e-ISSN 2549-7472
Edumatic: Jurnal Pendidikan Informatika
Vol. 6 No. 2, Desember, 2022, **Hal. 334-343**
DOI: 10.29408/edumatic.v6i2.6466

Network Forensic Serangan DoS pada Jaringan *Cloud* berdasarkan *Generic Framework for Network Forensics* (GFNF)

Tri Hesti Damayanti [1,*], Ira Rosianal Hikmah [1]

[1] Program Studi Rekayasa Keamanan Siber, Politeknik Siber dan Sandi Negara, Indonesia
[*] Correspondence: tri.hesti@student.poltekssn.ac.id

Received: 27 Agustus 2022 | Revised: 29 Agustus 2022 | Accepted: 7 November 2022 | Published: 20 Desember 2022

Abstrak
Meningkatnya serangan pada jaringan yang menjalankan layanan *cloud computing* dapat menyebabkan berbagai kerugian mulai dari tidak dapat diaksesnya layanan hingga hilangnya kepercayaan dari para pengguna. Owncloud merupakan salah satu implementasi *cloud* pada jaringan dengan jumlah lebih dari 200 juta pengguna. Penelitian ini bertujuan untuk menemukan bukti digital dari serangan DoS berupa SSH *brute force*, SYN *flood*, *ping of death*, dan *port scanning* pada jaringan Owncloud berupa IP penyerang, waktu terjadinya serangan, jenis serangan, serta informasi penggunaan *resource* CPU dan RAM. Penelitian ini menggunakan *tools* Wireshark dan Snort dalam menganalisis jaringan dan menggunakan metode *Generic Framework for Network Forensic* (GFNF) sebagai kerangka kerja selama proses simulasi hingga presentasi hasil temuan barang bukti. Simulasi serangan dilakukan masing-masing selama 1 menit dengan 30 kali percobaan. Hasil dari penelitian ini ditemukan bukti digital berupa IP penyerang, waktu terjadinya serangan, jenis serangan, serta informasi penggunaan *resource* CPU dan RAM yang meningkat ketika terjadi serangan. Hasil dari bukti digital berupa IP penyerang, waktu terjadinya serangan, dan jenis serangan yang ditemukan tersebut divisualisasikan dalam bentuk tabel temuan dan disajikan pada *dashboard* ELK Stack.

Kata kunci: forensik jaringan; bukti digital; gfnf; snort; elk stack

Abstract
The attacks on cloud-based networks have increased and could lead to various disadvantages such as the inaccessibility of services until the loss of user's trust. Owncloud is one cloud implementation that runs on a network with more than 200 million users. The aims of these researches are to find digital evidence from DoS attacks. Some DoS attacks are SSH brute force, SYN flood, ping of death, and port scanning on the Owncloud network and then finding the digital evidence such as the attacker's IP, time occurred of the attack, types of the attack, also the resource usage of CPU and RAM. This research uses Wireshark and Snort tools to analyze the network and the method of Generic Framework for Network Forensic (GFNF) as a framework during the simulation process until performing the evidence. The simulation was carried out for 1 minute with 30 trials for each attack. The results of this study found the attacker's IP, time of the attack occurred, types of attack, and also the increase of the resource usage on CPU and RAM when an attack occurred. The found of results digital evidence such as the attacker's IP, the time occurred of attack, and the types of attack were visualized as a table and presented on the ELK Stack dashboard.

Keywords: *network forensic; digital evidence; gfnf; snort; elk stack*

PENDAHULUAN

Kasus serangan siber pada jaringan semakin meningkat (Satria et al., 2021). Hal tersebut tidak menutup kemungkinan dapat terjadi pada layanan Owncloud (Abadi et al., 2018). Dilansir dari laman resmi Owncloud, kini layanan Owncloud telah digunakan dan

dipercaya oleh lebih dari 200 juta pengguna di seluruh dunia (Jupriyadi & Prabowo, 2017). Dilansir dari laman resmi Owncloud, kini layanan Owncloud telah digunakan dan dipercaya oleh lebih dari 200 juta pengguna di seluruh (Jupriyadi & Prabowo, 2017). Sistem informasi terpusat seperti pada *cloud computing*, sangat rawan terhadap berbagai ancaman keamanan jaringan seperti serangan DOS, SYN *flood*, maupun serangan lainnya (Sahren, 2021). Menurut Khalaf, et al. (2019) dan Lukman & Suci Melati (2020), serangan DoS yang sering dilakukan pada jaringan *cloud* diantaranya seperti UDP *flood*, SYN *flood*, *ping of death*, smurf, HDoS, dan XDoS. Berdasarkan masalah yang terdapat pada fakta-fakta tersebut, teori pada penelitian sebelumnya yang dilakukan oleh Pilli, et al. (2010) dan Fathoni, Fitriyani & Nurkahfi, (2016) menyatakan bahwa solusi untuk mengungkap masalah keamanan jaringan yang terjadi yaitu melalui forensik jaringan.

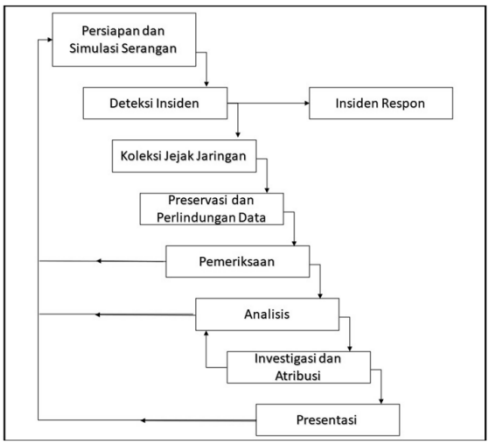
Forensik jaringan merupakan cabang dari forensik digital yang bekerja dengan cara menganalisa dan mengklasifikasikan urutan riwayat kejadian, syslog, *network traffic*, dan *attack behaviors* sesuai dengan serangan yang terjadi (Ridho et al., 2016). Berdasarkan metode *Generic Framework for Network Forensics* (GFNF), forensik jaringan terdiri dari sembilan tahapan yaitu tahap persiapan, deteksi insiden, penanganan insiden, koleksi jejak jaringan, preservasi dan perlindungan data, pemeriksaan, analisis, investigasi dan atribusi, dan tahapan terakhir yaitu presentasi bukti digital. Untuk menganalisa dan mengklasifikasikan urutan riwayat kejadian yang ada pada jaringan, diperlukan sistem pada sisi server yang dapat mencatat dan merekam setiap aktivitas yang berjalan salah satunya dengan menggunakan Snort (Fadilla et. al., 2022). Konsep dari tool Snort adalah apabila terdapat lalu lintas (*traffic*) yang bersifat anomali maka akan memunculkan *alert* atau notifikasi (Muhammad et al., 2013). *Tool* Snort yang digunakan untuk merekam aktivitas serangan pada server, akan menghasilkan sebuah data berupa log (Suharmanto et al., (2018). Namun, hasil log tersebut hanya berupa *command* yang sangat sederhana yang ditampilkan pada terminal linux (Riadi et al., 2020). Oleh karena itu, diperlukan *tool* yang dapat memvisualisasikan log tersebut. Salah satu yang dapat digunakan adalah dengan menggunakan ELK Stack (Elasticsearch, Logstash, dan Kibana). Secara singkat, Logstash akan memproses log dan membuat index log. Sementara itu, Elasticsearch akan menyimpan semua log yang masuk dari IDS seperti *tool* Snort dan kemudian log tersebut divisualisasikan pada *web interface* dalam grafik yang diinginkan melalui Kibana (Sholihah et al., 2020).

Pada penelitian sebelumnya yang dilakukan oleh (Helmi et al., 2019) telah melakukan forensik jaringan, namun belum menemukan adanya penggunaan *resource* CPU dan RAM yang dapat memperkuat temuan sehingga perlu dilakukan pengembangan dari penelitian tersebut. GFNF dapat dijadikan sebagai metode terobosan untuk dapat menemukan bukti digital yang dapat dipertanggung jawabkan secara hukum. Sehingga, pada penelitian ini akan menggabungkan antara penelitian sebelumnya agar menghasilkan bukti digital yang lebih komprehensif dan dapat menampilkannya pada sebuah *dashboard* sehingga mudah dipahami. Penelitian ini bertujuan untuk menemukan bukti digital pada serangan *SSH brute force, SYN flood, ping of death,* dan *port scanning* ke server Owncloud. *Network forensic* yang dijalankan berpedoman pada kerangka kerja *Generic Framework for Network Forensics* (GFNF). Hasil yang didapatkan dari semua skema pengujian serangan dianalisis untuk mendapatkan bukti berupa waktu serangan, penyerang (dalam bentuk IP), urutan rekonstruksi kejadian, jenis serangan, dan informasi penggunaan *resource* CPU dan RAM yang dapat memperkuat untuk dijadikan sebagai bukti digital.

METODE

Penelitian yang dilakukan termasuk penelitian kualitatif. Penelitian kualitatif menurut Rakhmat et al. (2021) dan Firmansyah et al. (2021) merupakan penelitian yang berfokus pada tujuan mengeksplorasi atau menjelaskan sesuatu berdasarkan data-data yang telah terhimpun.

Pengambilan data pada penelitian ini diperoleh dari hasil simulasi serangan SSH *brute force*, SYN *flood*, *ping of death*, dan *port scanning* ke jaringan Owncloud (Khalaf, et al., 2019; Yazhini, 2020). Setelah itu dilakukan pengamatan sesuai tahapan yang ada pada *Generic Framework for Network Forensics* (GFNF) seperti yang ditunjukkan pada gambar 1.



Gambar 1. Desain penelitian

Gambar 1 mendeskripsikan tahapan penelitian yang akan dilakukan. Pada tahap persiapan dan simulasi serangan dilakukan dengan menyiapkan segala kebutuhan penelitian, dilanjutkan dengan tahap deteksi insiden untuk mengetahui kemampuan *tool* yang sudah dipersiapkan dalam mendeteksi adanya serangan (Pichan et al., 2015). Tahap ketiga merupakan tahapan respon insiden yang dilakukan dengan mematikan sementara *service* yang berjalan agar serangan tidak meluas, dan tahapan koleksi jejak jaringan juga dilakukan ketika serangan terjadi. Tahapan preservasi dan perlindungan data merupakan tahapan yang dilakukan untuk memastikan integritas bukti digital tetap terjaga. Tahap pemeriksaan dan analisis diimplementasikan dengan melakukan pengamatan secara detail pada hasil log dan file pcap yang ada. Pada tahap investigasi dan atribusi sudah dapat mengidentifikasi bukti apa yang ditemukan, serta tahap presentasi yang digunakan untuk menyajikan bukti digital yang diperoleh. Masing-masing tahapan dilakukan secara urut mulai dari menyiapkan segala kebutuhan penelitian, hingga penyajian bukti temuan dalam tabel dan *dashboard* ELK Stack (Riadi et al., 2021).

HASIL DAN PEMBAHASAN

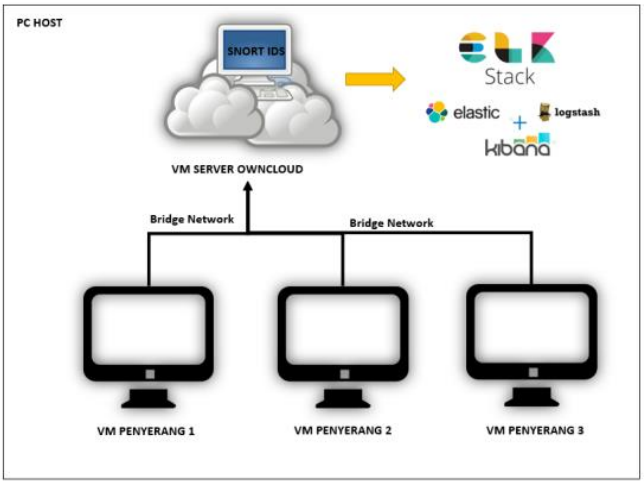
Hasil

Hasil dari tahapan ini seluruh kebutuhan pengujian sudah terpasang serta simulasi serangan dapat dijalankan. Gambar 2 menunjukkan hasil atau *output* dari tahap persiapan yaitu Owncloud sudah terpasang pada sisi server. Adapun simulasi serangan yang dilakukan menggunakan topologi jaringan lokal dengan dokumentasi yang ditunjukkan pada gambar 3.

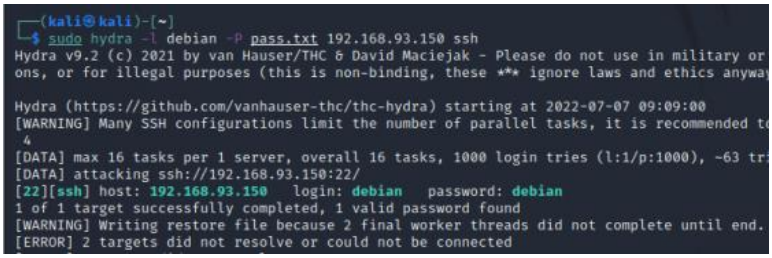
Skenario serangan SSH *brute force*, SYN *flood*, *ping of death*, dan *port scanning* yang dijalankan untuk selanjutnya dianalisis dilakukan masing-masing selama 1 menit dengan 30 kali percobaan. Dokumentasi simulasi serangan yang diimplementasikan pada server disajikan pada gambar 4. Hasil ini menjukkan bahwa serangan berhasil dilakukan ke target yaitu server Owncloud dengan IP *address* 192.168.93.150.

Pada tahapan deteksi insiden, *rules* Snort yang diatur berhasil mendeteksi adanya serangan SSH *brute force*. Pada gambar 5 mendeskripsikan bahwa dari keseluruhan percobaan serangan yang dilakukan snort berhasil mendeteksi keseluruhan percobaan tersebut dengan akurasi 100%. Akurasi tersebut didapatkan dari 30 kali percobaan serangan, seluruhnya dapat ditangkap oleh Snort.

Gambar 2. Tampilan *login* owncloud pada virtualbox debian



Gambar 3. Topologi pada lingkungan penelitian



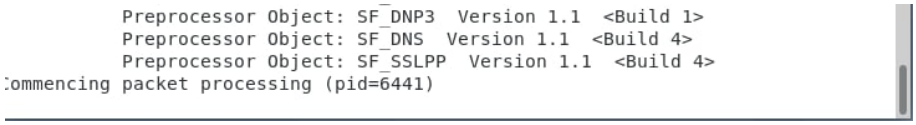
Gambar 4. Simulasi SSH *brute force*

```
05/31-21:11:34.561166  [**] [1:1000020:1] Potential SSH Brute Force Attack [**] [Cl
assification: Attempted Denial of Service] [Priority: 2] {TCP} 192.168.93.184:45826
 -> 192.168.93.150:22
05/31-21:11:34.561166  [**] [1:1000020:1] Potential SSH Brute Force Attack [**] [Cl
assification: Attempted Denial of Service] [Priority: 2] {TCP} 192.168.93.184:45810
 -> 192.168.93.150:22
```

Gambar 5. Hasil deteksi insiden serangan SSH *brute force*

Penelitian ini tidak berfokus pada tahapan respon insiden. Namun, pada pengujian ini tetap dilakukan tindakan respon insiden dengan mematikan sementara jaringan Owncloud. Ketika Snort menyimpan log, terjadi peningkatan *resource* penyimpanan yang besar hingga memenuhi *disk* pada server Owncloud. Sehingga, dilakukan tindakan respon insiden dengan memindahkan log tersebut ke komputer *host*. Pengumpulan koleksi jejak jaringan dilakukan untuk mendapatkan jejak jaringan. Gambar 6 menunjukkan tahapan pengumpulan jejak jaringan menggunakan Snort. Gambar 6 merupakan rangkaian dari Snort saat melakukan pengumpulan *traffic* dari jaringan Owncloud yang sedang berjalan. Koleksi jejak jaringan dilakukan pada setiap akan melakukan serangan dengan melakukan *capture* menggunakan

tool Snort dengan *rules* yang dipakai adalah yang sesuai dengan serangan yang hendak dilakukan.



Gambar 6. Implementasi pengumpulan *traffic* serangan

Preservasi dan perlindungan data dilakukan *cloning* dan disalin ke perangkat *backup* dalam hal ini disalin ke *directory* perangkat *host.* Hasil *cloning* selanjutnya dilakukan *hashing* untuk memastikan bahwa selama proses *cloning* tidak terdapat data yang hilang. Gambar 7 merupakan dokumentasi hasil *hash file* hasil *cloning* dan membuktikan bahwa file yang akan dilakukan analisis merupakan file yang sama dengan file asli dari barang bukti digital yang didapatkan. Hal tersebut terlihat dari adanya nilai hash yang sama antara file asli dan file hasil *cloning*.

Filename	MD5	SHA1
SKENARIO 1 s...	723679989472f0b5a41e3ee564b15bf5	dbedb0ae717c20c021d161f8b4dda81ff8aba...
SKENARIO 1 s...	723679989472f0b5a41e3ee564b15bf5	dbedb0ae717c20c021d161f8b4dda81ff8aba...
SKENARIO 2 s...	c167c2072f72e8b3fcc7af7f64fbe035	c7ba845b457a4ae33f7c1039131b582b69def...
SKENARIO 2 s...	c167c2072f72e8b3fcc7af7f64fbe035	c7ba845b457a4ae33f7c1039131b582b69def...
SKENARIO 3 s...	ca1e45d97e5801be136042295e5720b1	777b4c82ce5368f6d06a9fe15c0b34a94173e5...
SKENARIO 3 s...	ca1e45d97e5801be136042295e5720b1	777b4c82ce5368f6d06a9fe15c0b34a94173e5...
SKENARIO 4 s...	2def7872c047ca2cd2ea165c4725d8bb	dc3cd3ee6076b0ac0acbd8d019009b9ff998b...
SKENARIO 4 s...	2def7872c047ca2cd2ea165c4725d8bb	dc3cd3ee6076b0ac0acbd8d019009b9ff998b...

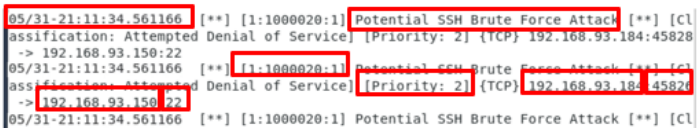
Gambar 7. Hasil persamaan nilai *hash* file *cloning* dan file asli

Selanjutnya, data yang diperoleh dari Snort disatukan dalam satu folder agar infomasi tidak terpisah dengan data lain. Pemeriksaan dilakukan dengan melihat isi dari paket Wireshark hasil log. Contoh *traffic* pada saat terjadi serangan SSH *brute force* yang terlihat pada gambar 8. Dari pemeriksaan ini didapatkan informasi bahwa penyerang melakukan serangan dengan destinasi *port* 22 dimana *port* ini merupakan *port* untuk melakukan SSH. Hal ini juga sama dilakukan pemeriksaan ketika ketiga serangan lainnya terjadi.

No.	Time	Source	Destination	Protocol	Lengt	Delta_time_disp	Info
1	0.000000	192.168.93.184	192.168.93.150	TCP	66	0.000000	45772 → 22 [ACK] Seq=1 Ack=1 Win=502 Len=0
2	0.061034	192.168.93.184	192.168.93.150	TCP	66	0.061034	45776 → 22 [ACK] Seq=1 Ack=1 Win=502 Len=0
3	0.074945	192.168.93.184	192.168.93.150	TCP	66	0.013911	45752 → 22 [ACK] Seq=1 Ack=1 Win=501 Len=0
4	0.139068	192.168.93.184	192.168.93.150	TCP	66	0.064123	45758 → 22 [ACK] Seq=1 Ack=1 Win=501 Len=0

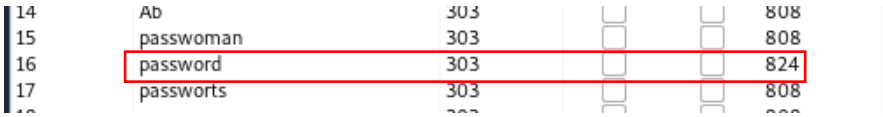
Gambar 8. Hasil pemeriksaan log snort pada serangan SSH *brute force*

Analisis dilakukan dengan pengamatan secara mendalam pada hasil log Snort yang kemudian dibantu dengan dibaca menggunakan Wireshark. gambar 9 menunjukkan informasi penting yang dapat dijadikan sebagai bukti telah terjadi serangan pada server Owncloud dimana penyerang menggunakan IP 192.168.93.184 dan serangan tersebut dilakukan pada 31 Mei 2022 pukul 21:11:34 hingga 1 Juni 2022 pukul 02:25:20 *SE Asia Standard Time*. Pada penelitian ini Snort mampu mendeteksi adanya pergerakan *brute force* yang menyerang langsung ke halaman *login*. Hal ini digunakan sebagai penguat bukti sejauh mana pelaku melakukan serangan. Serangan dilakukan secara terstruktur terbukti dengan adanya percobaan kata sandi yang dimasukkan urut berdasarkan abjad mulai dari huruf awal aa hingga terakhir zaeland seperti membentuk *wordlist.*



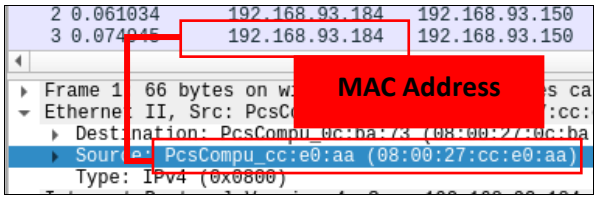
Gambar 9. Informasi penting hasil *alert* serangan SSH *brute force*

Analisis dilanjutkan pada *dashboard* server admin Owncloud yang ternyata dari Owncloud sendiri sudah memiliki fitur yang dapat merekam log pergerakan siapa saja yang mengakses Owncloud tersebut. Ketika dilakukan serangan *brute force* menggunakan Burpsuite penyerang berhasil mendapatkan kredensial yang benar seperti pada gambar 10.



Gambar 10. Penyerang mendapatkan kredensial masuk owncloud

Dilakukan analisis pada log Owncloud dan hasil yang diperoleh adalah benar bahwa pada Owncloud juga mendeteksi adanya percobaan masuk gagal yang sangat banyak. Analisis dilakukan menggunakan informasi dari hasil Wireshark ditemukan informasi *mac address* dari penyerang dengan dokumentasi yang ditunjukkan pada Gambar 11. Informasi mengenai *mac address* pelaku sangat penting dalam forensik guna melakukan *blocking* terhadap sumber tersebut. Selain analisis pada paket log hasil dari *tool* Snort, analisis juga dilakukan pemantauan terhadap penggunaan *resource* CPU dan RAM ketika terjadi serangan dan ketika kondisi normal. Secara keseluruhan perbedaan dari tiap serangan secara rinci dijelaskan pada Tabel 1.

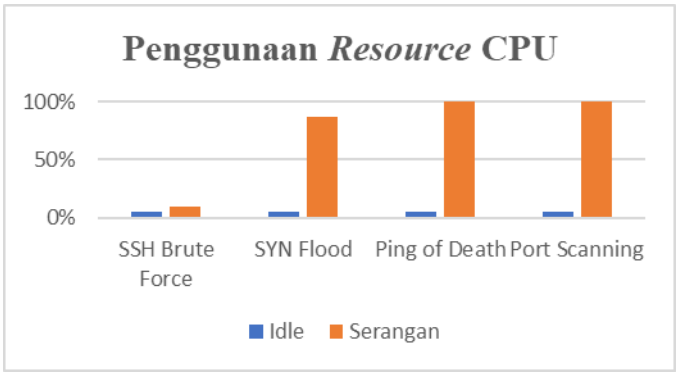


Gambar 11. *Mac address* pelaku serangan ssh *brute force*

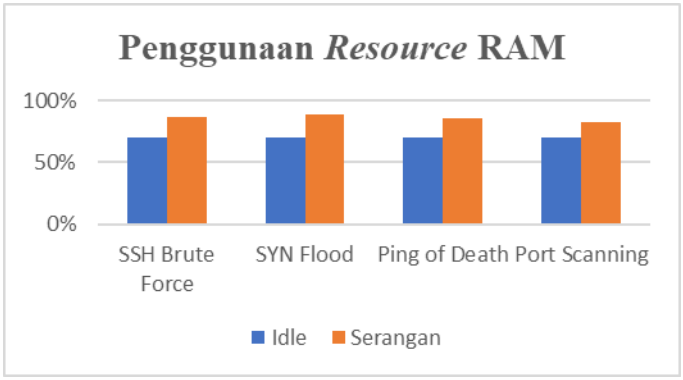
Tabel 1. Hasil Pengujian Penggunaan *Resource* CPU dan RAM.

Kondisi	**Penggunaan *Resource***		**Keterangan**
	CPU	RAM	Nilai CPU diambil berdasarkan rata-rata dari tiap serangan selama 1 menit
Idle	5,1%	70,52%	
Serangan SSH *brute force*	9,95%	86,64%	
Serangan SYN *flood*	86,45%	88,41%	
Serangan *ping of death*	100%	86,14%	
Serangan *port scanning*	100%	82,61%	

Berdasarkan tabel 1 dapat dilihat bahwa ketika terjadi serangan DoS penggunaan *resource* baik CPU dan RAM meningkat dengan tingkat kenaikan tertinggi terjadi pada serangan *ping of death* dan *port scanning* hingga menyebabkan layanan Owncloud tidak dapat diakses dengan baik. Gambar 12 dan 13 menggambarkan hasil visualisasi penggunaan *resource* CPU dan RAM secara keseluran dari serangan yang telah diujikan.



Gambar 12. Penggunaan *resource* cpu secara keseluruhan



Gambar 13. Penggunaan *resource* ram secara keseluruhan

Tabel 2. Hasil investigasi dan atribusi

No	Jenis Informasi Temuan	Keterangan
1.	Serangan yang terjadi pada server Owncloud	SSH *brute force*, SYN *flood*, *ping of death*, *port scanning*, dan *brute force login cloud.*
2.	*Tool* Wireshark yang digunakan untuk membaca hasil log dari Snort	Diperoleh informasi adanya paket ICMP, SYN, HTTP, dan TCP yang membanjiri server Owncloud hingga memenuhi *disk* atau penyimpanan.
3.	*Port source* penyerang	50436, 37278, 2518, 56812
4.	Kondisi rata-rata CPU dan RAM normal (5,1%) dan (70,52%). Saat terjadi serangan yaitu:	CPU: SSH *brute force* (9.95%), SYN *flood* (86,45%), *ping of death* (100%), *port scaning* (100%) RAM: SSH *brute force* (86,64%), SYN *flood* (88,41%), *ping of death* (86,14%), *port scaning* (82,61%)
5.	IP *address* penyerang	192.168.93.184, 192.168.93.87, 192.168.93.95
6.	*MAC address* penyerang	PcsCompu_cc:e0:aa (08:00:27:0C:BA:AA) PcsCompu_11:7c:5f (08:00:27:11:7C:5F) PcsCompu_a7:21:73 (08:00:27:A7:21:73)
7.	Rekonstruksi serangan terjadi	*Scanning* -> *port* 22 terbuka -> SSH *brute force* terjadi -> tidak ada batasan percobaan masuk -> *login brute force cloud* terjadi -> paket DOS terus dikirimkan oleh penyerang -> *disk* penuh dan server *down.*

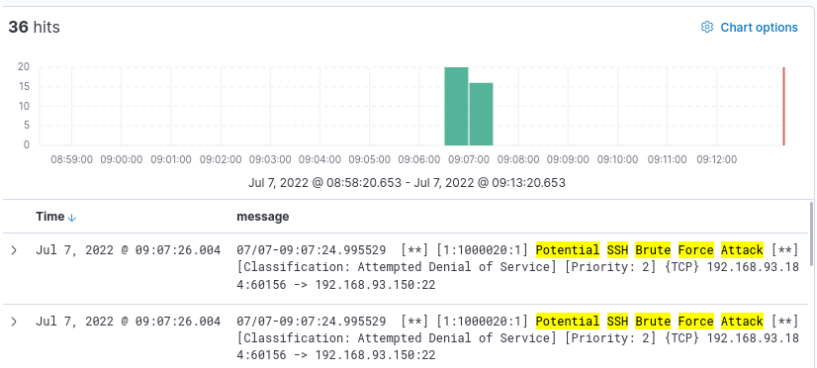
Kondisi *resource* yang ditunjukkan Gambar 12 dan 13 tersebut dijadikan sebagai bukti kuat bahwa server yang menjalankan layanan Owncloud yang diujikan telah diserang oleh suatu entitas tertentu.

Hasil investigasi dan atribusi secara keseluruhan dirincikan pada tabel 2. Hasil ini menjukkan bahwa rincian temuan bukti digital yang telah ditemukan dari hasil analisis yang telah dilakukan. Masing-masing temuan dituliskan dalam tabel tersebut secara jelas dijabarkan dalam kolom keterangan. Hasil dari forensik yang dilakukan dipresentasikan ke dalam tabel temuan yang berisi kolom *timestamp, source address, destination address, source port, destination port,* dan *event name* yang ditunjukkan pada tabel 3.

Tabel 3. Tabel hasil temuan penelitian

Timestamp	**IP *Source***	**IP *Destination***	***Port Src***	***Port Dst***	***Event Name***
05/31-21:11:34	192.168.93.184	192.168.93.150	45828	22	Potential SSH Brute Force
05/16-11:48:53	192.168.93.87	192.168.93.150	2027	80	SYN Flood Detected
05/16-11:24:51	192.168.93.95	192.168.93.150	-	-	Ping of Death Detected
07/05-03:44:37	192.168.93.87	192.168.93.150	35680	80	Port Scanning Detected

Selain menggunakan tabel temuan, hasil temuan bahwa jaringan yang menjalankan Owncloud terkena serangan disajikan dalam *dashboard* ELK. Visualisasi juga dapat ditemukan pada kasus serangan lainnya sesuai dengan log snort yang berjalan secara *real time*. Dokumentasi temuan untuk serangan SSH *brute force* digambarkan pada gambar 14 menunjukkan bahwa visualisasi dari log yang dihasilkan dari *tool* Snort berhasil masuk ke *dashboard* ELK Stack yang dibangun.



Gambar 14. Temuan serangan SSH *brute force* pada ELK server

Pembahasan

Berdasarkan hasil log Snort yang didapatkan, temuan kami memperoleh informasi penting bahwa penyerang menggunakan IP 192.168.93.184 dimana apabila dilihat menggunakan *range* dari *subnet* dengan server Owncloud penyerang masih dalam satu subnet yang sama. Artinya penyerang tersebut masih berada pada satu jaringan yang sama dengan server *cloud*. Serangan terjadi pada rentang 31 Mei 2022 pukul 21:11:34 *SE Asia Standard Time* hingga 1 Juni 2022 pukul 02:25:20 *SE Asia Standard Time.* Jenis serangan yang

ditemukan didapatkan dari hasil *alert* dengan notifikasi pada Snort dan ELK Stack dengan pesan telah terjadi serangan SSH *brute force*, SYN *flood*, *ping of death*, *port scanning*, dan *brute force login cloud*. Validasi log dibuktikan dari pcap Wireshark dan menghasilkan *output* yang sama dengan hasil log dari Snort yang menemukan adanya paket ICMP, SYN, HTTP, dan TCP yang membanjiri server Owncloud hingga memenuhi *disk* atau penyimpanan.

Kondisi rata-rata CPU pada keadaan normal berada pada angka 5.1% dan mengalami peningkatan tertinggi pada saat terjadi serangan *ping of death* dan *port scanning*. Sedangkan penggunaan memori pada kondisi normal berada pada angka 70,52% dan mengalami kenaikan tertinggi ketika terjadi serangan SYN *flood.* Pada *dashboard* ELK Stack dapat terlihat diagram yang memvisualisasikan *count* serangan atau jumlah *drop event* serangan yang terjadi pada tiap *timestamp* yang ditunjukkan. *Dashboard* yang dibuat berhasil menunjukkan bahwa telah terjadi serangan SSH *brute force*, SYN *flood*, *ping of death, port scanning*, dan *brute force login cloud* lengkap dengan *timestamp* waktunya.

Penelitian sebelumnya yang relevan dengan penelitian ini dengan objek yang sama dilakukan oleh Fadilla et. al. (2022), hanya sebatas melakukan analisis pada log hasil serangan pada server Owncloud dan tidak melakukan analisis pcap. Penelitian lain dengan objek yang sama dilakukan oleh Riadi et al. (2021) dan hanya menemukan bukti digital berupa *timestamp, port destination*, dan IP penyerang, namun tidak menemukan bukti *resource* penggunaan CPU dan RAM. Berdasarkan penelitian-penelitian sebelumnya yang telah disebutkan dapat disimpulkan bahwa penggunaan metode GFNF dapat memberikan solusi yang tepat dalam melakukan forensik jaringan, dan dapat diterapkan pada kasus-kasus serupa pada layanan lainnya.

SIMPULAN

Bukti digital dari kasus serangan DoS pada Owncloud yang berhasil ditemukan pada penelitian ini diantaranya berupa data waktu terjadinya serangan, penyerang dalam bentuk IP, *mac address* penyerang, sumber *port*, tujuan *port* serangan, *payload*, dan informasi penggunaan *resource* CPU dan RAM yang meningkat ketika serangan terjadi. Hasil implementasi ELK Stack yang dirancang dapat memvisualisasikan bukti temuan yang didapatkan sehingga mempermudah proses respon insiden yang ada pada salah satu tahapan forensik jaringan.

REFERENSI

Abadi, J., Arianti, B. D. D., & Wirasasmita, R. H. (2018). Pengembangan Media Lembar Kerja Siswa (LKS) Berbasis Web Pada Mata Pelajaran Jaringan Dasar. *EDUMATIC: Jurnal Pendidikan Informatika*, *2*(1), 42–51. https://doi.org/10.29408/edumatic.v2i1.939

Fathoni, W., Fitriyani, F., & Nurkahfi, G. N. (2016). Deteksi Penyusupan Pada Jaringan Komputer Menggunana IDS Snort. *eProceedings of Engineering*, *3*(1), 1169-1172.

Firmansyah, M., Masrun, M., & Yudha S, I. D. K. (2021). Esensi Perbedaan Metode Kualitatif Dan Kuantitatif. *Elastisitas - Jurnal Ekonomi Pembangunan*, *3*(2), 156–159. https://doi.org/10.29303/e-jep.v3i2.46

Fadilla, M. K., Sugiantoro, B., & Prayudi, Y. (2022). Membangun Framework Konseptual Terintegrasi Menggunakan Metode Composite Logic untuk Cloud Forensic Readiness pada Organisasi. *Jurnal Media Informatika Budidarma*, *6*(1), 144-153. https://doi.org/10.30865/mib.v6i1.3427

Helmi, I., Widiyasono, N., & Gunawan, R. (2019). Simulasi Analisis Bukti Digital Pada Layanan Cloud Computing Menggunakan Metode NIST 800-86. *Jurnal Media Informatika Budidarma*, *3*(3), 217-224. https://doi.org/10.30865/mib.v3i3.1193

Jupriyadi, & Prabowo, R. (2017). Implementasi ownCloud Sebagai Private Storage Berbasis

Web pada Perguruan Tinggi XYZ. *Seminar Nasional Sains Dan Teknologi*, *2*(1), 1–5.

Khalaf, B. A., Mostafa, S. A., Mustapha, A., Ismaila, A., Mahmoud, M. A., Jubaira, M. A., & Hassan, M. H. (2019). A simulation study of syn flood attack in cloud computing environment. *AUS journal*, *26*(1), 188-197.

Lukman, & Suci Melati. (2020). Analisis Perbandingan Kinerja Snort Dan Suricata Sebagai Intrusion Detection System Dalam Mendeteksi Serangan Syn Flood Pada Web Server Apache. *Jurnal Teknologi Informasi*, *XV*(1907–2430), 1–15. https://doi.org/http://dx.doi.org/10.35842/jtir.v15i2.343

Muhammad, R. M., Irawati, I. D., & Iqbal, M. (2013). Implementasi Sistem Keamanan Jaringan Lokal Menggunakan Honeypot Dionaea, dan IDS, Serta Analisis Malware. *Jurnal Elektro Telekomunikasi Terapan*, *7*(3), 1–7.

Pichan, A., Lazarescu, M., & Soh, S. T. (2015). Cloud forensics: Technical challenges, solutions and comparative analysis. *Digital Investigation*, *13*(2), 38–57. https://doi.org/10.1016/j.diin.2015.03.002

Pilli, E. S., Joshi, R. C., & Niyogi, R. (2010). A framework for network forensic analysis. In *International Conference on Advances in Information and Communication Technologies* (pp. 142-147). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15766-0_21

Rakhmat, E., Dwiyatno, S., Sulistiyon, S., Irawan, A., & Setiawan, F. (2021). Pemanfaatan Aplikasi Owncloud Pada Sistem Keamanan Cloud Computing. *Jurnal Sistem Informasi Dan Informatika (Simika)*, *4*(2), 146–155. https://doi.org/10.47080/simika.v4i2.1454

Riadi, I., Fadlil, A., & Hafizh, M. N. (2020). Analisis Bukti Serangan Address Resolution Protocol Spoofing menggunakan Metode National Institute of Standard Technology. *Edumatic Jurnal Pendidikan Informatika*, *4*(1), 1–11. https://doi.org/10.29408/edumatic.v4i1.2046

Riadi, I., Yudhana, A., & Al Barra, M. (2021). Forensik Mobile pada Layanan Media Sosial LinkedIn. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *6*(1), 9-20. https://doi.org/10.14421/jiska.2021.61-02

Ridho, F., Yudhana, A., & Riadi, I. (2016). *Analisis Forensik Router Untuk Mendeteksi Serangan Distributed Danial of Service (DDoS) Secara Real Time*. *2*(1), 111–116.

Sahren. (2021). Implementasi Teknologi Firewall Sebagai Keamanan Server dari SYN Flood Attack. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, *7*(2), 159–164. https://doi.org/10.33330/jurteksi.v7i2.933

Satria, A. F., Adam, R. I., & Carudin, C. (2021). Analisis Digital Watermarking untuk Otentikasi pada Citra Manipulasi Menggunakan Metode Least Significant Bit. *Edumatic: Jurnal Pendidikan Informatika*, *5*(2), 204–213. https://doi.org/10.29408/edumatic.v5i2.3901

Sholihah, W., Pripambudi, S., & Mardiyono, A. (2020). Log event management server menggunakan elastic search logstash kibana (elk stack). *JTIM: Jurnal Teknologi Informasi dan Multimedia*, *2*(1), 12-20. https://doi.org/10.35746/jtim.v2i1.79

Suharmanto, A. Y., Lumenta, A. S., & Najoan, X. B. (2018). Analisa Keamanan Jaringan Wireless Di Universitas Sam Ratulangi. *Jurnal Teknik Informatika*, *13*(3), 1-10.

Yazhini, P. (2020). The Survey DDoS Attack Prevention and Defense Technique. *International Journal of Innovative Science and Research Technology*, *5*(2), 65–58.