How to Get Your Name on a CVE

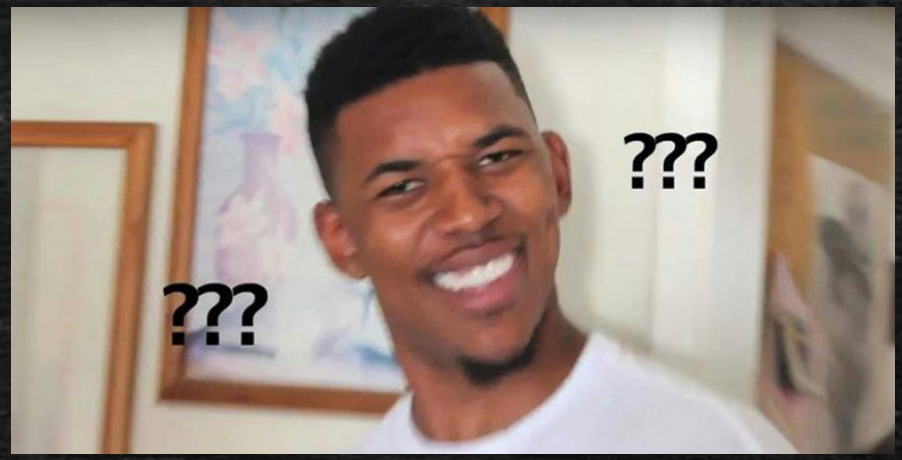
And Make an Impact!

whoami

- Alias @m0n1x90
- Founder @TamilCTF
- Security Engineer @Zoho
- Red Teaming / Application Security / Malware Dev
- Part time full stack development
- Blog at m0n1x90.dev

Confused ..?

- Vulnerability
- Exploit
- CVE
- 0-Day
- N-Day



Why Get Your Name on a CVE?

- Recognition, Reputation
- Contribution, Improving security community
- Career Benefits, Networking and opportunities
- Educational Value, Research learning curve

How Vulnerabilities are Discovered

- Code Review & SAST
- Fuzzing & DAST
- Exploit Development & Security Research
- Penetration Testing & Vendor Disclosure Programs

SAST Approach

- SAST – Static Application Security Testing
- White box approach
- Analyzes code without executing it
- Searches for patterns
- May generate false positives, requiring manual verification
- Easy integration
- Snyk, Semgrep, SonarQube, Checkmarx etc.

Demo - Code Review & SAST

Easy way for finding bugs from source code

Vulnerable Code

File: **uaf.c**

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main() {
    char *data = (char *)malloc(64);
    strcpy(data, "Sensitive data");
    free(data);
    printf("Data after free: %s\n", data);  // UAF
    return 0;
}
```

SAST Scan


1 Code Finding
uaf.c
›› use-after-free
Possible use-after-free detected: accessing memory after it has been freed.
6 char *data = (char *)malloc(64);
7 strcpy(data, "Sensitive data");
8 free(data);
9 printf("Data after free: %s\n", data);
// UAF

Vulnerable Pattern

File: **uaf-detection.yaml**

```yaml
rules:
  - id: use-after-free
    patterns:
      - pattern: |
          $DATA = (char *)malloc($SIZE);
          ...
          free($DATA);
          ...
          printf(..., $DATA);
      - pattern: |
          $DATA = (char *)malloc($SIZE);
          ...
          free($DATA);
          ...
          $USE_AFTER_FREE;
```

DAST Approach

- DAST – Dynamic Application Security Testing
- Black/Gray box approach
- Analyzes in runtime environment
- Simulates attacks and validates the behavior
- May generate false positives, requires manual verification
- Web - BurpSuite, OWASP ZAP, Nessus, SQLMap etc.
- Program Fuzzers – AFL , LibFuzzer, ClusterFuzz etc.

Demo – DAST (Web)

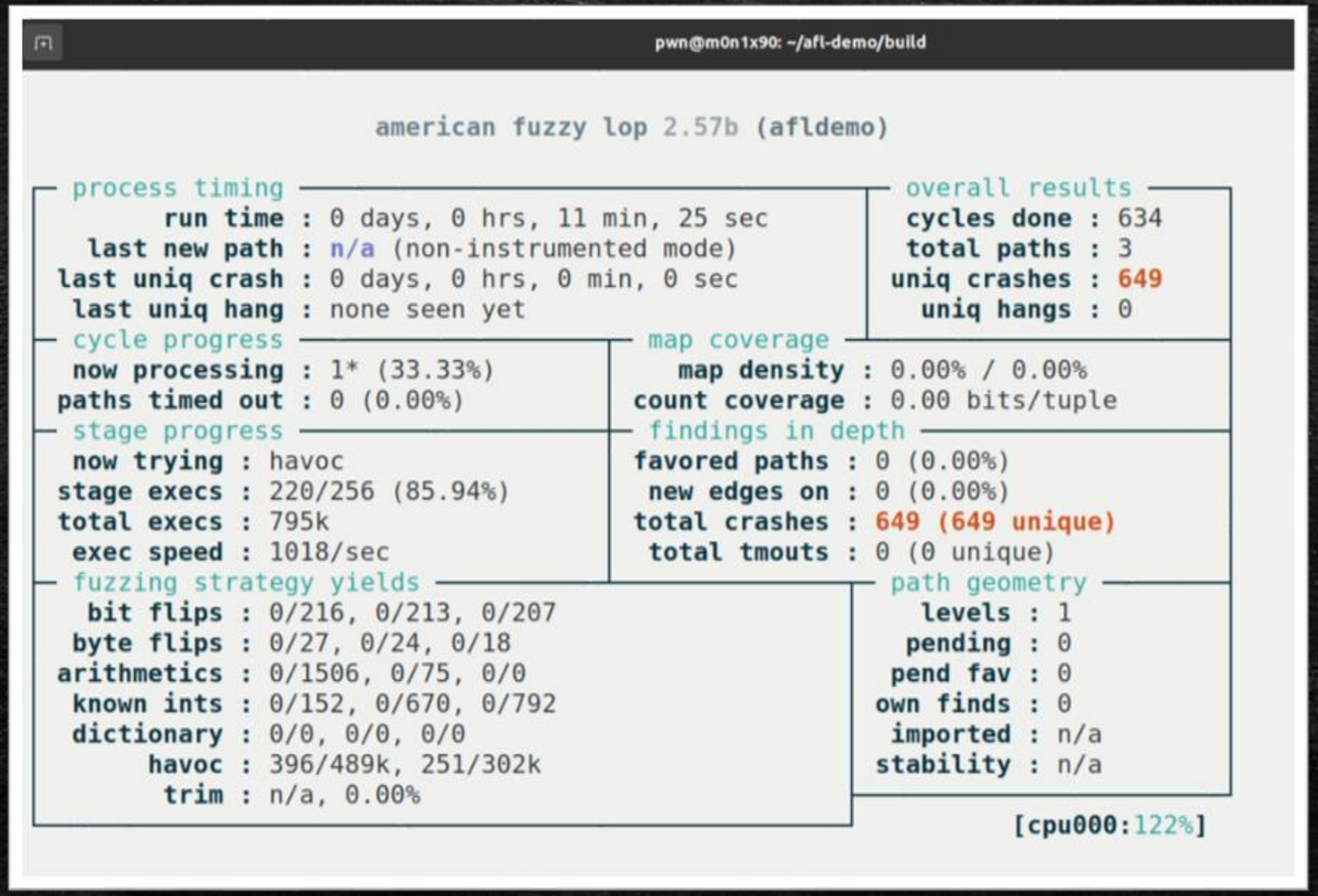
Finding bugs from fuzzing web application

```
        ___ ___[']_____ ___ ___  {1.6.4#stable}
 ___ ___[.]_____ ___ ___  {1.6.4#stable}
|_ -| . [.]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 15:56:48 /2025-03-14/

[15:56:48] [INFO] parsing HTTP request from 'req.txt'
[15:56:48] [INFO] testing connection to the target URL
[15:56:48] [INFO] testing if the target URL content is stable
[15:56:48] [INFO] target URL content is stable
[15:56:48] [INFO] testing if POST parameter 'username' is dynamic
[15:56:48] [WARNING] POST parameter 'username' does not appear to be dynamic
[15:56:48] [WARNING] heuristic (basic) test shows that POST parameter 'username' might not be injectable
[15:56:48] [INFO] testing for SQL injection on POST parameter 'username'
[15:56:49] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:56:49] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[15:56:49] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[15:56:49] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[15:56:49] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[15:56:49] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[15:56:49] [INFO] testing 'Generic inline queries'
[15:56:49] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[15:56:49] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[15:56:49] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[15:56:49] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[15:56:59] [INFO] POST parameter 'username' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y
[15:57:05] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[15:57:05] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
```

```
[15:56:49] [INFO] testing 'Generic inline queries'
[15:56:49] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[15:56:49] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[15:56:49] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[15:56:49] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[15:56:59] [INFO] POST parameter 'username' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y
[15:57:05] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[15:57:05] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[15:57:05] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extend
ng the range for current UNION query injection technique test
[15:57:05] [INFO] target URL appears to have 3 columns in query
got a 302 redirect to 'http://127.0.0.1:5000/menu'. Do you want to follow? [Y/n] Y
redirect is a result of a POST request. Do you want to resend original POST data to a new location? [y/N] Y
do you want to (re)try to find proper UNION column types with fuzzy test? [y/N] y
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y
[15:57:15] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')
[15:57:15] [INFO] target URL appears to be UNION injectable with 3 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y
[15:57:17] [INFO] checking if the injection point on POST parameter 'username' is a false positive
POST parameter 'username' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 130 HTTP(s) requests:
---
Parameter: username (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: username=test' AND (SELECT 8493 FROM (SELECT(SLEEP(5)))EqPf) AND 'ftzu'='ftzu&password=test
---
[15:57:41] [INFO] the back-end DBMS is MySQL
[15:57:41] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
back-end DBMS: MySQL >= 5.0.12
[15:57:41] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 48 times
```

Demo – DAST (Programs)

Finding bugs from fuzzing executable programs


pwn@m0n1x90: ~/afl-demo/build
american fuzzy lop 2.57b (afldemo)
process timing
run time : 0 days, 0 hrs, 11 min, 25 sec
last new path : n/a (non-instrumented mode)
last uniq crash : 0 days, 0 hrs, 0 min, 0 sec
last uniq hang : none seen yet
overall results
cycles done : 634
total paths : 3
uniq crashes : 649
uniq hangs : 0
cycle progress
now processing : 1* (33.33%)
paths timed out : 0 (0.00%)
map coverage
map density : 0.00% / 0.00%
count coverage : 0.00 bits/tuple
stage progress
now trying : havoc
stage execs : 220/256 (85.94%)
total execs : 795k
exec speed : 1018/sec
findings in depth
favored paths : 0 (0.00%)
new edges on : 0 (0.00%)
total crashes : 649 (649 unique)
total tmouts : 0 (0 unique)
fuzzing strategy yields
bit flips : 0/216, 0/213, 0/207
byte flips : 0/27, 0/24, 0/18
arithmetics : 0/1506, 0/75, 0/0
known ints : 0/152, 0/670, 0/792
dictionary : 0/0, 0/0, 0/0
havoc : 396/489k, 251/302k
trim : n/a, 0.00%
path geometry
levels : 1
pending : 0
pend fav : 0
own finds : 0
imported : n/a
stability : n/a
[cpu000:122%]

```
pwn@m0n1x90:~/afl-demo/build$
pwn@m0n1x90:~/afl-demo/build$ ./afldemo <<<$(echo Hello)
Hello
pwn@m0n1x90:~/afl-demo/build$ ./afldemo <<<$(echo Fuzzing)
Fuzzing
pwn@m0n1x90:~/afl-demo/build$ ./afldemo <<<$(echo Demo)
Demo
pwn@m0n1x90:~/afl-demo/build$ ./afldemo <<<$(cat ../findings/crashes/id
:000000,sig:11,src:000001+000002,op:splice,rep:64)
bash: warning: command substitution: ignored null byte in input
Segmentation fault
pwn@m0n1x90:~/afl-demo/build$ ./afldemo <<<$(cat ../findings/crashes/id
:000003,sig:11,src:000002,op:havoc,rep:8)
bash: warning: command substitution: ignored null byte in input
Segmentation fault
pwn@m0n1x90:~/afl-demo/build$ ./afldemo <<<$(cat ../findings/crashes/id
:000021,sig:11,src:000002+000001,op:splice,rep:128)
bash: warning: command substitution: ignored null byte in input
Segmentation fault
pwn@m0n1x90:~/afl-demo/build$ dmesg | tail
[ 1040.138749] Code: b6 00 3c 2b 75 11 48 8b 45 f8 48 8d 50 01 48 89 55
 f8 c6 00 20 eb 15 48 8b 45 f8 48 8d 50 01 48 89 55 f8 48 8b 55 e8 0f b
6 12 <88> 10 48 83 45 e8 01 e9 42 ff ff ff 48 8d 05 7b 2d 00 00 5d c3 f
3
[ 1040.428895] afldemo[866246]: segfault at 563b9bab8000 ip 0000563b9ba
b4512 sp 00007ffd89e42200 error 6 in afldemo[563b9bab4000+1000]
[ 1040.428907] Code: b6 00 3c 2b 75 11 48 8b 45 f8 48 8d 50 01 48 89 55
 f8 c6 00 20 eb 15 48 8b 45 f8 48 8d 50 01 48 89 55 f8 48 8b 55 e8 0f b
6 12 <88> 10 48 83 45 e8 01 e9 42 ff ff ff 48 8d 05 7b 2d 00 00 5d c3 f
3
[ 1140.491151] show_signal_msg: 17 callbacks suppressed
[ 1140.491154] afldemo[868861]: segfault at 55eafddfa000 ip 000055eafdd
f6512 sp 00007fff28ffcc80 error 6 in afldemo[55eafddf6000+1000]
[ 1140.491161] Code: b6 00 3c 2b 75 11 48 8b 45 f8 48 8d 50 01 48 89 55
 f8 c6 00 20 eb 15 48 8b 45 f8 48 8d 50 01 48 89 55 f8 48 8b 55 e8 0f b
```

```
pwn@m0n1x90:~/afl-demo/findings$ ls
crashes  fuzz_bitmap  fuzzer_stats  hangs  plot_data  queue
pwn@m0n1x90:~/afl-demo/findings$ cd crashes/
pwn@m0n1x90:~/afl-demo/findings/crashes$ ls
id:000000,sig:11,src:000001+000002,op:splice,rep:64
id:000001,sig:11,src:000001+000002,op:splice,rep:32
id:000002,sig:11,src:000002,op:havoc,rep:128
id:000003,sig:11,src:000002,op:havoc,rep:8
id:000004,sig:11,src:000002+000001,op:splice,rep:128
id:000005,sig:11,src:000002+000001,op:splice,rep:16
id:000006,sig:11,src:000000,op:havoc,rep:32
id:000007,sig:11,src:000001,op:havoc,rep:16
id:000008,sig:11,src:000001+000002,op:splice,rep:64
id:000009,sig:11,src:000002,op:havoc,rep:64
id:000010,sig:11,src:000002,op:havoc,rep:32
id:000011,sig:11,src:000002+000001,op:splice,rep:64
id:000012,sig:11,src:000002+000001,op:splice,rep:32
id:000013,sig:11,src:000002+000001,op:splice,rep:64
id:000014,sig:11,src:000000,op:havoc,rep:64
id:000015,sig:11,src:000000,op:havoc,rep:64
id:000016,sig:11,src:000001,op:havoc,rep:32
id:000017,sig:11,src:000001,op:havoc,rep:64
id:000018,sig:11,src:000001,op:havoc,rep:32
id:000019,sig:11,src:000001+000002,op:splice,rep:32
id:000020,sig:11,src:000002+000001,op:splice,rep:64
id:000021,sig:11,src:000002+000001,op:splice,rep:128
id:000022,sig:11,src:000002+000001,op:splice,rep:32
id:000023,sig:11,src:000002+000001,op:splice,rep:32
id:000024,sig:11,src:000000,op:havoc,rep:128
id:000025,sig:11,src:000001,op:havoc,rep:128
id:000026,sig:11,src:000001,op:havoc,rep:128
id:000027,sig:11,src:000001+000002,op:splice,rep:64
id:000028,sig:11,src:000001+000002,op:splice,rep:128
id:000029,sig:11,src:000001+000002,op:splice,rep:128
id:000030,sig:11,src:000002,op:havoc,rep:64
```

Exploit Development & Research

- Reverse Engineering Skill
- Better Code Understanding, Better Results
- Unexplored Area = Unknown Attack Surfaces
- Less Competition
- Requires more effort and time
- Unique CVEs, More Recognition, More Rewards $$$$

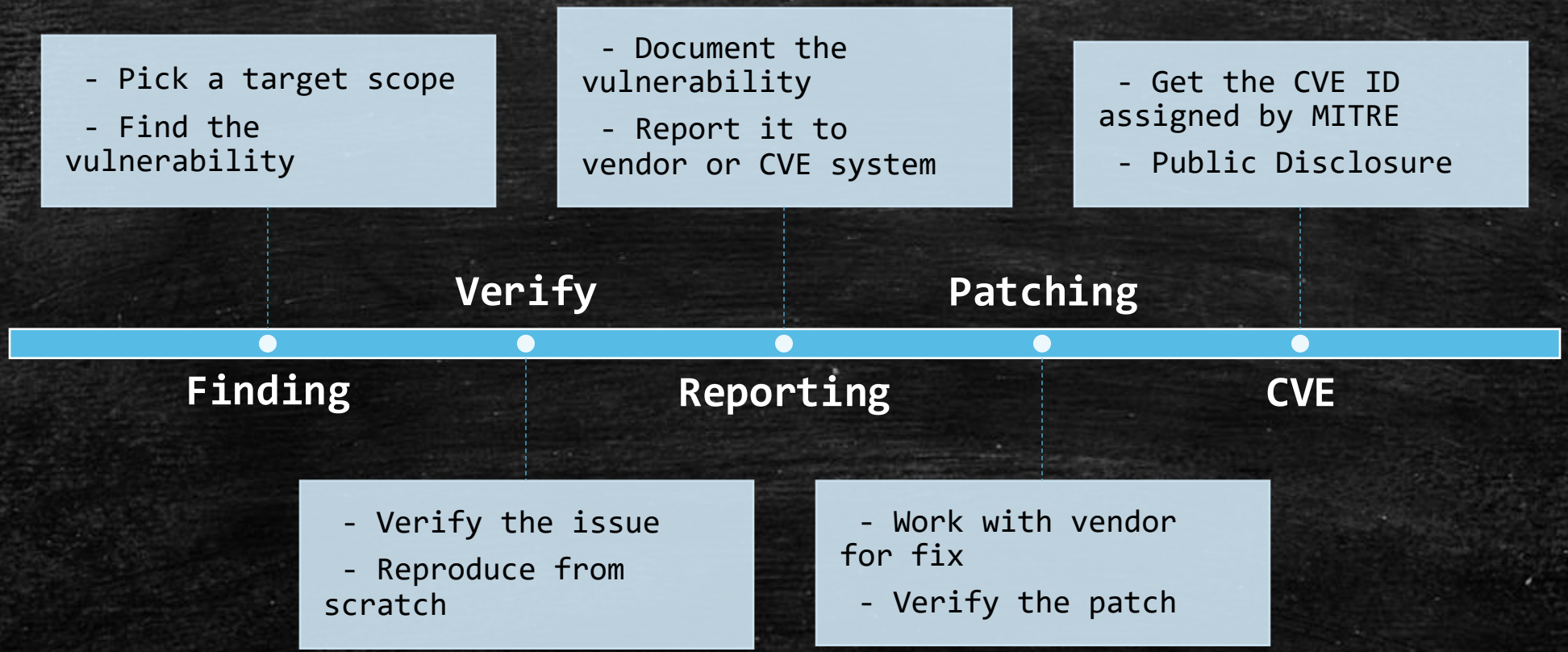
Penetration Testing & VDPs

- Hybrid SAST + DAST Approach
- Application Security / Bug Bounty Hunting
- Authorized attack simulation on product/enterprise
- More competition
- Rewards may vary based on the scope
- Possibility to find bypass for existing CVEs

Choose your approach !

- Open Source Projects/Frameworks
- CMS Bundles – WordPress, Joomla, Drupal, etc.
- Targets with Less Audience
- Targets with New Features
- Explore and study targets without documentation
- Having workflow other than traditional way
- Smart automation comes in handy

Steps to Get Your Name on a CVE



" Creative Offense always prevails over Traditional Defense "

Good Luck 😉

Follow Me On
@m0n1x90