Chaotic Rat

Snehal Sonvane, Chidambaram Senthil
GUIDE: Dr. Brad Reaves
North Carolina State University, Raleigh

Introduction

Resiliency is becoming an important factor for various computer systems and networks. Negligence of resiliency is a problem which should be solved in order to maintain the reliability of a system. In this paper, we propose design guidelines to create a tool which can make a system more resilient by breaking it more often. The tool 'Chaotic Rat' induces random failures within the system, which provides us a way to understand the effect of failures in it.

In our attempt to provide theoretical metric for Resilient system, we propose that a system is resilient if,

1. Ensures confidentiality by limiting access to information
2. Ensures the information is true, accurate and trustworthy by maintaining Integrity at all times.
3. Guarantees of the availability of the system/service when required
4. Retaining performance required to achieve needs of the system
5. Exhibiting fault tolerance with respect to above attributes in face of unexpected alterations, attacks or even increased load

Design

What should be the nature of failures induced?

The main goal of the tool is to systematically inject failures in the system. The failures programmed in the tool should help in improving the quality of the application in terms of security and serviceability as expected of it. In addition, there should be a thorough analysis on what scale should these failures will affect the infrastructure.

When should the failures be induced?

The rat should be allowed to break things after any alterations in the code, after patch or fix implementation on a build, if some configurations are changed and/or after additional functionalities are added to the application. This allows to catch the bugs or small vulnerabilities in the system at an earlier stage, which might be a big threat in future, at the earliest stage. The nature of failure can be either random or deterministic in nature depending on the control the engineer expects to have.

How should the failures be analyzed?

In case, the system fails to stand in the face of the imparted problems, the engineer should analyze the reason and think of the effective measures that can be taken to avoid such mishaps in the future. However, the process should not stop here but the implementation of the failure should be advanced to cover any scenarios not considered previously

Methodology

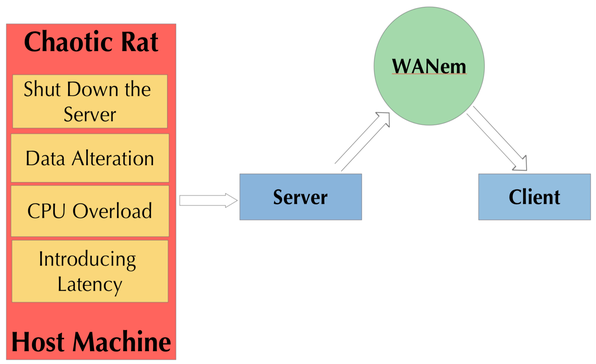


Fig. 1: The Chaotic Rat running on the management machine creates the above mentioned breaches in the server

Results

We successfully made Chaotic Rat to break the primary server in The following ways and which in turn ideally redirects the Incoming requests to another backup server

A. Rat Shuts down the server:

The Rat successfully shuts down the server making it unavailable to the client.

B. Rat Alters data in the server:

```
mysql> select * from employees;
+------+---------+------+------+--------+
| ID   | name    | age  | sex  | salary |
+------+---------+------+------+--------+
| 1000 | rob     |   42 | M    | 275000 |
| 2000 | gareth  |   28 | M    | 325000 |
| 3000 | ronaldo |   32 | M    | 575000 |
| 4000 | anthony |   24 | M    | 175000 |
| 5000 | tony    |   27 | M    |  75000 |
| 6000 | lucas   |   26 | M    | 205000 |
| 7000 | sergiio |   30 | M    | 215000 |
| 8000 | serena  |   29 | F    | 200000 |
| 9000 | roger   |   32 | M    | 300000 |
+------+---------+------+------+--------+
9 rows in set (0.00 sec)
```

Fig. 2: Initial data in the server

```
<terminated> dbconnect_new [Ja
Tue Nov 28 02:03:00 EST
Row is updated by RAT.
```

```
mysql> mysql> select * from employees;
+------+---------+------+------+--------+
| ID   | name    | age  | sex  | salary |
+------+---------+------+------+--------+
| 1000 | rob     |   42 | M    |      0 |
| 2000 | gareth  |   28 | M    |      0 |
| 3000 | ronaldo |   32 | M    |      0 |
| 4000 | anthony |   24 | M    |      0 |
| 5000 | tony    |   27 | M    |      0 |
| 6000 | lucas   |   26 | M    |      0 |
| 7000 | sergiio |   30 | M    |      0 |
| 8000 | serena  |   29 | F    |      0 |
| 9000 | roger   |   32 | M    |      0 |
+------+---------+------+------+--------+
9 rows in set (0.00 sec)
```

Fig. 3: Data altered by 'Chaotic Rat'

C. Rat Overloads the CPU:

```
1  [|||||||||                 24.5%]   Tasks: 367, 744 thr; 1 running
2  [||                         3.3%]   Load average: 1.48 1.36 1.34
3  [||||||                    17.8%]   Uptime: 1 day, 12:34:36
4  [||                         2.6%]
Mem[|||||||||||||||||| 7.41G/16.0G]
Swp[|||||||            420M/2.00G]
```

Fig. 4: Initial CPU performance

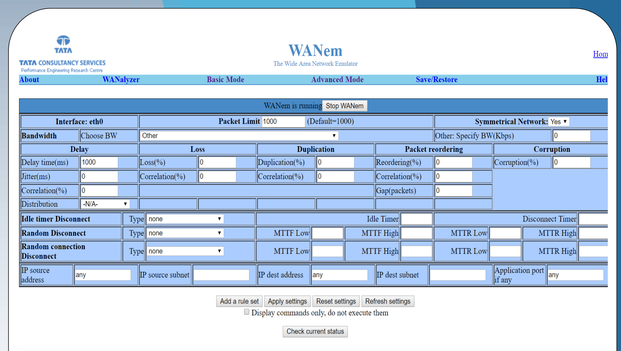
```
1  [||||||||||||||||||||||||98.7%]   Tasks: 369, 759 thr; 17 running
2  [||||||||||||||||||||||||98.7%]   Load average: 10.29 3.84 2.26
3  [||||||||||||||||||||||||98.7%]   Uptime: 1 day, 12:35:53
4  [||||||||||||||||||||||||98.7%]
Mem[|||||||||||||||| 7.43G/16.0G]
Swp[|||||||            420M/2.00G]
```

Fig. 5: CPU Overloaded by Chaotic Rat by creating random unwanted process threads

D. Rat Introduces Latency:

```
client@client-VirtualBox:~$ ping 192.168.0.97
PING 192.168.0.97 (192.168.0.97) 56(84) bytes of data.
64 bytes from 192.168.0.97: icmp_seq=1 ttl=64 time=1.49 ms
64 bytes from 192.168.0.97: icmp_seq=2 ttl=64 time=0.453 ms
64 bytes from 192.168.0.97: icmp_seq=3 ttl=64 time=0.645 ms
64 bytes from 192.168.0.97: icmp_seq=4 ttl=64 time=0.933 ms
64 bytes from 192.168.0.97: icmp_seq=5 ttl=64 time=0.456 ms
64 bytes from 192.168.0.97: icmp_seq=6 ttl=64 time=0.554 ms
64 bytes from 192.168.0.97: icmp_seq=7 ttl=64 time=0.697 ms
64 bytes from 192.168.0.97: icmp_seq=8 ttl=64 time=0.351 ms
64 bytes from 192.168.0.97: icmp_seq=9 ttl=64 time=0.445 ms
64 bytes from 192.168.0.97: icmp_seq=10 ttl=64 time=0.382 ms
64 bytes from 192.168.0.97: icmp_seq=11 ttl=64 time=0.477 ms
64 bytes from 192.168.0.97: icmp_seq=12 ttl=64 time=0.670 ms
64 bytes from 192.168.0.97: icmp_seq=13 ttl=64 time=0.684 ms
64 bytes from 192.168.0.97: icmp_seq=14 ttl=64 time=0.380 ms
64 bytes from 192.168.0.97: icmp_seq=15 ttl=64 time=0.451 ms
```

Fig. 6: Initial Round Trip Time required for pings from client to server




```
Interface: eth0
Delay: 1000ms
WANem packet limit: 1000 packets
Bytes sent: 34169 Packets sent: 98 Packets dropped: 0,
```

Fig. 7: Configuration of WANem Machine by Chaotic Rat to introduce Latency

```
client@client-VirtualBox:~$ ping 192.168.0.97
PING 192.168.0.97 (192.168.0.97) 56(84) bytes of data.
64 bytes from 192.168.0.97: icmp_seq=1 ttl=63 time=4018 ms
64 bytes from 192.168.0.97: icmp_seq=2 ttl=63 time=3006 ms
64 bytes from 192.168.0.97: icmp_seq=3 ttl=63 time=2006 ms
64 bytes from 192.168.0.97: icmp_seq=4 ttl=63 time=2005 ms
64 bytes from 192.168.0.97: icmp_seq=5 ttl=63 time=2001 ms
64 bytes from 192.168.0.97: icmp_seq=6 ttl=63 time=2002 ms
64 bytes from 192.168.0.97: icmp_seq=7 ttl=63 time=2004 ms
64 bytes from 192.168.0.97: icmp_seq=8 ttl=63 time=2006 ms
64 bytes from 192.168.0.97: icmp_seq=9 ttl=63 time=2006 ms
64 bytes from 192.168.0.97: icmp_seq=10 ttl=63 time=2005 ms
64 bytes from 192.168.0.97: icmp_seq=11 ttl=63 time=2006 ms
64 bytes from 192.168.0.97: icmp_seq=12 ttl=63 time=2008 ms
64 bytes from 192.168.0.97: icmp_seq=13 ttl=63 time=2007 ms
64 bytes from 192.168.0.97: icmp_seq=14 ttl=63 time=2003 ms
64 bytes from 192.168.0.97: icmp_seq=15 ttl=63 time=2003 ms
^C
--- 192.168.0.97 ping statistics ---
```

Fig. 8: Round Trip Time of Pings from Client to Server after Latency is introduced by Rat

Conclusion

From the above results, we have been successful in implementing the tool 'Chaotic Rat' which induces failures in the server such as Shutting down the server, alteration of data in the server, overloading the server machine by creating new processes, introducing latency between the client and server machines. These failures help us to analyze the behavior of the system in the face of intentional or unintentional failures. This in turn will point us towards the cracks and vulnerabilities present in the system. Thus helping us to work towards making the system more resilient.

"The best way to avoid failures is to fail the system constantly" --Netflix