Jurnal Sistem Komputer dan Informatika (JSON)
Volume 2, Nomor 3, Mei 2021
e-ISSN 2685-998X
DOI 10.30865/json.v2i3.2998

Analisis Forensik Digital Memori *Volatile* untuk Mendapatkan Kunci Enkripsi Aplikasi Dm-Crypt

Vipkas Al Hadid Firdaus[1,*], Dodit Suprianto[1], Rini Agustina[2]

[1] Jurusan Teknik Informatika, Politeknik Negeri Malang, Malang, Indonesia
[2] Program Studi Sistem Informasi, Universitas Kanjuruhan Malang, Malang, Indonesia
Email: [1,*]vipkas@polinema.ac.id, [2]dodit.suprianto@polinema.ac.id, [3]riniagustina@unikama.ac.id
Submitted: **01/05/2021**; Accepted:**12/05/2021**; Published: **30/05/2021**

Abstrak–Teknologi enkripsi disk merupakan teknologi yang sangat bermanfaat dalam mengamankan data. Di sisi lain, enkripsi disk juga dapat digunakan oleh pelaku kejahatan untuk menyembunyikan bukti digital. Informasi dalam disk akan sangat berguna untuk penyelidikan, tetapi jika disk pada barang bukti digital terenkripsi maka hal tersebut akan menghambat proses penyelidikan. Kondisi tersebut tentu akan menjadi tantangan bagi penyidik *cybercrime* untuk bagaimana mendapatkan kunci enkripsi disk tersebut, terlebih jika pelaku tidak bekerjasama dalam proses penyidikan. Dalam mengatasi masalah ini, analisis pada memori *volatile* pada komputer untuk mendapatkan kunci enkripsi akan membantu dalam penyelidikan. Keseluruhan data pada memori *volatile* komputer yang menjadi barang bukti tindak kejahatan akan direkam untuk kemudian di analisis, dengan menggunakan hasil *dump* memori yang dilakukan secara *Live* pada barang bukti komputer yang diperoleh pada saat wadah terbuka, kunci dekripsi dapat dipulihkan. Dalam makalah ini dibahas mengenai analisis forensik memori *volatile* untuk mendapatkan kunci enkripsi disk aplikasi *dm-crypt* yang digunakan untuk mengenkripsi disk pada sistem operasi Linux dan membuktikan bahwa melalui analisis forensik memori *volatile* pada hasil *dump* memori komputer secara *Live*, kunci enkripsi disk *dm-crypt* dapat ditemukan dengan persentase keberhasilan 80%. Pada makalah ini penelitian difokuskan pada analisis forensik memori di sistem operasi Linux dengan aplikasi enkripsi dm-crypt.

Kata Kunci: Forensik Komputer; Enkripsi Disk; Forensik Digital; Dm-Crypt; Live Forensic Aqquisition

Abstract–Disk encryption technology is something very useful in securing data. On the other hand, disk encryption can be used by criminals to hide the digital evidence. The information in the disk will be very useful for the investigation, but if the disk on the computer evidence encrypted then it will hamper the investigation process. The conditions will certainly be a challenge for investigator cybercrime to be able to find the disk encryption key, especially if the perpetrator did not cooperate in the investigation process. The analysis of the image memory to get the encryption key will be helpful in the investigation. In the overall memory activity on the computer evidence will be recorded, using a live image memory dump on the computer evidence, the decryption keys can be recovered. This paper will discuss about forensic analysis to getting the disk encryption key on the dm-crypt is used to encrypt the disk on Linux operating system and prove that through forensic image memory on a live memory dump, key dm-crypt disk encryption can be found with a success percentage of 80%. On this paper the research will be focused on the Linux operating system with dm-crypt function to full disk encryption.

Keywords: Computer Forensic; Disk Encryption; Digital Forensic; Dm-Crypt; Live Forensic Acquisition

1. PENDAHULUAN

Di bidang digital forensik, pengumpulan bukti digital sangat penting dalam penyelidikan cybercrime. Informasi digital tentang bukti disk komputer akan sangat membantu dalam menyelesaikan kasus kejahatan siber dan akan menjadi bukti dalam proses pengadilan. Oleh karena itu, dalam setiap kasus kejahatan siber, mengamankan danmengumpulkan bukti digital terhadap prioritas utama sebelum penyelidikan dilakukan. Seperti yang dikatakan dalam [1] "memori yang mudah menguap adalah komponen penting dari TKP digital", jadi bagi penyelidik menjadi praktik umum dalam proses penyelidikan untuk membuat gambar memori dump.

Peningkatan teknologi enkripsi disk di satu sisi dapat mengamankan informasi dari kejahatan siber, tetapi di sisi lain dalam beberapa kasus penjahat pada saat ini, enkripsi disk juga digunakan oleh penjahat untuk melindungi atau menyembunyikan bukti dari kejahatan penyelidik, yang akan membuat kejahatan tidak terbukti dan tidak terungkap dalam proses pengadilan [2]. Kondisi itu menjadi tantangan bagi penyidik untuk mendapatkan cara menemukan dan mengekstrak kunci enkripsi disk pada bukti komputer. Untuk mendapatkan kunci enkripsi disk, akan lebih mudah jika yang pelakuk kejahatan dapat bekerjasama, tetapi jika tidak, penyelidik membutuhkan cara lain untuk mendapatkan kunci [3].

Dalam teknologi enkripsi disk telah ada beberapa alat yang dapat digunakan untuk mengenkripsi disk komputer, seperti TrueCrypt, Bitlocker, dm-crypt dll [4]. Selain sistem operasi Linux juga telah mengembangkan alat baru untuk melakukan enkripsi disk. Enkripsi disk di Linux dapatdilakukan menggunakan dm-crypt yang merupakan alat yang disediakan oleh linux. dm-crypt adalah subsistem enkripsi diskyang suport di linux kernel versi 2.6 dan yang lebih baru. Dm-crypt juga menjadi bagian dari infrastruktur pemetaan perangkat, dan menggunakan rutinitas kriptografi dari Crypto API kernel.

Pengembangan teknologi enkripsi disk sangat berperan dalam membantu mengamankan data, tetapi sisi lain dari teknologi ini juga digunakan oleh penjahat untuk menyembunyikan bukti. Dalam digital forensik, pecahkan kunci enkripsi pada enkripsi disk menjadi diperlukan untuk mendapatkan informasidigit al padabukti penyimpanan disk. Dengan menerapkan metode penggunaan memori forensik, kunci beberapa alat enkripsi disk seperti BitLocker, TrueCrypt dapat ditemukan. Di lingkungan windows, kunci enkripsi dapat ditemukan bitlocker

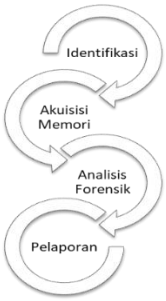
Jurnal Sistem Komputer dan Informatika (JSON)
Volume 2, Nomor 3, Mei 2021
e-ISSN 2685-998X
DOI 10.30865/json.v2i3.2998

[5], [6]. Dalam penelitian sebelumnya, analisis untuk menemukan kunci enkripsi pada memory dump telah dilakukan. Di ingkungan Windows, kunci enkripsi dapat ditemukan kunci enkripsi bitlocker, mereka berhasil memulihkan informasi drive dengan sistem analisis metode langsung pada dua mode enkripsi bitlocker. Studi lain telah dilakukan oleh [7] pada TrueCrypt dengan lingkungan Windows, mereka menganalisis struktur memori menggunakan pemindaian linear dan telah mengembangkan perangkat lunak untuk menemukan kunci secara tomatis, dan juga mereka juga telah berhasil mendapatkan kunci trueCrypt encryption ". Sementara di lingkungan sistem operasi Linux, enkripsi disk dapat dilakukan menggunakan TrueCrypt dan dm-crypt. Analisis pada dump memori, terkait menemukan kunci enkripsi truecypt yang berfokus terutama pada pada distribusi Linux Ubuntu telah dilakukan oleh [8]. Dalam studi itu, analisis terkait untuk mendapatkan kunci enkripsi yang dilakukan dengan mengidentifikasi kunci enkripsi truecypt dalam dump memori, mereka menggunakan dua pendekatan major, yaitu dengan mencari kunci pada alamat memori tetap dan dengan menemukan pola tertentu dalam memori.

Studi ini [5], dan [7] berfokus pada lingkungan jendela dalam analisis untuk mendapatkan kunci enkripsi di BitLocker dan TrueCrypt. Dan pada lingkungan linux, [9] melakukan analisis dump memori yang berfokus dalam menemukan kunci enkripsi yang digunakan oleh TrueCrypt. Pada sistem operasi linux, enkripsi disk tidak hanya dapat dilakukan dengan menggunakan TrueCrypt, tetapi juga dapat dilakukan kitaing dm-crypt. Dengan demikian, dalam makalah ini akan dibahas pendekatan dalam melakukan analisis forensik pada dump memori linux, untuk mendapatkan kunci enkripsi digunakan oleh dm-crypt, dan membuktikan bahwa melalui memori gambar forensik dm-crypt kunci enkripsi dapat ditemukan. Makalah ini akan membahas analisis forensik untuk mendapatkan kunci enkripsi disk pada DM-Crypt yang digunakan untuk mengenkripsi disk pada sistem operasi Linux. Sekaligus membuktikan bahwa melalui analisis forensic pada memori volatile komputer, kunci enkripsi disk dm-crypt dapat ditemukan. Dalam penelitian ini, proses akuisisi memori dilakukan menggunakan LiME dan analisis dilakukan melalui analisis terstruktur terhadap alamat memori aplikasi enkripsi dm-crypt.

2. METODE PENELITIAN

Dalam melakukan analisis kunci enkripsi disk pada sistem operasi linux menggunakan dm-crypt, pada Gambar 1 tahapan dalam memahami terkait detail proses aplikasi dalam melakukan enkripsi disk menjadi bagian penting untuk melakukan analisis forensik dan mengekstrak kunci enkripsi dari hasil akuisis memori [10]. Selain itu, tahapan metode dalam melakukan akuisisi memori harus memenuhi standar baku agar hasil temuan dapat diterima sebagai bukti digital dalam persidangan. Di bagian ini akan dibahas tentang detail bagaimana dm-crypt dalam mengenkripsi disk pada sistem operasi Linux serta metode yang digunakan dalam penelitian ini untuk bagaimana menganalisa dan menemukan kunci enkripsi yang digunakan.



Gambar 1. Metode Analisis Forensik

2.1 Dm-crypt

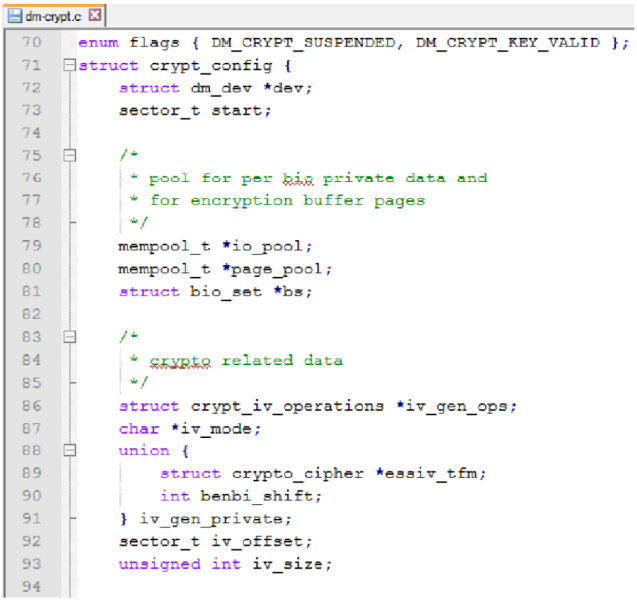
Dm-crypt adalah metode enkripsi disk, yang merupakan subsistem dalam kernel Linux. Pada linux, dukungan dm-crypt di kernel Linux versi 2.6 ke atas yang membangun infrastruktur pemetaan perangkat [11]. Subsistem dm-crypt mendukung struktur Linux Unified Key Setup (LUKS), yang memungkinkan beberapa kunci untuk mengakses data terenkripsi, serta memanipulasi tombol (seperti mengubah kunci, menambahkan frasa sandi tambahan, dll.) Pemetaan perangkat dm-crypt menargetkan Residi seluruhnya di ruang kernel, dan hanya berkaitan dengan enkripsi perangkat blok dm-crypt untuk dienkripsi menggunakan rutinitas kriptografi dari kernel CryptoAPl, dan dirancang khusus untuk mendukung beberapa mode seperti XTS, LRW dan ESSIV. Di dm-crypt, ketika akan digunakan untuk mengenkripsi disk, file digunakan sebagai kontainer mereka pertama-tama perlu dikonversi ke block-devices menggunakan *loop-subsystem*.

2.2 Proses Penguncian Pada dm-crypt

Pada dm-crypt, kunci yang digunakan untuk mengenkripsi akan dikonfigurasi sesuai dengan keinginan pengguna. Kunci enkripsi akan dikonfigurasi pertama kali saat aplikasi dm-crypt digunakan. Konfigurasi dm-crypt akan tetap disimpan di crypt_config selama proses enkripsi pada computer berjalan. Selanjutnya, struktur penyimpanan kunci

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 2, Nomor 3, Mei 2021
e-ISSN 2685-998X
DOI 10.30865/json.v2i3.2998

dalam dm-crypt yaitu cryp_config akan ditentukan dalam file dm-crypt.c yang untuk lebih jelasnya, lihat gambar 1 di bawah ini.



Gambar 2. Dm-crypt Source List

2.3 Proses Penguncian Pada dm-crypt

Proses penguncian dm-crypt dilakukan dengan menggunakan aplikasi cryptsetup. Pada cryptsetup, keying support dengan beberapa mode operasi enkripsi sebagaimana tertera pada Tabel 1, tetapi secara umum dan secara default dm-crypt menggunakan mode LUKS untuk proses keying. Yang lain dari format dm-crypt adalah mode polos din-crypt, mode legacy loopaes, dan mode kompatibilitas Truecrypt. Dalam tulisan ini, kita akan fokus pada mode LUKS, Dalam metode mode LUKS, cryptsetup di dm-crypt disebut luksFormat. Fungsi luksformat, akan mengatur header LUKS dan akan mengenkripsi master-key dengan semacam enkripsi sesuai dengan pilihan penggunaan.

Tabel 1. Daftar Mode Operasi Enkripsi

Options	Cryptsetup	Example
-cipher, -c	Acs-xts-plain64	acs-xts-plain64
-key-size, -s	256	512
-hash, -h	sha1	Sha512
-iter-time, -i	1000	5000
-use-random	--use-urandom	--use-random
-verify-passphrase, -y	Yes	-

2.4 Proses Penguncian Pada dm-crypt

Tahap akuisisi data pada penelitian ini dilakukan secara live forensics, Tahapan akuisisi data secara live forensics mengacu pada penelitian yang dilakukan sebelumnya oleh [12], [13]. Pada penelitian lainnya yang dilakukan oleh [14] dilakukan akuisisi terhadap random access memory windows untuk menemukan *user_id* dan password. Dalam penelitian kali ini selain meneliti random access memory untuk mendapatkan informasi terkait user_id, password, email serta informasi lainnya juga untuk meneliti bagaimana karakteristik random access memory dan bagaimana kondisi random access memory pada perangkat laptop setelah dilakukan hibernate. Kondisi utama yang harus dipenuhi dalam *live forensics* adalah sistem dalam kondisi running atau beroperasi dikarenakan beberapa data dan informasi pada random access memory bersifat volatile artinya jika komputer mati atau reebot maka data akan hilang [15]. Untuk itu perlu dilakukan penanganan secara khusus dalam melakukan akusisi data yang bersifat volatile.

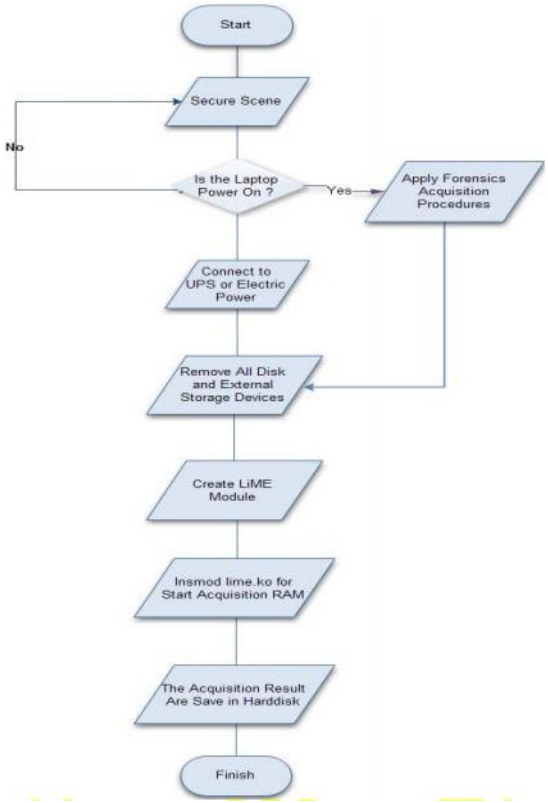
Karena data dan informasi tertentu dalam *random access memory* bersifat volatile, artinya jika *power sullply* pada perangkat komputer dalam kondisi *off* atau perangkat dihidupkan ulang maka akan berdampak pada hilangnya data. Oleh sebab itu persyaratan utama yang harus dipenuhi dalam akuisisi *random access memory* secara langsung adalah bahwa sistem pada perangkat harus dalam kondisi menyala. Untuk itu maka diperlukan persiapan serta perhatian yang cermat dalam melaksanakan proses akuisisi data pada memori *volatile* [8].

2.5 Rancangan Proses Akuisisi RAM

Untuk melakukan penanganan barang bukti komputer diperlukan suatu tahapan yang akan menggambarkan alur proses penanganan barang bukti tersebut. Disamping itu diperlukan juga bagaimana tahapan dalam melakukan

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 2, Nomor 3, Mei 2021
e-ISSN 2685-998X
DOI 10.30865/json.v2i3.2998

akusisi data guna mendapatkan bukti informasi digital yang secara sah dan valid dapat diterima sebagai barang bukti digital dalam persidangan [16]. Tahapan tersebut menjelaskan alur proses dalam penanganan barang bukti digital serta bagaimana langkah-langkah dalam melakukan proses akuisisi data dari perangkat fisik memori volatile pada komputer untuk dapat diperoleh sebuah barang bukti digital yang sah dan diakui secara hukum serta dapat digunakan untuk proses analisis forensik dan menemukan informasi dalam bentuk kunci enkripsi yang digunakan oleh aplikasi dm-crypt maka diperlukan sebuah tahapan prosedur untuk mengelola barang bukti digital dari perangkat komputer melalui proses akuisisi memori volatile. Tahapan prosedur tersebut tercantum pada Gambar 3 berikut.



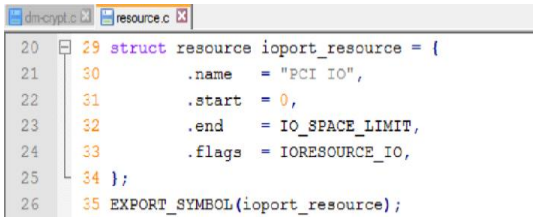
Gambar 3. Flowchart Proses Akuisisi Random Access Memory

3. HASIL DAN PEMBAHASAN

Pada bagian ini dibahas terkait proses analisis yang dilakukan untuk mendapatkan kunci enkripsi dari aplikasi dm-crypt. Selanjutnya juga akan dibahas terkait detail dari alur proses dalam melakukan analisis forensic pada memori volatile. Dan pada bagian akhir dari bab ini disajikan terkait hasil dari penelitian ini berupa temuan kunci enkripsi yang digunakan untuk melakukan enkripsi disk.

3.1 Getting Memory Image in Linux

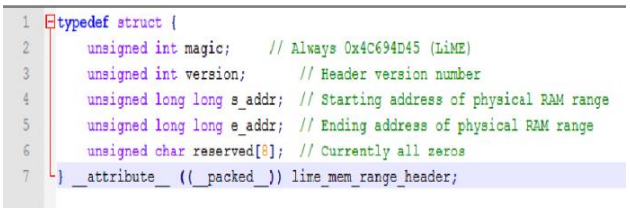
Di linux, kita bisa mendapatkan image memory dengan menggunakan tool lime (Linux Memory Extractor). Untuk melakukan dumping memori menggunakan Linux, pertama kita harus menginstal perangkat lunak LiME, LiME dapat melakukan cloning secara langsung dari github. Lime juga dapat diinstal dan dimuat ke dalam kernel Linux, selain itu dapat digunakan untuk melakukan dumping fisik memori ke komputer melalui jaringan atau menyimpan hasil dumping memori fisik di penyimpanan lokal komputer, LiME akan memanfaatkan iomem_resources. Untuk melakukan dumping atau akuisisi fisik memori, LiME akan menggunakan iomem_resources. Pada Gambar 4 berikut ini adalah struktur kernel, yang digunakan untuk mendapatkan kisaran alamat memori fisik.



Gambar 4. Daftar sumber resource.c

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 2, Nomor 3, Mei 2021
e-ISSN 2685-998X
DOI 10.30865/json.v2i3.2998

Pada LiME, untuk menyimpan image memori dari hasil proses akuisisi ada tiga format image yang bisa digunakan. Ketiga format tersebut adalah raw, padded, dan lime. Dari ketiga format Image tersebut, pada makalah ini akan digunakan format image dalam fomat raw untuk image memori yang diakusisi. Setiap hasil akuisisi memori yang dibangun dengan format LiME akan memiliki ukuran header yang tetap sebagaimana terlihat pada Gambar 5 berikut.



```
typedef struct {
    unsigned int magic;     // Always 0x4C694D45 (LiME)
    unsigned int version;       // Header version number
    unsigned long long s_addr;  // Starting address of physical RAM range
    unsigned long long e_addr;  // Ending address of physical RAM range
    unsigned char reserved[8];  // Currently all zeros
} __attribute__ ((__packed__)) lime_mem_range_header;
```

Gambar 5. LiMe Image Format Header

Setelah dilakukan proses *compile*, hasilnya akan terdapat modul yang bernama sama dengan nama kernel yang kita gunakan. Dalam makalah ini, memiliki nama *lime-3.16.0-34 file-generic.ko*. Selanjutnya kita perlu menyalin modul tersebut ke *flash drive*, yang selanjutkan akan digunakan untuk akuisisi memori komputer. Tahapan selanjutnya adalah dengan jalankan proses *memory dump* dengan menggunakan LiME. Dan didapatkan hasil akuisisi memoi dengan nama file ouput dari hari hasil *memory dump* adalah hasil.raw.

```
root@ubuntu:~$ ls –lh /media/mrrwx/ADATA_123/
total 1,7G
-rw-r—r—1 mrrwx 1,7G Okt 14 14:36 hasil.raw
-rw-r—r—1 mrrwx 7,1K Okt 14 14:28 lime-3.16.0-34-generic.ko
```

Gambar 6. Hasil dari *Image Memory Dump*

3.2 Analisa Memory Dump

Setelah akuisisi memori selesai, selanjutnya memori dump atau image memory akan dianalisa. Pada dm-crypt, struktur dari file crypt_config tidak memberikan informasi apa pun tentang kunci yang digunakan untuk mengenkripsi. Struktur dari crypt_config hanya memberikan sedikit informasi tentang pointer, pointer yang terdapat pada crypt_config adalah dev, io_pool, page_pool, iv_gen_ops, iv_mode, iv_gen_private dan tfm. Dalam beberapa kasus, membedakan pointer dari struktur perangkat untuk menemukan key mungkin dilakukan, tetapi akan sangat sulit untuk membuktikannya.

Meskipun, untuk menemukan kunci dengan membedakan pointer dari struktur perangkat sangat sulit, tetapi pada makalah ini diasumsikan bahwa keseluruhan pointer terletak pada alamat dengan nilai OxC0000000 atau lebih. Jadi, dimungkinkan untuk mencari alamat memori dengan kisaran tertentu. Dalam makalah ini, kami merancang sebuah tahapan dalam melakukan analisis kunci enkripsi dm-crypt dari hasil memory dump.

Tabel 2. Hasil Evaluasi Implementasi Metode Forensik Memori

	Akuisisi Memori *Volatile*	Password dm-crypt
Skenario 1	Berhasil	Ditemukan
Skenario 2	Berhasil	Ditemukan
Skenario 3	Berhasil	Ditemukan
Skenario 4	Berhasil	Ditemukan
Skenario 5	Berhasil	Tidak Ditemukan

Dengan asumsi kunci enkripsi terletak di alamat dengan nilai OxC0000000, maka lakukan pencarian pada pointer. Kunci enkripsi dm-crypt yang digunakan dalam penelitian ini menggunakan AES, sehingga nilai panjang iv_size dan key_size adalah 16 bit atau 32 bit. Berdasarkan asumsi tersebut maka pencarian alamat untuk nilai tertentu dapat dilakukan. Dalam proses enkripsi disk dengan dm-crypt kunci enkripsi akan secara otomatis dimuat oleh cryptsetup saat pertama kali pengguna menggunakan dm-crypt. Jadi dengan analis pada memory dump dengan alamat tertentu, kunci enkripsi akan dapat ditangkap. Pada dm-crypt, cryptsetup-LUKS di luksheader khusus, ketika dimulai proses enkripsi disk, perangkat komputer akan menambahkan fitur terkait semua opsi enkripsi yang dapat digunakan. Sehingga informasi tentang kriptografi dan kunci yang dipilih oleh pengguna akan diketahui. Pada Gambar 7 berikut adalah hasil dari analisis terkait kunci enkripsi dari dm-crypt yang dapat ditemukan melalui proses analisis forensic pada hasil akuisisi memori perangkat komputer di system operasi Linux. Pada Tabel 2 menyajikan hasil pengujian yang telah dilakukan terkait akuisisi memori dan temuan password dm-crypt.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 2, Nomor 3, Mei 2021
e-ISSN 2685-998X
DOI 10.30865/json.v2i3.2998

keylenght: 256
key:
4B6E4FE4612D456CDF81361C43845128D28DDA4546280E769
8C2D5274F66192

Gambar 7. Hasil Temuan Kunci Enkripsi

4. KESIMPULAN

Pada makalah ini hasil analisis forensik pada memori komputer menunjukkan hasil bahwa kunci enkripsi disk dari dm-crypt dapat ditemukan. Analisis dilakukan dengan melakukan pencarian pada alamat yang diasumsikan sebagai tempat penyimpanan kunci, dengan terlebih dahulu diketahui algoritma kriptografi, mode enkripsi dan alat yang digunakan oleh pengguna untuk melakukan enkripsi pada disk komputer. Melalui studi ini, desain rancangan tahapan forensik pada kunci enkripsi dm-crypt dapat dibuktikan, bahwa kunci enkripsi dm-crypt dapat ditemukan melalui prose analisis forensik memori. Dengan menemukannya kunci enkripsi dari *memory dump*, kedepan akan sangat membantu terutama bagi penyelidik forensik untuk mengekstraksi informasi dari disk yang terenkripsi. Dalam makalah ini analisis difokuskan pada *memry dump* pada sistem operasi Linux, dan untuk mendapatkan citra memori digunakan LiME (Linux Memor Extractor). Untuk menemukan kunci enkripsi dm-crypt difokuskan pada mekanisme mode LUKS dengan menggunakan kriptografi AES. Tahapan analisis forensik akan dapat bekerja secara optimal jika algoritma kriptografi, mode enkripsi dan perangkat lunak yang digunakan untuk enkripsi telah diketahui. Untuk kasus lain dimana informasi tersebut belum diketahui, maka dapat menjadi area studi lebih lanjut.

REFERENCES

[1] M. I. Al-Saleh, E. Qawasmeh, and Z. A. Al-Sharif, "Utilizing debugging information of applications in memory forensics," *Journal of Universal Computer Science*, vol. 26, no. 7, 2020.
[2] D. Chakraborty, C. M. López, and P. Sarkar, "Disk Encryption: Do We Need To Preserve Length?," *Journal of Cryptographic Engineering*, vol. 8, no. 1, 2018, doi: 10.1007/s13389-016-0147-0.
[3] H. Alamsyah, R. -, and A. al Akbar, "Analisa Keamanan Jaringan Menggunakan Network Intrusion Detection and Prevention System," *JOINTECS (Journal of Information Technology and Computer Science)*, vol. 5, no. 1, 2020, doi: 10.31328/jointecs.v5i1.1240.
[4] C. Meijer and B. van Gastel, "Self-encrypting deception: Weaknesses in the encryption of solid state drives," in *Proceedings - IEEE Symposium on Security and Privacy*, 2019, vol. 2019-May, doi: 10.1109/SP.2019.00088.
[5] S. M. Pg Scholar and M. Krishnan, "Forensic Recovery of Fully Encrypted Volume," *International Journal of Computer Applications*, 2014.
[6] C. Tan, L. Zhang, and L. Bao, "A Deep Exploration of BitLocker Encryption and Security Analysis," in *International Conference on Communication Technology Proceedings, ICCT*, 2020, vol. 2020-October, doi: 10.1109/ICCT50939.2020.9295908.
[7] L. Zhang, X. Deng, and C. Tan, "An extensive analysis of truecrypt encryption forensics," 2019, doi: 10.1145/3331453.3361328.
[8] F. Franzen, M. Andreas, and M. Huber, "FridgeLock: Preventing Data Theft on Suspended Linux with Usable Memory Encryption," 2020, doi: 10.1145/3374664.3375747.
[9] Š. Balogh and M. Pondelik, "Capturing encryption keys for digital analysis," 2011, doi: 10.1109/IDAACS.2011.6072872.
[10] Y. Hu, J. C. S. Lui, W. Hu, X. Ma, J. Li, and X. Liang, "Taming energy cost of disk encryption software on data-intensive mobile devices," *Future Generation Computer Systems*, vol. 107, 2020, doi: 10.1016/j.future.2017.09.025.
[11] A. Visconti, O. Mosnáček, M. Brož, and V. Matyáš, "Examining PBKDF2 Security Margin—Case Study Of LUKS," *Journal of Information Security and Applications*, 2019, doi: 10.1016/j.jisa.2019.03.016.
[12] J. A. Lapso, G. L. Peterson, and J. S. Okolica, "Whitelisting system state in windows forensic memory visualizations," *Digital Investigation*, vol. 20, 2017, doi: 10.1016/j.diin.2016.12.002.
[13] F. Block and A. Dewald, "Windows Memory Forensics: Detecting (Un)Intentionally Hidden Injected Code by Examining Page Table Entries," *Digital Investigation*, vol. 29, 2019, doi: 10.1016/j.diin.2019.04.008.
[14] N. Lewis, A. Case, A. Ali-Gombe, and G. G. Richard, "Memory forensics and the windows subsystem for linux," 2018, doi: 10.1016/j.diin.2018.04.018.
[15] X. Zhang, L. Hu, S. Song, Z. Xie, X. Meng, and K. Zhao, "Windows volatile memory forensics based on correlation analysis," *Journal of Networks*, vol. 9, no. 3, 2014, doi: 10.4304/jnw.9.3.645-652.
[16] I. Riadi and I. M. Nasrulloh, "Analisis Forensik Solid State Drive (Ssd) Menggunakan Framework Grr Rapid Response Forensic Analysis Of Solid State Drives (Ssd) Using The Grr Rapid Response Framework," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 6, no. 5, 2019.