veshraj:~# gau --help  
Usage of gau:  
--blacklist strings  list of extensions to skip  
--fc strings         list of status codes to filter  
--fp               remove different parameters of the same endpoint  
--from string       fetch urls from date (format: YYYYMM)  
--ft strings        list of mime-types to filter  
--json             output as json  
--mc strings        list of status codes to match  
--mt strings        list of mime-types to match  
--o string          filename to write results to  
--providers strings list of providers to use (wayback,commoncrawl,otx,urlscan)  
--proxy string      http proxy to use  
--retries uint      retries for HTTP client  
--subs             include subdomains of target domain  
--threads uint      number of workers to spawn (default 1)  
--timeout uint      timeout (in seconds) for HTTP client (default 45)  
--to string         fetch urls to date (format: YYYYMM)  
--verbose           show verbose output  
--version           show gau version  
root@veshraj:~#
```


GF

- By Tomnomnom
- A wrapper around grep, to help you grep for things
- Patterns based on json
- <https://github.com/tomnomnom/gf>

```
root@veshraj: ~  
root@veshraj:~# gf --help  
Usage of gf:  
-dump      prints the grep command rather than executing it  
-list      list available patterns  
-save      save a pattern (e.g: gf -save pat-name -Hnri 'search-pattern')  
root@veshraj:~#
```

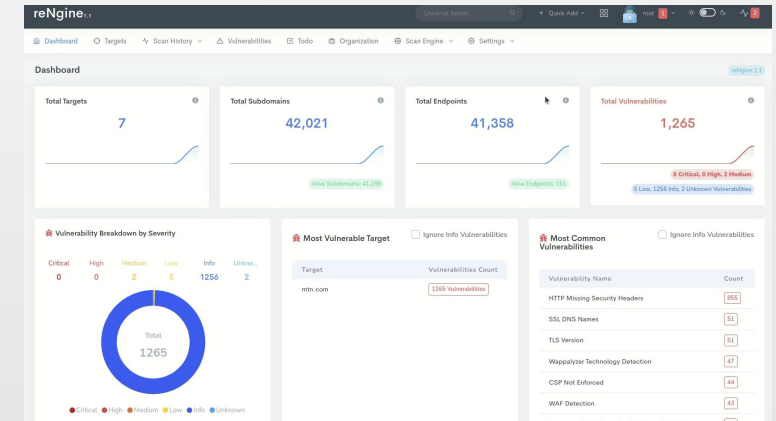
Dalfox

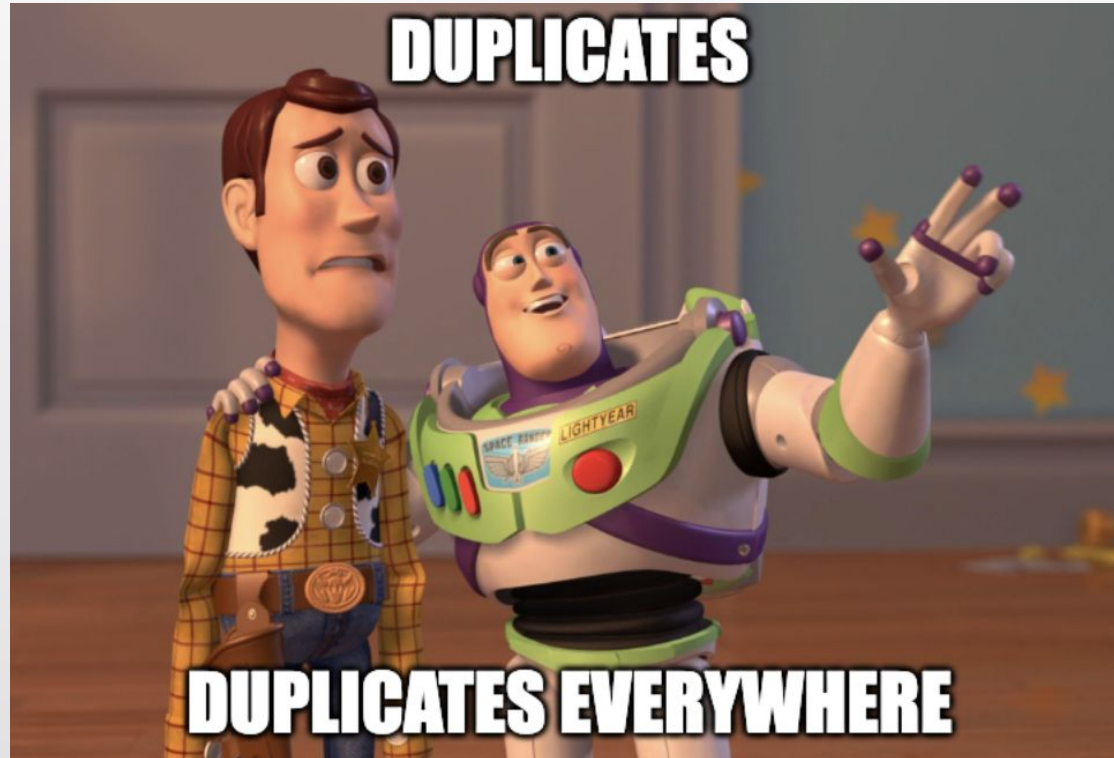
- By Hahwul
- Powerful open source XSS scanning tool and parameter analyzer, utility
- <https://github.com/hahwul/dalfox>

[illegible]

reNginer

- By Yogesh Ojha
- Automated reconnaissance framework for web applications with a focus on highly configurable streamlined recon process via Engines, recon data correlation and organization, continuous monitoring, backed by a database, and simple yet intuitive User Interface.
- <https://github.com/yogeshojha/rengine>







Some tips to avoid duplicates:

- Modify existing/ build Own Templates
- Combine tools
- Build custom wordlist
- Explore full features/ Don't just run on default settings
- Don't just depend upon tools

The End 😊



You can ping me on twitter if you need any help, I am pretty active there:

@GhimireVeshraj