Linux Ransomware Debut Fails on Predictable Encryption Key

No need to crack RSA when you can guess the key

———————————

Update: There have been some developments regarding this ransomware. It was brought to our attention that the decryption tool was not working on particular cases. Upon investigation we were surprised to find out that some victims were infected more than one time (the ransomware was accidentally started more than once).

This means that some files were encrypted using a key, and others using another set of keys. However, in so doing, the race condition generated leads to some files getting irreparably damaged (their content is truncated to zero). And in some cases even the ransom notes became encrypted!

We updated the decryption utility and the README. Please read it for the new instructions.

/update

File-encrypting ransomware Trojans are almost ubiquitous on Windows, and it was only a matter of time until the advent of the first piece targeting Linux. Dubbed **Linux.Encoder.1**, this first piece of Linux ransomware is extremely similar in behavior to **CryptoWall**, **TorLocker** and other notorious ransomware families for Windows.

How does it work?

Linux.Encoder.1 is executed on the victim's Linux box after remote attackers leverage a flaw in the popular Magento content management system app. Once executed, the Trojan looks for the */home*, */root* and */var/lib/mysql* folders and starts encrypting their contents. Just like Windows-based ransomware, it encrypts the contents of these files using AES (a symmetric key encryption algorithm), which provides enough strength and speed while keeping system resources usage to a minimum. The symmetric key is then encrypted with an asymmetric encryption algorithm (RSA) and is prepended to the file, along with the initialization vector used by AES.

Once the files have been encrypted, the Trojan attempts to also encrypt the contents of the root (/), skipping only critical system files, so the operating system will be able to boot up again.

At this point, it would be safe to assume that users can't get their data back unless they pay the operators a fee in exchange for the RSA private key to decrypt the AES symmetric one. However, a major flaw in the way the Encoder Trojan is designed allowed Bitdefender researchers to recover the AES key without having to decrypt it with the RSA private key.

A primer on encryption

Throughout 2015, most crypto-ransomware Trojans have used mixed encryption algorithms to hold valuable information hostage. To rapidly and effectively encrypt large amounts of data, crypto-ransomware Trojans rely on the Advanced Encryption Standard (AES for short) – an encryption algorithm that uses a symmetric key (the same key for both encryption and decryption). To avoid interception of the encryption key as it is sent from the command and control server, crypto-ransomware operators usually complement AES with RSA (an asymmetric key encryption algorithm). RSA generates a pair of complementary public-private keys – the public key is used for encryption and the private one for decryption. These keys are usually generated on the hackers' server and only the public key gets sent to the victim PC. Since RSA is less resource-effective on big chunks of data, the public key is only used to encrypt a small, yet critical, piece of information: the encryption key used by the AES algorithm that is generated locally. The RSA-encrypted AES key is then prepended to the beginning of every encrypted file, along with the original file permissions and an initialization vector (IV) used by the AES algorithm.

The million-dollar flaw

We mentioned that the AES key is generated locally on the victim's computer. We looked into the way the key and initialization vector are generated by reverse-engineering the Linux.Encoder.1 sample in our lab. We realized that, rather than generating secure random keys and IVs, the sample would derive these two pieces of information from the libc rand() function seeded with the current system timestamp at the moment of encryption. This information can be easily retrieved by looking at the file's timestamp. This is a huge design flaw that allows retrieval of the AES key without having to decrypt it with the RSA public key sold by the Trojan's operator(s).

Automated decryption tool now available

Bitdefender is the first security vendor to release a decryption tool that automatically restores affected files to their original state. The tool determines the IV and the encryption key simply by analyzing the file, then performs the decryption, followed by permission fixing. If you can boot your compromised operating system, download the script and run it under the root user.

Here is a step-by-step walkthrough to get your data back:

– Download the script from the Bitdefender Labs repository [link updated to include the fix for the recent evolution of the ransomware]

Related posts

Nov.09.2015

in Anti-Malware Research , by Alexandra GHEORGHE

Russian Hackers are Behind Cryptowall 4.0, Bitdefender Creates Vaccine

Sep.01.2015 in Uncategorized , by Liviu Arsene

Malicious Stored XSS Vulnerability in PayPal, Find Bitdefender Researchers

Aug.28.2015

in Anti-Malware Research , by Liviu Arsene

Vulnerability in JetAudio and JetVideo Media Players Allows for Arbitrary Code Execution

Aug.16.2015

in Anti-Malware Research , by Bogdan Botezatu

Dridex Now Targets Romania, Revives Word Macro Infection Technique

Jun.23.2015 in Uncategorized , by Liviu Arsene

Android Instapaper App Vulnerable to Man-in-the-Middle Attacks

Tags

adware android antispam antivirus autorun bitcoin bitdefender research bootkit botnet cloud cryptolocker development e-threat encryption facebook Flame google heuristics icepol immunizer malware mebroot MiniDuke mobile pihar police pushdo ransomware rootkit scam security spam statistics stuxnet TDL3 TDSS tld trojan usb usbimmunizer virus vulnerability wordpress yahoo yurn

FOLLOW US:

twitter rss

(chances are that encryption also affected the system and you might need to boot from a live CD or mount the affected partition on a different machine)
– Mount the encrypted partition using the mount /dev/[encrypted_partition]
– Generate a list of encrypted files by issuing the following command: /mnt# sort_files.sh encrypted_partition > sorted_list
– Issue a head command to get the first file: /mnt# head -1 sorted_list
– Run the decryption utility to get the encryption seed: /mnt# python decrypter.py –f [first_file]
– Decrypt everything using the displayed seed: /mnt# python /tmp/new/decrypter.py -s [timestamp] -l sorted_list

Given the complexity of the task, we provide free support to any user in need of assistance. Just drop us a line via the comment form below and we'll do our best to help you.

Today's Takeaway

If your machine has been compromised, consider this a close shave. Most crypto-ransomware operators pay great attention to the way keys are generated in order to ensure your data stays encrypted until you pay. Mistakes such as the one described above are extremely fortunate, but also extremely rare. Next time, consider a couple precautions:

– Never run applications that you don't completely trust as root user. This is a great security risk that will likely compromise your machine or the integrity of the data on it;
– Backup early, backup often. If your computer falls victim to ransomware, it would be better to simply restore the affected files from an earlier backup than to pay the decryption fee. Remember that easy money is the primary driver for crypto-ransomware operators to build these Trojans and perfect them in time. The less profit they make, the lower their interest in developing crypto-ransomware.

-If your Linux device is on an organization's network, you might want to add a security solution such as Bitdefender Gravity Zone. An antimalware solution blocks this type of threat before it manages to irreversibly encrypt files.

Update your web applications as often as possible. This includes installations of WordPress, Magento or other third-party CMS apps that can be leveraged for remote code execution.

Decryption utility was available courtesy of Bitdefender cryptography specialist Radu Caragea. We would like to extend our thanks to Bitdefender antimalware researchers Codrut Marinescu, Razvan Benchea, Cristina Vatamanu and Alexandru Maximciuc.

118 Responses to *Linux Ransomware Debut Fails on Predictable Encryption Key*

Show Pingbacks

1. ***virender singh*** says:
   November 10, 2015 at 8:00 am

   i have centos 5.8 and i hit by ransomware . i am newby please help me to resolve this issue

2. ***tim*** says:
   November 10, 2015 at 8:35 am

   When running the decrypt i'm getting an error.
   ```
   python decrypter.py –f /home/admin/.bash_logout.encrypted
   File "decrypter.py", line 43
   with open(filepath, 'rb') as f:
   ^
   SyntaxError: invalid syntax
   ```

3. ***Radu*** says:
   November 10, 2015 at 10:48 am

   Tim, you seem to be using an older version of Python that does not have the "with" statement. The script was made for Python 2.7

4. ***Arne*** says:
   November 10, 2015 at 10:55 am

   very nice find. bitdefender 1 – 0 hackers

5. ***tim*** says:
   November 10, 2015 at 11:47 am

   ok, that did the trick. Thanks 🙂
   Copied encrypted files to box with python 2.7 first got the seed from the old box.

6. ***0xdeadbeef*** says:
   November 10, 2015 at 1:35 pm

   hackers hack (either white/black hat). Malware writers write malware. Crackers crack software and code serials. Let's start naming things properly, please.

7. ***Crow*** says:
   November 10, 2015 at 1:40 pm

   Good work, I'll probably change the ToyOS(Win) home computers to Bitdefender.
   Hey, people has to play also 🙂

8. ***Jan*** says:
   November 10, 2015 at 4:11 pm

   Hi! About half of the encryted files get decrypted just fine, but for the other half I get "IV not found for…" messages. What could be the possible cause of such behaviour?
   And when I generate encrypte file list(which is huge in our case), can I pick a block of files(particular project) from the middle of that list and run the script on that particular block only? Does it work this way?

9. ***Jan*** says:
   November 10, 2015 at 4:34 pm

   One more comment. When I run decrypter.py script on full sorted file list it stops after some 10 minutes with following messages:

   ```
   Traceback (most recent call last):
   File “decrypter.py”, line 139, in
   sys.exit(main(**vars(options)))
   File “decrypter.py”, line 115, in main
   decrypt_files(seed, filelist)
   File “decrypter.py”, line 100, in decrypt_files
   decrypt_file(d, filepath)
   File “decrypter.py”, line 61, in decrypt_file
   old_mode = struct.unpack(“<I", f.read(4))[0]
   struct.error: unpack requires a string argument of length 4
   ```

10. ***com*** says:
    November 10, 2015 at 5:58 pm

    Smart approach and doing the public a service, hats off!

11. ***Radu*** says:
    November 10, 2015 at 7:31 pm

    Jan, it’s hard to say without further information. A few questions:
    – Can you pinpoint the exact file that the decryption doesn’t work anymore (IV not found message)? Can you check the files before and after it (in the sorted list) and see if there is any noticeable discrepancy in their timestamps?
    – Exactly how many files are in total? How many are decrypted ok and how many not at all?
    – Can you try changing line 92 in decrypter.py from “range(count * 2)” to something bigger such as 4 or 6 ? (“range(count * 6)”). Let’s see how that goes first.

12. ***Pigsy*** says:
    November 10, 2015 at 10:05 pm

    How does it get to root? how does it get to home?
    Do I download a program and then run the program as root? Is it another one of these selling sand to Arabs?
    You understand anti-malware trying to sell anti-malware to everybody and anybody. Some people are just far too greedy for their own good.

13. ***francesco*** says:
    November 11, 2015 at 2:10 am

    great job, it works very well. But it does memory leak when it comes to process a lot of files in sorted_list.
    a temporary solution is adding swap to the server and modify decrypter.py decrypt_file function as the follow

    ```
    with open(decrypted_path, ‘wb’) as f:
    f.write(decrypted_content)
    del f
    del decrypted_content
    ```

14. ***David Deppner*** says:
    November 11, 2015 at 2:36 am

    One of my test servers got hit with this last week. I just tried this solution on a backup of the VM and can confirm that it works. Details of the attack are available here: http://daviddeppner.com/blog/magento-ransomware

15. ***Jan*** says:
    November 11, 2015 at 11:07 am

    Radu, Your suggestion to change line 92 as written above did not help. Instead I found out that the reason for Your script to stop with “struct.error: unpack requires a string argument of length 4” happens when encrypted file is for some reason shorter than decrypt_file(d, path) is trying to read from it. So basically script stopped when it ran into zero length encrypted file. I solved this one with adding a file size check as follows:

    ```
    with open(path, ‘rb’) as f:
    f.seek(0, os.SEEK_END)
    size = f.tell()
    f.seek(0, os.SEEK_SET)
    if size < 25:
    print 'File is too short, skipped… %s' % path
    f.close()
    return
    ```

    I thought that 25 is a good enough 🙂
    But I'm still struggling with te "IV not found for…" files.

16. ***Radu*** says:
    November 11, 2015 at 11:31 am

    Jan, I’ll drop you a mail shortly to discuss things in more detail.

17. ***cFire*** says:
    November 11, 2015 at 12:20 pm

    I am also curious how it gets root access. Or does it just attempt to encrypt things and only succeeds if magento runs as root or your permissions are broken? Or does it actually have some privilege escalation mechanism?

18. ***Aidus*** says:
    November 11, 2015 at 1:51 pm

    Same to Jan’s problem. A lot of files with “IV not found for…”

19. ***Radu*** says:
    November 11, 2015 at 2:58 pm

    Jan, Aidus. We have identified the problem and we’re working on a fix.

20. ***Bogdan Botezatu*** says:
    November 11, 2015 at 5:29 pm

    We have updated the article and the tool to work properly when the ransomware starts more than once on the same machine.

21. ***Coder*** says:
    November 11, 2015 at 8:30 pm

    Hi, I follow all the steps, but when I try to “Find the seed” step, I get this error:

    ```
    Traceback (most recent call last):
    File “./decrypter.py”, line 142, in
    sys.exit(main(**vars(options)))
    File “./decrypter.py”, line 112, in main
    seed = find_seed(filename)
    File “./decrypter.py”, line 49, in find_seed
    key = f.read(key_sz)
    ```

```
OverflowError: Python int too large to convert to C long
```
Can you help me? Why occurs this?

22. ***Radu*** says:
    November 11, 2015 at 11:11 pm

    Coder, what is the exact command you are running? (And can you please redownload and use the updated version of the decrypter archive?)

23. ***Alin*** says:
    November 12, 2015 at 4:33 am

    Hello ! In my case the things are a bit different.
    The attack was made through an abandoned Magento install.
    BUT – they encrypted ONLY the /var/www folder
    and only the apache owned files. The files owned by root were not encrypted.
    How should I know if this decrypter applies for me?

24. ***Alin*** says:
    November 12, 2015 at 4:42 am

    I cannot find the /tmp/new/decrypter.py path

25. ***Alin*** says:
    November 12, 2015 at 4:44 am

    I tried to run like this:
    ```
    # python decrypter.py -s 1446655061 -l sorted_list
    [!] The seed, filelist, and errorfilelist are all required
    ```

26. ***Coder*** says:
    November 12, 2015 at 8:22 am

    Hi, I try with the new version, but I get the same error. My steps were the following:
    ```
    root@kali:~/Escritorio# bash decrypter/sort_files.sh > sorted_list
    root@kali:~/Escritorio# head -1 sorted_list
    1446655006.0000000000 ./decrypter/pdf.encrypted
    root@kali:~/Escritorio# python decrypter/decrypter.py -f ./decrypter/pdf.encrypted
    Traceback (most recent call last):
    File “decrypter/decrypter.py”, line 182, in
    sys.exit(main(**vars(options)))
    File “decrypter/decrypter.py”, line 150, in main
    seed = find_seed(filename)
    File “decrypter/decrypter.py”, line 65, in find_seed
    iv_file, _ = parse_header(filepath)
    File “decrypter/decrypter.py”, line 49, in parse_header
    key = f.read(key_sz)
    OverflowError: Python int too large to convert to C long
    ```
    Am I doing something wrong?

27. ***Coder*** says:
    November 12, 2015 at 8:26 am

    Maybe, for the next reply, I can use the return key to be a little more readable. Sorry for that.

28. ***Jan*** says:
    November 12, 2015 at 9:04 am

    Latest script which I downloaded some half an hour ago(Decrypter_0-1.3), is unable to get seed from the first file of my sorted_list, it does not do anything at all. I have to stop the script with Ctrl-C. However older version 0.2 of the decrypter.py scipt provides correct seed from the same file with -f option instantly.

29. ***Radu*** says:
    November 12, 2015 at 9:16 am

    Jan, you need to leave it longer as it now accounts for the multiple seed case. I gave you more details in the mail yesterday but you haven’t responded

30. ***Radu*** says:
    November 12, 2015 at 9:17 am

    Coder, I dropped you a mail for further info. Please check your inbox

31. ***Radu*** says:
    November 12, 2015 at 9:19 am

    Alin, please read the README, you have step by step instructions there. (in your case you are missing the -e parameter from step 5)

32. ***Alin*** says:
    November 12, 2015 at 9:30 am

    Hi Radu ! Can you please help me with some tips on how to solve in my case?

33. ***Alin*** says:
    November 12, 2015 at 9:34 am

    sorry ! I just read your message above

34. ***Jon*** says:
    November 12, 2015 at 10:31 am

    decrypter 1.3 hangs and does nothing for files that 1.2 worked fine on. Deb 7, python 2.7. Sits on a read in strace and never completes.

35. ***nik247*** says:
    November 12, 2015 at 10:48 am

    I can’ t get “seed” with both version too.
    First 3 files from step 3:
    ```
    # ls -l ./files/*
    -rw-rw-rw- 1 nik nik 2329 Aug 25 04:17 ./files/dhcp-mikrotik-static.sh.encrypted
    -rw-rw-rw- 1 nik nik 36211 Aug 25 04:17 ./files/forumdisplay.php.encrypted
    -rw-rw-rw- 1 nik nik 295618 Aug 25 04:17 ./files/fsbackup-1.2pl2.tar.gz.encrypted
    ```
    Step 4:
    ```
    # python decrypter_v2.py -f ./files/dhcp-mikrotik-static.sh.encrypted
    Traceback (most recent call last):
    File “decrypter_v2.py”, line 182, in
    sys.exit(main(**vars(options)))
    File “decrypter_v2.py”, line 150, in main
    seed = find_seed(filename)
    File “decrypter_v2.py”, line 65, in find_seed
    iv_file, _ = parse_header(filepath)
    ```

```
File “decrypter_v2.py”, line 49, in parse_header
key = f.read(key_sz)
OverflowError: Python int too large to convert to C long
# python decrypter_v2.py -f ./files/forumdisplay.php.encrypted
Traceback (most recent call last):
File “decrypter_v2.py”, line 182, in
sys.exit(main(**vars(options)))
File “decrypter_v2.py”, line 150, in main
seed = find_seed(filename)
File “decrypter_v2.py”, line 65, in find_seed
iv_file, _ = parse_header(filepath)
File “decrypter_v2.py”, line 49, in parse_header
key = f.read(key_sz)
OverflowError: Python int too large to convert to C long
# python decrypter_v2.py -f ./files/fsbackup-1.2pl2.tar.gz.encrypted
[!] File is probably truncated
[!] Seed not found! Timestamps corrupt?
```

36. ***Aidus*** says:
    November 12, 2015 at 10:55 am

    Hi Radu
    Trying to locate second seed in first file from error.list. Seems like script hangs.

37. ***Radu*** says:
    November 12, 2015 at 11:27 am

    The new version accounts for multiple infections and because of this takes significantly longer when finding the seed. It does not hang, please let it finish what it is doing.
    Also, please note that this decrypter ONLY works for Linux.Encoder.1. Trying files encrypted with something else will not work. (For example, someone sent me a file encrypted with TeslaCrypt; won’t work)

38. ***nik247*** says:
    November 12, 2015 at 11:42 am

    Hi Radu.
    How I can detect encrypted with Linux.1.Encoder?

39. ***Aidus*** says:
    November 12, 2015 at 11:50 am

    Radu. I have decoded about 70% of files. 30% were encoded at the same time, so I think it is Linux1.Encoder too. Just maybe other seed. But script (finding seed in first file) is working more than 1 hour. Is that normal? Size of file is 5832 bytes.

40. ***Radu CARAGEA*** says:
    November 12, 2015 at 12:23 pm

    Aidus, I sent you a mail for further info

41. ***Alin*** says:
    November 12, 2015 at 1:04 pm

    IT WORKED !!!!!!

42. ***Aidus*** says:
    November 12, 2015 at 2:26 pm

    Radu, sorry, don’t see email from you.

43. ***Petschko*** says:
    November 12, 2015 at 3:11 pm

    Same here… it seems that step 4 isn’t working at all… It says Timestamp not found or it hangs for hours…is there a solution?
    In the most cases I can use the number that is displayed before the file in step 3 use the first numbers as seed to the dot. but sometimes it doesn’t and then I have to use step 4 but it wont work =/

44. ***SysAdm*** says:
    November 12, 2015 at 4:00 pm

    Upon executing decrypter.py I get this error message:
    ImportError: No module named Crypto.Cipher I am running python 2.7.7 and I have python-dev package installed. Any ideas?

45. ***Douglas de Souza*** says:
    November 12, 2015 at 4:28 pm

    Sorry, but, i dont understand the step-by-step walkthrough, i am in aws, and i have magento files cryption, i can ´t mount another instance. What I really have to do? Sorry for my ignorance. Very limited. And I have no help from outside.Thanks in advance for the help.

46. ***Jan*** says:
    November 12, 2015 at 8:31 pm

    Sysadm, Don’t know which Linux distro you have, but on Debian and alike systems it means just “apt-get install python-crypto” or something similar.

47. ***Jan*** says:
    November 12, 2015 at 8:59 pm

    Also, I’m happy to report that I managed to decrypt all our files with the help of this particular script made by Bitdefender team. Thank You, it works.

48. ***CDM*** says:
    November 12, 2015 at 9:23 pm

    My files say .gpg instead of .encrypted. Changing the script to look for gpg I get the first file to have been encrypted. Running the decrypter I only get memoryerror on line 49 (key = f.read(key_sz)

49. ***SysAdm*** says:
    November 12, 2015 at 9:23 pm

    Ubuntu 14.04 – already installed python-crypto, pip freeze shows pycrypto==2.6.1

50. ***Gera*** says:
    November 12, 2015 at 11:06 pm

    I have only backup a part of encrypted files before reinstall the server. There is no way to decrypt only files I still have?

51. ***Javad*** says:
    November 12, 2015 at 11:21 pm

    My backup file in cPanel is infected. Can I download the backup and use your tool on a local machine? I’m not familiar with pyton or linux.

52. ***Radu CARAGEA*** says:
November 13, 2015 at 9:05 am

SysAdm, you have a system configuration problem most likely. So sorry, I can't help you with that.

53. ***Radu CARAGEA*** says:
November 13, 2015 at 9:51 am

Gera, you need the original filesystem (with the original timestamps) to be able to decrypt files

54. ***Bertil*** says:
November 13, 2015 at 4:43 pm

Hi, i am running in the following error while decrypting

```
Traceback (most recent call last):
File "./decrypter.py", line 182, in
sys.exit(main(**vars(options)))
File "./decrypter.py", line 156, in main
decrypt_files(seed, filelist, errorfilelist)
File "./decrypter.py", line 127, in decrypt_files
if decrypt_file(d, filepath):
File "./decrypter.py", line 98, in decrypt_file
with open(decrypted_path, 'wb') as f:
IOError: [Errno 13] Permission denied: '/srv/www/magento_3e/peler/phpmailer.zip'
```

so there is a file wich the decrypter can not access – what solution do you suggest?
Many thanks

55. ***Radu CARAGEA*** says:
November 13, 2015 at 4:52 pm

Bertil, you do not have permissions for that directory. It says that it can't create a new file with that path in order to write the decrypted content inside. Fix the permissions or run as a privileged user.

56. ***Javad*** says:
November 13, 2015 at 7:23 pm

I wanted to thank you for creating this wonderful tool. I managed to run it on my server with the help of Sucuri team. Thanks again!

57. ***Radu CARAGEA*** says:
November 13, 2015 at 8:57 pm

Javad, good to know you got your files back!

58. ***Javad*** says:
November 13, 2015 at 11:21 pm

Is prycrypto required to run the script?

59. ***Jeli*** says:
November 14, 2015 at 11:58 am

Hi, my problem is that I dont have the first encrypred file/timestamp anymore. So I am trying to go backwards in the filechangedtime fron an enrypted file I still have by changing
ts = int(os.path.getmtime(filepath))
to
ts = int(os.path.getmtime(filepath)) – 3600 * x
I am realy no expert im python, so I would like to ask you, if I understood the script right and if this will work.
Thanks for your Time an afford.

60. ***Radu CARAGEA*** says:
November 14, 2015 at 1:06 pm

Jeli, you first need to find the timestamp (seed).
If this file was encrypted among the first 10000 files within 3600 seconds of modification time it should find the timestamp. That is what the "find_seed" function does. So if you think that the file you require is among the first 10000 encrypted then you can try the step to find the seed on it. (You can add a print for the "i" variable at line 71 to see the search progress)
But if you copied the file from somewhere else it will not work (the modification time changes)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name

Email

SEND MESSAGE »

Recommended
Malware City
Unices
Bitdefender Forum

Bitdefender
Bitdefender Antivirus
About
Contact

Lab Area
Homepage
Projects
Blog
Contact

Bitdefender LABS powering: Bitdefender