



Introduction OWASP Nettacker

Automated Penetration Testing Framework

Ali Razmjoo Qalaei
ali@offsec.ir
ali.razmjoo@owasp.org

self.author

- Software Engineering Student
- OWASP Chapter and Project Leader
 - Iran Chapter Leader ([link](#))
- OWASP ZSC Project Leader ([link](#))
 - Founder/Creator (9 Contributors) ([link](#))
 - Contributed OWASP ZSC API in GDB Peda ([link](#))
 - Google Summer of Code (GSoC 2016) ([link](#))
 - DEF CON Demo Lab ([link](#))
 - Blackhat Arsenal 2016 ([link](#))
 - Presented in OFFSECONF 2016 ([link](#))
 - Top 10 Best Security Tools by ToolsWatch ([link](#))
 - OWASP Code Sprint 2017 ([link](#))
- Vira: Strategic Development of Security Arenas (PJS)
 - CEO/Founder (viraintel.com)
 - Full Stack Developer
- OWASP Nettacker Project Leader
 - Founder/Creator (5 Contributors) ([link](#))
 - R&D Phase
- Programmer and Security Researcher
 - Researcher
 - Mentor
 - Penetration Tester
 - OFFSEC Leader (Iran) ([link](#))

self.intro

OWASP Nettacker project is created to automate information gathering, vulnerability scanning and eventually generating a report for networks, including services, bugs, vulnerabilities, misconfigurations, and other information. This software will utilize TCP SYN, ACK, ICMP and many other protocols in order to detect and bypass Firewall/IDS/IPS devices. By leveraging a unique method in OWASP Nettacker for discovering protected services and devices such as SCADA. It would make a competitive edge compared to other scanner making it one of the bests.



OWASP

The Open Web Application
Security Project







































self.urls

- OWASP Page: https://www.owasp.org/index.php/OWASP_Nettacker
- Home: <http://nettacker.z3r0d4y.com/>
- Github: <https://github.com/viraintel/OWASP-Nettacker>
- Mailing List: <https://groups.google.com/forum/#!forum/owasp-nettacker>

self.why_owasp_netrunner

- Easy to run and use
 - Run everywhere with Python 2.x, 3.x <https://travis-ci.org/viraintel/OWASP-Netrunner/>

Build Jobs

✓ # 110.1	 </> Python: 2.6	 no environment variables set	 3 min 2 sec	
✓ # 110.2	 </> Python: 2.7	 no environment variables set	 1 min	
✓ # 110.3	 </> Python: 3.3	 no environment variables set	 2 min 29 sec	
✓ # 110.4	 </> Python: 3.4	 no environment variables set	 1 min 21 sec	
✓ # 110.5	 </> Python: 3.5	 no environment variables set	 28 sec	
✓ # 110.6	 </> Python: 3.6	 no environment variables set	 40 sec	
✓ # 110.7	 </> Python: 3.6-dev	 no environment variables set	 1 min 1 sec	
✓ # 110.8	 </> Python: 3.7-dev	 no environment variables set	 2 min 45 sec	
✓ # 110.9	 </> Python: nightly	 no environment variables set	 3 min 19 sec	

self.why_owasp_netstacker

- Easy to use
 - Enabled default options
 - Customizable Configuration
 - Profiles

```
C:\Users\Zombie\Documents\GitHub\OWASP-Nettacker>python nettacker.py -i 127.0.0.1
```


The terminal output shows the following sequence of events:

- Terminal command: `C:\Users\Zombie\Documents\GitHub\OWASP-Nettacker>python nettacker.py -i 127.0.0.1`
- Stylized ASCII art for "OWASP" and "Nettacker".
- Text: `Version 0.0.1` and `SAME`.
- Links: `github.com/viraintel`, `owasp.org`, and `viraintel.com`.
- Status messages:
 - `[+] Nettacker engine started ...`
 - `[+] You are using the last version of OWASP Nettacker ...`
 - `[+] 9 modules loaded ...`
 - `[+] target 127.0.0.1 submitted!` (twice)
 - `[+] start attacking 127.0.0.1, 1 of 6`


self.why_owasp_netstacker

- High Speed
 - Highest speed with fewest hardware resources (CPU: Intel Core i7-3610QM 2.30 GHz)

- 10 threads
 - 0.9% of 1 core of CPU

 python.exe	0.9%	13.6 MB	0 MB/s	0 Mbps
--	------	---------	--------	--------

- 1000 threads
 - 6.4% of 1 core of CPU

 python.exe	6.4%	27.6 MB	0 MB/s	0 Mbps
--	------	---------	--------	--------

- Measure-Command {python netstacker.py -i 192.168.0.1 -t 1000 -T 0.5 -m tcp_connect_port_scan}
 - TotalSeconds : 2.8466034



self.why_owasp_netstacker

- Multi Language Support
 - Supporting 20 language
 - Easy to contribute languages ([JSON](#))

Engine:

Engine input options

```
-L LANGUAGE, --language LANGUAGE
    select a language ['ru', 'fr', 'en', 'nl', 'el', 'vi',
    'id', 'de', 'tr', 'ps', 'ur', 'fa', 'hy', 'hi', 'ko',
    'it', 'zh-cn', 'ar', 'ja', 'es']
-v VERBOSE_LEVEL, --verbose VERBOSE_LEVEL
    verbose mode level (0-5) (default 0)
-V, --version
    show software version
```


self.why_owasp_netacker

- Multi Language Support
 - Supporting 20 language
 - Easy to contribute languages ([JSON](#))

```
!!! بهتر است که از تعداد ریسک کمتر از 100 استفاده کنید، به هر حال ما ادامه می دهیم...  
[+] انجین Netacker شروع به کار کرده...  
  
[+] 9 ماژول بارگذاری شد ...  
[+] هدف 127.0.0.1 ارسال شد!  
[+] هدف 127.0.0.1 ارسال شد!  
[+] شروع حمله به 127.0.0.1، 1 از 1  
[+] ماست: 127.0.0.1 درگاه: 135 پیدا شد!  
[+] ماست: 127.0.0.1 درگاه: 445 پیدا شد!  
[+] ماست: 127.0.0.1 درگاه: 1001 پیدا شد!  
[+] ماست: 127.0.0.1 درگاه: 1080 پیدا شد!  
[+] ماست: 127.0.0.1 درگاه: 65000 پیدا شد!  
[+] در حال بای کردن فایل های موقتی!  
[+] در حال مرتب سازی نتایج!  
[+] در حال ساخت گراف ...  
[+] پایان ساخت گراف!
```


self.why_owasp_netstacker

- Getting focus on IoT Security
 - Show an example? D-Link DWR 932C
 - By Ehsan Nezami and Ali Razmjoo Qalaei –ViralIntel Full Stack Developers
- Mobile Application
 - Telegram BruteForce Passcode with adb Android ([link](#))
 - By Sepehr Keshavarz Haddad –ViralIntel Full Stack Developers

self.owasp_nettacker_categories

- Brute
 - ssh_brute
 - ftp_brute
 - smtp_brute

- Scan
 - dir_scan
 - tcp_connect_port_scan
 - viewdns_reverse_ip_lookup_scan

- Graph
 - [d3_tree_v1_graph](#) (closed) – D3 JS Lib
 - [d3_tree_v2_graph](#) (opened) – D3 JS Lib
 - [jit_circle_v1_graph](#) - JIT JS Lib

self.owasp_nettacker_categories

OWASP Nettacker Categories are not limited, we already working on WiFi Networks, Exploiting and Payloads, Fuzzers, AI and Machine Learning, and other categories. (Publish Soon)

- WiFi
 - disconnect_all_clients_wifi
 - wps_vulnerability_wifi
 - wpa2_attack_wifi
- Scan
 - device_scan
 - os_scan
 - router_scan
 - tcp_udp_signature_scan
- Fuzzer
 - web_application_fuzz
 - binary_argv_fuzz
- Exploit
 - heartbleed_exploit
 - eternal_blue_exploit
 - microsoft_office_hta_exploit
 - dlink_dir605L_dos_exploit
- Payload ([OWASP ZSC](#))
 - windows_shellcode_x86_payload
 - windows_shellcode_x64_payload
 - osx_shellcode_x86_payload
 - osx_shellcode_x64_payload
 - linux_shellcode_x86_payload
 - linux_shellcode_x64_payload
 - windows_x86_egg_hunter_payload
 - linux_x86_egg_hunter_payload

self.owasp_nettacker_features

- Smart and Automated
 - Won't ask any question while scanning
- HTML& TEXT Report
 - Graphs (Network Map)
 - Multi Language
- Attack IP Range
 - Use RIPE database
- Find Subdomains
 - Use sublister
- Customize attacks
 - Each module with different output (defaults are available)
 - Define timeout
 - Define thread number for a host
 - Define thread for scan hosts
 - Define retries
 - Define ping before scan
 - Define time to sleep (delay)
 - Define proxy
 - Define language
- Available Soon
 - Use tor
 - Local API
 - GUI, Web UI
 - Online API
 - Google Chrome and Mozilla Firefox plugin/addons
 - API Fuzzer
 - Documents with video tutorials

self.usage

Documents (Wiki Page): <https://github.com/viraintel/OWASP-Nettacker/wiki>

```

OWASP-Nettacker master
save some bytes for f**k's sake!
A1 Razmjoo · c14002e · 3 changed files
Version 0.0.1
SAME
core/compatible.py
core/parse.py

github.com/viraintel/OWASP-Nettacker
owasp.org
viraintel.com
bug fixed in d3_tree graph
18 hours ago by A1 Razmjoo

update readme, features

usage: Nettacker [-L LANGUAGE] [-v VERBOSE_LEVEL] [-V] [-c] [-o LOG_IN_FILE]
               [--graph GRAPH_FLAG] [-h] [-i TARGETS] [-I TARGETS_LIST]
               [-m SCAN_METHOD] [-x EXCLUDE_METHOD] [-u USERS]
               [-U USERS_LIST] [-p PASSWDS] [-P PASSWDS_LIST] [-g PORTS]
               [-T TIMEOUT_SEC] [-w TIME_SLEEP] [-r] [-s] [-t THREAD_NUMBER]
               [-M THREAD_NUMBER_HOST] [-R PROXIES]
               [--proxy-list PROXIES_FILE] [--retries RETRIES]
               [--ping-before-scan] [--method-args METHODS_ARGS]
               [--method-args-list]
    
```


self.developers_status



self.how_to_develop

If anyone interested to contribute OWASP Nettacker Project, we gladly support and appreciate that!

Developers Wiki Page: <https://github.com/viraintel/OWASP-Nettacker/wiki/Developers>

QUESTION?

Contact

<https://github.com/viraintel/OWASP-Nettacker/issues>

ali.@offsec.ir

ali.Razmjoo@owasp.org

OFFSEC
w w w . o f f s e c . i r

THE END – DROP THE MIC



OFFSEC
w w w . o f f s e c . i r