

# Linux/Moose: Still breathing

By Editor

Archived: 2026-04-05 22:05:31 UTC

ESET Research

For the past year, ESET and the security firm GoSecure combined their skills in order to research Linux/Moose further. Here's some of what was uncovered.

02 Nov 2016 • , 5 min. read

## What is a Moose - Introduction

Linux/Moose is a malware family that primarily targets Linux-based consumer routers but that can also infect other Linux-based embedded systems in its path. The compromised devices are used to steal unencrypted network traffic and offer proxying services to the botnet operator. In practice, these capabilities are used to steal HTTP Cookies on popular social network sites and perform fraudulent actions such as non-legitimate "follows", "views" and "likes".

In May 2015 ESET released a [whitepaper](#) on the malware family we named Linux/Moose. After publication, Linux/Moose's command and control servers went down and we lost track of the animal. [A few months later](#), in September 2015, we got a new sample of Linux/Moose —with, as expected, some evolution after our publication.

For the past year, ESET and the security firm GoSecure combined their skills in order to research Linux/Moose further. GoSecure investigated the social media fraud aspect and shed some light on an unknown market they called "The Ego Market". This market is highlighted in a [new whitepaper published by GoSecure](#). This blog will cover the technical changes between the Moose variants we described in our whitepaper and the new variants that appeared in September 2015.

## Moose in the bushes - Hiding the address of C&C

The first thing we noticed when we got the new sample was that there was no more command and control (C&C) IP address inside the binary. It seems that the operators read our report carefully and decided to make things a little bit harder for us. In this new version the C&C IP address is given as an encrypted command line argument, as shown in the following output:

```
[...]  
#echo -n -e "\x00\x00\xe6\x13\x02\x00...[REDACTED]" >> /tmp/crondd  
#chmod +x /tmp/crondd  
#/tmp/crondd 763473758  
Loading modules...  
Modules are loaded
```

This new feature implies that we can no longer run the sample by ourselves; our test machines need to be compromised by an embedded device spreading the threat in-the-wild in order to retrieve the C&C IP address. The attentive reader will notice that the IP address shown is in 32-bit integer format.

The purpose of encrypting the IP address here is, if the binary is found alone, useless without the value passed as argument. Also, the value alone makes no sense without the binary having the correct value to decrypt the argument. The value is XORed with a static value as shown in the following code:

```
[...]  
if ( !cnc_ip )  
    return 0;  
cnc_ip ^= 0xF789AC9E;  
[...]  
DECOMPILER OUTPUT
```

Decompiler output

To the best of our knowledge, this value has stayed the same over the last few months. Here is a Python snippet to decrypt the C&C 32-bit Integer value:

```
import socket  
import struct  
argument = 763473758  
static_value = 0xF789AC9E  
decrypted_argument = argument ^ static_value  
print(socket.inet_ntoa(struct.pack("<I", decrypted_argument)))
```

Decrypt C&C IP address

## Moose molted - Network communication

The network protocol changed but it kept the basis of its protocol and added new layers. Here is a quick look of a packet capture from both samples. By the look of things the main change here is from binary protocol to ASCII printable protocol. In Figure 1, on the left side there is the old network protocol and on the right side there is the new one.

00000000	1c 00 00 00 01 00 00 00 01 00 00 00 01 00 00 00	.....	00000000	47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 0d 0a	GET / HT TP/1.1..
00000010	00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00	.....	00000010	48 6f 73 74 3a 20 77 77 77 2e 63 68 61 6c 6c 70	Host: ww w.cha1lp
00000020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000020	6f 6b 2e 63 6e 0d 0a 55 73 65 72 2d 41 67 65 6e	ok.cn..U ser-Agen
00000030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000030	74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28	t: Mozil la/5.0 (
00000040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000040	57 69 6e 64 6f 77 73 20 4e 54 20 36 2e 32 3b 20	Windows NT 6.2;
00000050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000050	72 76 3a 34 30 2e 30 29 20 47 65 63 6b 6f 2f 32	rv:40.0) Gecko/2
00000060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000060	30 31 30 30 31 30 31 20 46 69 72 65 66 6f 78 2f	010010) Firefox/
00000070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000070	34 30 2e 30 0d 0a 41 63 63 65 70 74 3a 20 2a 2f	40.0..Ac cept: */
00000080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000080	2a 0d 0a 41 63 63 65 70 74 2d 4c 61 6e 67 75 61	*..Accep t-Langua
00000090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000090	2e 35 0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64	ge: en-U S,en;q=0
00000100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000100	69 6e 67 3a 20 67 7a 69 70 2c 20 64 65 66 6c 61	..S.Acce pt-Encod
00000110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000110	74 65 0d 0a 43 6f 6f 6b 69 65 3a 20 50 48 50 53	ing: gz1 p, defla
00000120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000120	45 53 53 49 44 3d 42 43 41 41 41 41 42 41 42 41	te..Cook ie: PHP5
00000130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000130	41 41 41 41 46 42 47 41 44 42 41 41 45 47 45 47	ESSID=BC AAAABABA
00000140	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000140	41 41 41 41 4d 4d 4d 4d 41 41 41 41 41 41 41 41	AAAAFBGA DBAAEGEG
00000150	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000150	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAA AAAAAAAA
00000160	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000160	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAA AAAAAEAE
00000170	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000170	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAA AAAAAAAA
00000180	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000180	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAA AAAAAAAA
00000190	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000190	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAA AAAAAAAA
00000200	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000200	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAA AAAAAAAA
00000210	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000210	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAA AAAAAAAA
00000220	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	00000220	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAA AAAAAAAA

Figure 1: Network Protocol Differences

In the old sample the configuration was sent by the C&C server to the malware, and contains various fields like bits field to determine what feature to enable, IP address fields, whitelist list field and password list field. These fields are still present in the new version, but separated in three keys (see Table 1). The operator uses the Cookie: and Set-Cookie: [HTTP headers](#) to send these config fields. This config is encrypted by a simple XOR loop like in the first version but it's also encoded in order to be printable in the HTTP headers thanks to the following function:

```
char * __fastcall to_printable(char *result, char *a2, int len)
{
    unsigned __int8 v3; // ST13_1@2
    int i; // [sp+14h] [bp-8h]@1

    for ( i = 0; i < len; ++i )
    {
        v3 = a2[i];
        result[2 * i] = (v3 & 0xF) + 'A';
        result[2 * i + 1] = (v3 >> 4) + 'A';
    }
    return result;
}
```

Decompiler output

Table 1 (key-value table) summarizes the main configuration of Moose v.2:

key	value
PHPSESSID	main config (local/external scan, sniffer, kill process)
LP	password list
WL	whitelist list

The PHPSESSID key contains the encrypted value of bit fields that allow to enable or disable some features on Moose (local/external scan, sniffer, kill process). The LP key contains the password list. Linux/Moose still tries to spread itself by bruteforcing Telnet credentials. There was a big change in this list, from around 300 logins and passwords in 2015 versus around 10 in 2016. See below:

```
support support
admin admin
root root
guest
admin smcadmin
root
admin
adm
1234 1234
root 12345
admin 1234
```

The WL key contains the whitelist list. Again the list was shorted from 50 to 10 IP addresses. These IP addresses are in the IoC section. Linux/Moose still has the ability to run a proxy service by listening on TCP port 20012. The previous variant used to listen on port 10073. The proxy feature allows IP addresses from the whitelist to interact with the malware.

## Conclusion

Linux/Moose's authors have clearly done a lot of work to stay under the radar with the new version by hiding its C&C server location more effectively and changing the network protocol. By doing this, Moose avoids the Indicators of Compromise (IoCs) released with ESET's 2015 whitepaper. Shortening the whitelist and password list shows a more delicate approach with Moose. Still, some misleading traces are inside the binary like the fake domain `www.challpok.cn` found in cleartext in the list of strings or even filenames that can correspond to bitcoinminer or DDoS malware. Linux/Moose stays exclusively a memory-resident threat; rebooting the embedded device will end its execution.

## Indicators of Compromise

### Hashes

version 0x1F (31)

c6edfa2bf916d374e60f1b5444be6dbbee099692  
c9ca4820bb7be18f36b7bad8e3044b2d768a5db8  
5b444f1ac312b4c24b6bde304f00a5772a6a19a4  
f7574b3eb708bd018932511a8a3600d26f5e3be9

#### **version 0x20 (32)**

34802456d10efdf211a7d486f7108319e052cd17  
0685cb1d72107de63fa1da52930322df04a72dbc  
2876cad26d6dabdc0a9679bb8575f88d40ebd960  
f94b6cc5aea170cee55a238eaa9339279fba962f  
274ef5884cb256fd4edd7000392b0e326ddd2398  
c3f0044ffa9d0bc950e9fd0f442c955b71a706b6  
f3daea1d06b1313ec061d93c9af12d0fe746839a

#### **version 0x21 (33)**

7767c8317fb0bbf91924bddffe6a5e45069b0182  
1caac933ae6ca326372f7e5dd9fff82652e22e34  
5dea6c0c4300e432896038661db2f046c523ce35  
e8dc272954d5889044e92793f0f637fe4d53bb91  
0843239b3d0f62ae6c5784ba4589ef85329350fa  
1d1d46c312045e17f8f4386adc740c1e7423a24a  
d8b45a1114c5e0dbfa13be176723b2288ab12907

#### **version 0x22 (34)**

c35d6812913ef31c20404d9bbe96db813a764886

### ***IP addresses***

#### **Primary C&C servers**

192.3.8.218  
192.3.8.219

#### **Whitelist**

155.133.18.64  
178.19.111.181  
151.80.8.2  
151.80.8.19  
151.80.8.30  
62.210.6.34

Moose's IoCs are also available and updated on ESET's [malware-ioc Github repository](#).

## **Let us keep you up to date**

Sign up for our newsletters



---

Source: <http://www.welivesecurity.com/2016/11/02/linuxmoose-still-breathing/>