Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 5 Iss. 1 January 2025, pp: 159-167
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Implementation of Rule Based Method in Detecting Brute Force Attacks on Owncloud

Implementasi Metode Rule Based dalam Mendeteksi Serangan Brute Force pada Owncloud

Khazin Mubarok[1*], Moh. Ali Romli[2]

[1,2]Program Studi Informatika, Universitas Teknologi Yogyakarta, Indonesia

E-Mail: [1]khazinnismo@gmail.com, [2]ali.romli@uty.ac.id

Received Sep 8th 2024; Revised Nov 11th 2024; Accepted Nov 15th 2024; Available Online Dec 5th 2024
Corresponding Author: Khazin Mubarok

Abstract

Owncloud is a cloud-based open-source storage medium that offers the best server for data storage. However, data security is a primary concern when using Owncloud's server. As the server administrator, one cannot guarantee the security of data stored on the Owncloud server. To protect and detect attacks on Owncloud, network analysis is required to monitor the pattern of brute-force attacks on the server. Forensic analysis is also necessary to determine whether an intruder has launched an attack against the Owncloud server. The required software includes Snort, a packet sniffing tool, and Wireshark, a packet capturing tool based on Intrusion Detection System (IDS), to test attacks. A rule-based method is employed in testing brute-force attacks. This method involves pre-defined rules for identifying suspicious patterns of activity. This includes examining login patterns, check the number of failed logins from the same IP address over a specified period of time.

Keyword: Attack Detection, Brute Force, Owncloud, Rule Based Method

Abstrak

Owncloud merupakan sebuah media penyimpanan awan *(cloud storage)* yang bersifat *open source*. *Owncloud* merupakan *server* terbaik sebagai tempat penyimpanan *data*. Tetapi, keamanan *data* juga menjadi perhatian utama dalam penggunaan media *server Owncloud*. Pengelola *server Owncloud* tidak dapat menjamin keamanan *data* pada *server Owncloud* yang dikelolanya. Dalam melindungi dan mendeteksi serangan pada *Owncloud*, diperlukan analisis jaringan untuk mengamati pola serangan *brute force* pada *server Owncloud*. Analisis forensik juga diperlukan untuk mengetahui apakah ada penyerang *(intruder)* yang melakukan penyerangan terhadap *server Owncloud*. Analisis diperlukan *software Snort* sebagai paket *sniffing* dan *Wireshark* sebagai paket *capturing* yang berbasis *Intrusion Detection System* (IDS) dalam menguji serangan. Metode *rule based* dilakukan dalam pengujian serangan *brute force*. Penerapan metode *rule based* yang dilakukan melibatkan aturan penggunaan skenario yang sudah ditentukan sebelumnya untuk mengidentifikasi pola serangan mencurigakan. Hal ini mencakup pemeriksaan pola aktivitas *login*, seperti mengetahui jumlah upaya *login* gagal selama periode waktu tertentu dari IP yang sama.

Kata Kunci: *Brute Force*, Deteksi Serangan, Metode *Rule Based, Owncloud*

1. PENDAHULUAN

Owncloud merupakan media penyimpanan awan *(cloud storage)* yang bersifat *open-source* seperti Google Drive dan Dropbox. *Platform* ini diluncurkan pertama kali pada tahun 2010. *Owncloud* memungkinkan pengguna dapat menyimpan, mengelola, dan berbagi *file* antar pengguna dengan memberikan kendali penuh kepada penggunanya. Pengguna tidak bisa menjamin kemanan data pada *server Owncloud* yang dikelola. Saat kita menyimpan file yang akan ditransfer ke *database internet* seperti yang sering dilakukan saat ini dengan menyimpan data pada *email*, Google Drive, atau Dropbox, akan sering mengalami keterbatasan penyimpanan. Teknik ini sering disarankan untuk mengurangi kekhawatiran kita tentang kehilangan *data* dengan melakukan ini, kita dapat merasa tenang karena mengetahui bahwa kita memiliki cadangan data yang tersimpan di Dropbox, Google Drive, dan *email* [1][2]. Dikarenakan bersifat gratis dan bisa dikembangkan, kerentanan pada

Owncloud terhadap serangan *cyber* sangat tinggi. Untuk itu diperlukan analisis jaringan untuk mengamati pola serangan *brute force* pada *server Owncloud*.

Brute force merupakan jenis serangan yang masuk dalam sistem menggunakan cara yang mudah untuk menemukan solusi, yaitu dengan mencoba semua opsi yang tersedia, seperti kombinasi antara huruf, angka, dan simbol. Memakan waktu lebih lama untuk menemukan solusi jika semakin banyak pilihan yang harus dicoba. *Brute Force* adalah algoritma yang dapat menemukan semua solusi karena algoritma ini mencoba semua kemungkinan solusi yang ada dan mengandalkan faktor keberuntungan *(Lucky)* untuk menemukan solusinya, namun jika faktor keberuntungan *(Lucky)* tersebut tidak didapat maka algoritma ini adalah *worst-algorithm* [3]. Algoritma *brute force* memiliki kelebihan dan kekurangan, sama seperti algoritma lainnya. Meskipun algoritma *brute force* memiliki kelebihan karena dapat menangani hampir semua masalah, algoritma ini memiliki kekurangan karena tidak seefektif pendekatan pemecahan masalah lainnya [4]. Dalam konteks *Owncloud*, serangan *brute force* dapat mengancam keamanan pengguna dengan mencoba menebak kata sandi pengguna atau *administrator* yang dapat mengakibatkan pelaku dapat mengakses *file* sensitif. Oleh karena itu, penting menggunakan teknik keamanan seperti waktu tunda *login* ulang ketika batas kegagalan *login* di *server* tercapai dan *Completely Automated Public Turing test to tell Computer and Human Apart* (CAPTCHA), untuk mencegah keberhasilan serangan *brute force.*

Pada penelitian sebelumnya [5], Membahas tentang Penerapan Metode *Rule Based* dalam Mendeteksi Serangan *Multi Attack* pada *Network Attached Storage*. Dalam penelitian ini penulis menggunakan berbagai jenis serangan *(multi attack)* untuk mengidentifikasi pola serangan yang memengaruhi *Network Attached Storage (NAS Server)*. Serangan *Brute Force* dan *Distributed Denial of Service (DDoS)* digunakan untuk mempengaruhi sistem keamanan *server NAS.* Pada penelitian yang lain [6], Membahas tentang Mengoptimalkan Pencegahan Serangan *Brute Force* pada *Linux* melalui Penerapan Metode *Aplikasi IDS Snort*. Dalam penelitian ini akan mengidentifikasi aktivitas ilegal jika ada akses tak terduga ke *server*. *Snort* kemudian akan mendeteksi dan memberi tahu *administrator jaringan* tentang adanya aktivitas tidak wajar. Hasilnya menunjukkan bahwa penggunaan *Snort* dan *IPTables* sebagai sistem keamanan *server* pada *jaringan nirkabel* secara efektif mengalahkan serangan pada *port ICMP, FTP, SSH, TELNET, dan HTTP* oleh berbagai penyerang, termasuk *DDoS, Brute Force, CMS/ Framework, Inject Malware, Email Fraud, dan Spam.*

Pada penelitian sebelumnya [7], Membahas tentang Implementasi *Honeypot Cowrie* dan *Snort* sebagai Alat Deteksi Serangan pada *Server*. Pendekatan *Network Development Life Cycle* (NDLC) digunakan dalam penelitian ini, dan penulis menggunakan *Honeypot Cowrie* dan *Snort* sebagai alat deteksi serangan *server*. Karena *Honeypot Cowrie* dirancang untuk mengidentifikasi dan menyelidiki serangan pada protokol *SSH* diantaranya adalah Serangan *Bruteforce, Credential Stuffing, Command Injection, Shellcode Execution, dan Remote Code Execution,* hasil pengujian tidak dapat mengidentifikasi Serangan *DDOS* yang menyusup ke *server*. Meskipun *Snort* adalah sistem deteksi intrusi (IDS) yang dirancang untuk mengidentifikasi serangan yang masuk, tetapi *Snort IDS* hanya dapat mengidentifikasi serangan yang masuk dan tidak dapat menyelidikinya. Hasilnya, *Snort* mampu mendeteksi serangan *Bruteforce dan DDOS*. Juga pada penelitian sebelumnya [8], Membahas tentang Analisis Penyerangan *Bruteforce* terhadap *Secure Shell* (SSH) Menggunakan Metode *Penetration Testing*. Tujuan dari penelitian ini adalah untuk menggunakan pendekatan pengujian *Penetration Testing* guna memeriksa serangan *bruteforce* terhadap *SSH*. Diharapkan pemahaman yang lebih mendalam tentang metode serangan *bruteforce*, dampaknya terhadap keamanan sistem, dan tindakan pencegahan akan diperoleh sebagai hasil dari penelitian ini.

Pada penelitian ini penulis akan melakukan analisis forensik jaringan *(Network Forensic)* menggunakam tools seperti *Wireshark* dan *Snort* pada *server owncloud* yang diserang menggunakan serangan *brute force attack* terhadap sistem keamanan *server owncloud*, dengan aturan-aturan *(rule-based)* pada sistem *server owncloud.* Penerapan metode *rule based* pada penelitian yang dilakukan melibatkan aturan penggunaan dan skenario yang sudah ditentukan sebelumnya untuk mengidentifikasi pola serangan yang mencurigakan. Hal ini termasuk memeriksa pola aktivitas *login*, seperti mengetahui jumlah upaya *login* yang gagal selama periode waktu tertentu dari IP yang sama. Metode *rule based* menggunakan aturan *(rule)* sebagai representasi pengetahuan yang diterapkan dalam sistem.

2. BAHAN DAN METODOLOGI PENELITIAN

Penelitian ini melibatkan beberapa tahapan dan langkah-langkah, sebelum nantinya mendapatkan hasil akhir penerapan metode dan serangan *brute force* yang terjadi pada *Owncloud* diantaranya meliputi:

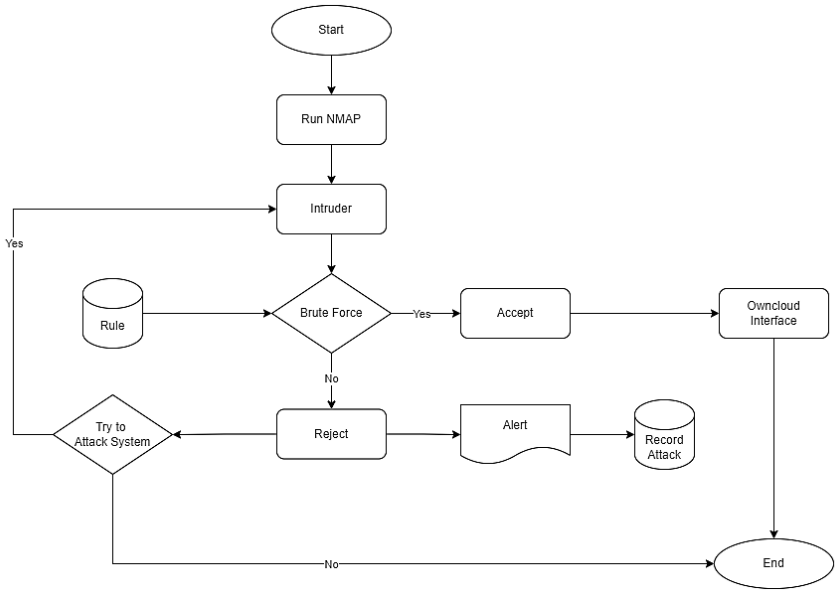
1. Analisa Kebutuhan

   Analisa kebutuhan perangkat diperlukan. Kebutuhan perangkat dibagi menjadi dua kategori yaitu kebutuhan perangkat keras *(hardware)* dan kebutuhan perangkat lunak *(software).* penulis akan melakukan analisis terhadap kebutuhan perangkat keras dan perangkat lunak yang diperlukan.

2. Perancangan Konfigurasi *Owncloud dan VirtualBox*
   Langkah-langkah yang terlibat dalam perancangan komputer fisik yang menggunakan *Owncloud* sebagai solusi penyimpanan *cloud* dan virtualisasi sistem server dengan *VirtualBox* yang menjalankan *Linux Ubuntu* pada *PC* yang terhubung ke jaringan lokal pribadi.

3. Perancangan dan Konfigurasi *Intruder*
   Tahap ini melibatkan proses *instalasi* dan konfigurasi intruder *(penyerang).* Konfigurasi *intruder* terdiri dari pemasangan alat serangan pada *kali linux.*

4. Implementasi Serangan terhadap *Owncloud*
   Implementasi adalah pelaksanaan yang dilakukan dengan aksi atau tindakan pada dunia nyata [9]. Salah satu fase paling penting dalam menentukan apakah suatu proyek yang direncanakan dan dirancang sebelumnya akan berhasil atau gagal adalah implementasi. Pada tahap ini dilakukan pengujian serangan *intruder* terhadap *server Owncloud* menggunakan *tools nmap*.



Gambar 1. *Flowchart* Pengujian Serangan

2.1. Metode Rule Based

Sistem pakar adalah perangkat lunak komputer yang dirancang untuk menangani masalah di bidang tertentu dengan menggunakan pengetahuan atau keterampilan manusia. Para pakar di bidang tersebut biasanya merupakan sumber pengetahuan yang terintegrasi ke dalam sistem. Sistem pakar mampu meniru metode berpikir yang digunakan oleh para pakar dan menarik kesimpulan logis dari pengetahuan mereka. Sistem berbasis aturan *(rule based system)* adalah jenis sistem pakar yang umum. Karena banyaknya manfaatnya, metode sistem berbasis aturan umumnya digunakan dalam pengembangan sistem pakar [10].

2.2. Brute Force

Brute force adalah metode penyelesaian masalah yang sering kali dimulai dengan mendefinisikan ide-ide relevan dan pernyataan masalah. *Brute force* dapat menyelesaikan masalah dengan cepat dan dengan mentalitas yang lugas [11]. *Brute Force* merupakan teknik dalam mencocokkan kata atau *string* pada teks dengan *pattern* di tiap karakter dimulai dari kiri ke kanan. Teknik ini digunakan untuk mencari solusi dengan cara menguji semua kemungkinan secara sistematis. bekerja dengan membandingkan karakter satu per satu antara teks dan pola dari kiri ke kanan. Langkah-langkah pencocokkan dilakukan secara terstruktur dengan mengurutkan tiap solusi satu per satu untuk menemukan solusi terbaik. Contoh penerapan *algoritma brute force* adalah dalam mencocokkan *string* dengan *pattern*, dimana langkah pertama adalah melakukan pencocokkan karakter satu per satu antara *string* dan *pattern* [12]. Berikut ini adalah langkah-langkah khusus yang dilakukan metode *brute force* untuk mencocokkan *string* [13]:

1. Dari awal teks, *algoritma brute force* mulai mencocokkan pola.
2. Hingga salah satu keadaan berikut terpenuhi, metode *brute force* akan mencocokkan setiap pola karakter dari kiri ke kanan dengan karakter dalam teks yang relevan.
3. Pencarian selesai jika karakter dalam pola yang dibandingkan cocok.
4. Pencarian tidak berhasil jika ada perbedaan antara pola dan konten.

5. Hingga pola mencapai akhir teks, metode *brute force* terus memindahkannya satu langkah ke kanan dan kemudian kembali ke langkah 2.

Algoritma brute force memiliki kelebihan dan kekurangan, sama seperti algoritma lainnya. *Brute force* memiliki kelebihan sebagai berikut:

1. Hampir semua masalah dapat dipecahkan menggunakan pendekatan *brute force*.
2. *Algoritma brute force* memanggil kata-kata dengan lebih cepat.
3. Untuk sejumlah tugas penting, termasuk penyortiran, pencarian, pencocokan *string*, dan perkalian *matriks*, teknik *brute force* menghasilkan algoritma yang dapat digunakan.
4. Untuk tugas komputasi seperti menambahkan atau mengalikan n bilangan bulat atau mencari tahu elemen minimum atau maksimum dalam tabel (daftar), pendekatan *brute force* menghasilkan prosedur standar.

Kelemahan *algoritma brute force* adalah kurang efektif dibandingkan pendekatan alternatif untuk pemecahan masalah.

2.3. Owncloud

Owncloud adalah sekelompok profesional di industri masing-masing yang dipimpin oleh Frank Karlitschenck, seorang spesialis sumber terbuka *(open source)* yang berpengalaman dan dapat diandalkan, dan Markus Rex mendirikan perusahaan tersebut pada tahun 2011. *Owncloud* adalah *program* untuk membuat server penyimpanan awan *(cloud storage)* yang memungkinkan pengguna mengakses fitur-fitur seperti berbagi data dan penyimpanan pribadi dan publik [14]. Dalam kategori *Infrastructure as a Service (IaaS), Owncloud* adalah layanan *web* yang memfasilitasi berbagi berbagai jenis *data*, termasuk dokumen, audio, video, foto, dan banyak lagi. Pengguna dapat mengakses dan menyinkronkan berkas di *server Owncloud* menggunakan komputer desktop, perangkat seluler, atau peramban web [15].

2.4. NMAP

Aplikasi open source yang disebut *Nmap* digunakan untuk memeriksa dan meneliti keamanan jaringan. *Nmap* terkenal sebagai salah satu alat yang dapat menjelajahi jaringan dengan cepat, bahkan dalam ekosistem jaringan yang besar. *Administrator* jaringan dapat memperluas fungsinya dengan menambahkan kemampuan untuk menginventarisasi jaringan, mengelola jadwal pembaruan layanan, dan terus memantau ketersediaan host dan layanannya untuk memastikannya tetap aktif, selain menggunakan teknik pemindaian *port*, identifikasi *host*, dan *Nmap Scripting Engine (NSE)* untuk menemukan celah keamanan. *Nmap* dapat digunakan untuk mencari port yang terbuka atau tertutup untuk diselidiki menggunakan teknik pemindaian *port* atau dengan memasukkan lalu lintas jaringan [16].

Nmap menghasilkan daftar *host* target yang telah dipindainya beserta informasi lebih lanjut. "Tabel *port* yang menarik" merupakan komponen penting dari tabel ini. Protokol, nomor *port*, nama layanan, dan status tercantum dalam tabel ini. Ada empat status berbeda: terbuka, tersaring, tertutup, dan tidak tersaring. Aplikasi pada komputer target dikatakan terbuka jika mendengarkan koneksi atau paket pada *port* tersebut. Tersaring menunjukkan bahwa *port* diblokir oleh *firewall, filter,* atau pemblokiran jaringan lainnya, sehingga *Nmap* tidak dapat menentukan apakah *port* tersebut terbuka atau tertutup. *Aplikasi* tidak mendengarkan pada *port* yang tertutup, tetapi dapat terbuka kapan saja. Ketika *port* bereaksi terhadap pemeriksaan *Nmap* tetapi *Nmap* tidak dapat mengidentifikasi apakah *port* tersebut terbuka atau tertutup, *port* tersebut dikategorikan sebagai tidak tersaring. Ketika *Nmap* tidak dapat mengidentifikasi status yang mendefinisikan *port*, *Nmap* melaporkan kombinasi status terbuka|tersaring dan tertutup tersaring. Rincian tentang versi perangkat lunak yang diminta untuk melakukan pemeriksaan versi juga dapat disertakan dalam tabel *port. Nmap* memberikan rincian tentang *protokol IP* yang didukung alih-alih *port* pendengaran saat pemindaian *protokol IP* diminta [17].

2.5. Port Scanning

Salah satu metode untuk menemukan *port* dan layanan yang terbuka di sisi target adalah pemindaian *port*. Ada beberapa variasi metode pemindaian ini. Meskipun beberapa metode ini dimaksudkan untuk kejahatan, sebagian besarnya sebenarnya aman. Ada 65.535 port *TCP* dan 65.535 port *UDP*, yang merupakan jumlah *port* yang sangat besar. Nomor *port* yang paling dikenal luas adalah antara nol dan 1024. *Port* 80 ditautkan ke *HTTP*, misalnya, sementara *port* 21 sesuai dengan *FTP*, *port* 25 ke *SMTP*, dan seterusnya. Sebagai teknik pengintaian, pemindaian *port* mencari informasi tentang *port* dan layanan yang terbuka pada *host.* Selain itu, metode ini sering kali menentukan *port* mana yang dapat dibuka dan mengirimkan pesan ke semua *port* yang terbuka [18].

2.6 Virtual Box

Program virtualisasi yang disebut *Oracle VM VirtualBox* digunakan untuk menjalankan tugas sistem "ekstra" pada sistem "utama". Misalnya, ketika seseorang menginstal *Microsoft Windows* di komputernya,

mereka juga dapat menggunakan *sistem operasi* yang disukai. Program *virtualisasi* lintas platform yang disebut *Oracle VM VirtualBox* diinstal pada komputer dengan prosesor *AMD* atau *Intel*. Ini meningkatkan kapasitas komputer untuk menjalankan beberapa sistem operasi pada banyak mesin *virtual* sekaligus [19].

2.7 Wireshark

Wireshark adalah salah satu analisis paket bebas serta sumber terbuka [20]. *Aplikasi* bernama *Wireshark* dirancang untuk menyelidiki paket data jaringan. Nama lain untuk *Wireshark* adalah penganalisis paket jaringan, yaitu *program* yang merekam paket jaringan dan berupaya memberikan setiap *bit* informasi setepat mungkin. Penganalisis paket jaringan pada dasarnya adalah alat untuk memeriksa status jaringan yang sebenarnya, baik nirkabel maupun kabel. Memantau dan menganalisis paket saat paket tersebut bergerak melalui jaringan menjadi sangat mudah dengan *Wireshark* [21].

2.8 Snort

Snort adalah alat keamanan yang dapat digunakan sebagai perangkat lunak atau *aplikasi* untuk mengidentifikasi perintah jaringan dan mengidentifikasi intrusi jaringan. Lalu lintas jaringan dipantau secara pasif oleh *Snort*, sistem pencegahan dan deteksi intrusi berbasis aturan sumber terbuka yang memberi tahu pengguna saat bahaya teridentifikasi. Dengan kemampuannya yang sangat dapat dikonfigurasi, *Snort* dapat diatur sesuai kebutuhan. *Snort* dapat bekerja dengan sistem manajemen keamanan yang lebih komprehensif dan memfasilitasi integrasi dengan teknologi keamanan lainnya [22]. Salah satu dari tiga mode operasi *Snort* adalah *Packet Sniffer*, yang memungkinkan Anda melihat paket-paket jaringan. *Packet logger*, yang merekam setiap paket yang bergerak melintasi jaringan untuk penjelasan selanjutnya. Saat *Snort* berada dalam mode *NIDS (Network Intrusion Detection System)*, *Snort* dapat mengidentifikasi serangan yang dilakukan melintasi jaringan komputer [23].

3. HASIL DAN PEMBAHASAN

Pada titik ini, simulasi serangan dapat dilakukan dan semua kebutuhan uji serangan *brute force* telah terinstal. Setelah itu, sejumlah upaya serangan akan digunakan untuk memeriksa skenario serangan *brute force SSH*. Hasil ini menunjukkan bahwa *server Owncloud* target serangan *brute force* berhasil diserang. *Snort* digunakan sebagai detektor insiden *brute force*. Serangan *brute force SSH* pada sisi *server Owncloud* berhasil diidentifikasi oleh aturan *Snort* yang telah dikonfigurasi.

Gambar 2. Tampilan *Login Owncloud*

3.1. Kebutuhan Fungsional

Beberapa kebutuhan fungsional yang diperlukan dalam penelitian ini.

1. Mekanisme *Brute Force*: Diperlukan mekanisme untuk melakukan serangan *brute force* pada proses *autentikasi OwnCloud*. Mekanisme ini dapat berupa script atau *program* yang mampu mengirimkan kombinasi *username* dan *password* secara otomatis dan sistematis.
2. Daftar Kata Sandi: Untuk melakukan serangan *brute force*, dibutuhkan daftar kata sandi (*wordlist*) yang berisi kumpulan kata sandi yang mungkin digunakan oleh pengguna. Daftar ini dapat dibuat sendiri atau menggunakan daftar kata sandi yang sudah tersedia secara publik.
3. Manajemen Percobaan: Diperlukan mekanisme untuk mengelola jumlah percobaan login yang dilakukan oleh sistem *brute force*. Hal ini diperlukan untuk mencegah terkuncinya akun pengguna setelah mencapai batas maksimum percobaan *login* yang gagal.

4. Pengumpulan Data: Sistem harus dapat mengumpulkan data seperti waktu yang dibutuhkan untuk melakukan serangan *brute force*, jumlah percobaan yang dilakukan, dan informasi lain yang relevan untuk keperluan analisis dan evaluasi.
5. Pelaporan dan Visualisasi: Diperlukan mekanisme untuk membuat laporan dan visualisasi data yang telah dikumpulkan selama proses serangan *brute force*. Laporan dan visualisasi ini dapat membantu dalam menganalisis kerentanan sistem dan efektivitas serangan.
6. Integrasi dengan *OwnCloud*: Sistem penelitian harus dapat terintegrasi dengan lingkungan *OwnCloud* yang akan diserang. Hal ini dapat dilakukan dengan menginstal *OwnCloud* pada lingkungan pengujian atau dengan mengakses instance *OwnCloud* yang sudah ada.

3.2. Alat dan Bahan Penelitian

1. Perangkat Keras *(Hardware)*
   Spesifikasi perangkat keras yang dibutuhkan untuk mengembangkan dan menguji sistem pada penelitian tercantum pada tabel 1.

Tabel 1. Kebutuhan Perangkat Keras

Perangkat	Spesifikasi	Detail
Laptop Asus TUF FX505DT-R565B6T	*Computer Name*	*LAPTOP-TUF*
	Manuvacture	*ASUSTeK Computer Inc.*
	Model	*FX505DT*
	BIOS	*American Megatrends Inc.*
	Processor	*AMD Ryzen 5-3550H*
	Memory	*16 GB*
	Card Name	*NVIDIA GeForce GTX 1650*
	Manuvacture	*ASUSTeK Computer Inc.*
	Display Memory	*4 GB*
	Current Display Memory	*1920 x 1080 (64bit) (120hz)*
	Storage	*1256*

2. Perangkat Lunak (*Software*)
   Spesifikasi perangkat lunak yang dibutuhkan untuk mengembangkan dan menguji sistem pada penelitian tercantum pada tabel 2.

Tabel 2. Kebutuhan Perangkat Lunak

Sistem	Tools	Keterangan
Virtual Machine	Oracle VM VirtualBox	VirtualBox 7.0
Cloud	Owncloud	Owncloud X
Capture Traffic	Wireshark	Wireshark 4.2.4
NIDS	Snort	Snort 2.9.20
Attack Traffic	Kali Linux	Kali 2024.2

3.3. Analisis Masalah

Penggunaan penyimpanan awan *(cloud storage)* dengan kapasitas besar sangat menarik minat berbagai kalangan. Banyak dari kalangan yang menggunakan *Owncloud* sebagai penyimpanan awan *(cloud storage). Owncloud* memungkinkan pengguna dapat menyimpan, mengelola, dan berbagi *file* antar pengguna dengan memberikan kendali penuh kepada penggunanya. Pengguna tidak bisa menjamin kemanan data pada *server Owncloud* yang dikelola.

Banyak terjadi upaya serangan dengan melakukan *brute force attack* secara terus menerus yang dapat mengancam keamanan pengguna dengan mencoba menebak kata sandi pengguna atau *administrator* yang dapat mengakibatkan pelaku dapat mengakses *file sensitif*. Sehingga diperlukan analisa bagaimana pola serangan pada *Owncloud* tersebut dapat terdeteksi.

3.4. Tahapan Serangan *Intruder* terhadap *Owncloud*

Pada tahap ini dilakukan proses serangan *brute force* terhadap *server Owncloud*. Gambar 3 merupakan hasil dokumentasi serangan *brute force* menggunakan *tools nmap* pada *terminal kali linux*. Melakukan serangan terhadap *open port SSH* pada *Owncloud server*. *Open port SSH* pada *Owncloud* terdeteksi dari hasil *nmap* dengan nilai *port* 22.

```
┌──(khazin㉿khazin)-[~]
└─$ nmap 192.168.1.11 -p 22 --script ssh-brute --script-args userdb=/home/khazin/wordlist/userlist.txt
,passdb=/home/khazin/wordlist/passlist.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-02 11:59 EDT
NSE: [ssh-brute] Trying username/password pair: root:root
NSE: [ssh-brute] Trying username/password pair: root:khazin
NSE: [ssh-brute] Trying username/password pair: root:khazin14
NSE: [ssh-brute] Trying username/password pair: root:Khazin14
NSE: [ssh-brute] Trying username/password pair: root:123456
Nmap scan report for 192.168.1.11 (192.168.1.11)
Host is up (0.0011s latency).

PORT   STATE SERVICE
22/tcp open  ssh
| ssh-brute:
|   Accounts:
|     root:Khazin14 - Valid credentials
|_  Statistics: Performed 5 guesses in 4 seconds, average tps: 1.2

Nmap done: 1 IP address (1 host up) scanned in 4.12 seconds
```

Gambar 3. Serangan *brute force ssh* dengan *tools nmap*

Gambar 3 menampilkan serangan yang dilakukan kepada *Owncloud server*. Dimana *tools nmap* berhasil terdeteksi proses *brute force* yang menjelaskan bahwa "*Account: root*: Khazin14" berhasil terhubung dengan mencoba melakukan 10 percobaan *password list* yang telah dibuat dapat dilihat pada Gambar 4.

```
┌──(khazin㉿khazin)-[~/wordlist]
└─$ cat passlist.txt
khazin
khazin14
Khazin14
123456
admin1
password
lumpia
aku123
123masuk
10101010
```

Gambar 4. *Password list*

Pemeriksaan isi paket log *Wireshark* memungkinkan pemeriksaan hasil insiden *brute force*. Gambar 5 menunjukkan contoh lalu lintas selama serangan *brute force SSH*. Informasi tentang upaya penyerangan menggunakan *port* tujuan 22, yang merupakan *port* untuk *SSH* yang ditemukan selama penyelidikan ini.

No.	Time	Source	Destination	Protocol	Length	Info
1308	63.821710817	192.168.1.11	10.0.2.15	TCP	60	22 → 37982 [ACK] Seq=1122 Ack=1485 Win=65535 Len=0
1309	63.821821400	192.168.1.11	10.0.2.15	TCP	60	22 → 37982 [ACK] Seq=1122 Ack=1593 Win=65535 Len=0
1310	63.821821590	192.168.1.11	10.0.2.15	TCP	60	22 → 37996 [ACK] Seq=1122 Ack=1485 Win=65535 Len=0
1311	63.821821680	192.168.1.11	10.0.2.15	TCP	60	22 → 37996 [ACK] Seq=1122 Ack=1593 Win=65535 Len=0
1312	63.821893431	192.168.1.11	10.0.2.15	TCP	60	22 → 38012 [ACK] Seq=1122 Ack=1485 Win=65535 Len=0
1313	63.821893551	192.168.1.11	10.0.2.15	TCP	60	22 → 38012 [ACK] Seq=1122 Ack=1593 Win=65535 Len=0
1314	63.821960580	192.168.1.11	10.0.2.15	TCP	60	22 → 38022 [ACK] Seq=1122 Ack=1485 Win=65535 Len=0
1315	63.821960720	192.168.1.11	10.0.2.15	TCP	60	22 → 38022 [ACK] Seq=1122 Ack=1593 Win=65535 Len=0
1316	63.822020723	192.168.1.11	10.0.2.15	TCP	60	22 → 38024 [ACK] Seq=1122 Ack=1485 Win=65535 Len=0
1317	63.822020863	192.168.1.11	10.0.2.15	TCP	60	22 → 38024 [ACK] Seq=1122 Ack=1593 Win=65535 Len=0

Gambar 5. Hasil rekaman *snort* pada percobaan *SSH brute force*

Hasil *log Snort* diperiksa untuk melakukan analisis. Rincian penting yang dapat digunakan untuk menunjukkan bahwa telah terjadi serangan pada *server Owncloud* ditampilkan pada Gambar 6. Tanggal 3 Juni 2024, pukul 22:59, penyerang *(intruder)* meluncurkan serangan menggunakan *IP 10.0.2.15*. Halaman *login Owncloud* mungkin menjadi sasaran langsung serangan *brute force*, seperti yang terdeteksi oleh aturan *Snort* dalam penyelidikan ini.

```
06/02-15:59:05.806182  [**] [1:10000002:3] HTTP test [**] [Priority: 0] {TCP} 192.168.1.10:51209 ->
192.168.1.11:80
06/02-15:59:05.806667  [**] [1:10000002:3] HTTP test [**] [Priority: 0] {TCP} 192.168.1.10:51209 ->
192.168.1.11:80
06/02-15:59:05.869007  [**] [1:10000004:1] Possible SSH brute forcing! [**] [Priority: 0] {TCP} 192.
168.1.10:51215 -> 192.168.1.11:22
06/02-15:59:09.795999  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP} 192.168.1.11 -> 192.1
68.1.10
06/02-15:59:09.796030  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP} 192.168.1.11 -> 192.1
68.1.10
06/02-15:59:09.796045  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP} 192.168.1.11 -> 192.1
68.1.10
```

Gambar 6. Pesan peringatan serangan SSH *brute force*

Tabel 3. Hasil Investigasi

No.	Jenis Informasi Temuan	Keterangan
1.	Jenis serangan yang terjadi pada server Owncloud	SSH brute force, port scanning, dan brute force login Owncloud.
2.	Capture jaringan dapat dibaca menggunakan utilitas Wireshark	Ditemukan bahwa server Owncloud dipenuhi paket TCP yang pada gilirannya memenuhi disk atau penyimpanan.
3.	IP address intruder	10.0.2.15
4.	IP address server Owncloud	192.168.1.11
5.	Rekrontruksi terjadi serangan	Scanning SSH -> port open -> serangan brute force -> percobaan login tidak ada batasan -> berhasil login brute force Owncloud

4. KESIMPULAN

Bukti digital serangan brute force Owncloud yang berhasil ditemukan selama penelitian ini mencakup informasi tentang waktu serangan, alamat IP penyerang, port target serangan, protokol, dan alat yang digunakan. Dalam salah satu tahap forensik jaringan, proses serangan dapat difasilitasi oleh visualisasi bukti yang diperoleh melalui pelaksanaan alat nmap yang dimaksud.

Application Programming Interface (API) Owncloud memiliki kelemahan yang memungkinkan serangan brute force dilakukan pada API tersebut. Jumlah maksimum percobaan login yang gagal bagi pengguna tidak dibatasi oleh Owncloud pada halaman login. Selain mengharuskan pengguna untuk mengirimkan nama pengguna dan kata sandi mereka pada halaman login owncloud, sistem pengguna akan secara otomatis mengirimkan informasi tentang timezone, timezone-offset ,dan permintaan token. Untuk mencegah serangan brute force yang terjadi pada halaman login owncloud, nilai permintaan token dipilih secara random. Saat memproses raw data dengan banyak rules, diperlukan sebuah metode yang lebih efektif untuk mendeteksi serangan brute force pada owncloud..

REFERENSI

[1] Suryati, S., Disurya, R., Ermini, E., Sardana, L., Husnulwati, S., Wahyuningsih, S., & Jumroh, J. (2019). Sosialisasi Praktik dan Perlindungan Pengguna Internet di SMA Negeri 1 Sungai Liat. *Jurnal PkM Pengabdian Kepada Masyarakat*, *2*(02), 167. https://doi.org/10.30998/jurnalpkm.v2i02.3466

[2] Zahara, S., Sugianto, & M. Bahril Ilmiddafiq. (2019). Prediksi Indeks Harga Konsumen Menggunakan Metode Long Short Term Memory (LSTM) Berbasis Cloud Computing. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *3*(3), 357–363. https://doi.org/10.29207/resti.v3i3.1086

[3] Kadapi, M. (2021). Forensic Serangan Brute Force pada Public Cloud dengan Metode Rule Base. Retrieved from http://repository.unsri.ac.id/id/eprint/39519

[4] Hasbi, M., & Saputra, N. R. (2021). Analisis Quality of Service (Qos) Jaringan Internet Kantor Pusat King Bukopin Dengan Menggunakan Wireshark. *Universitas Muhammadiyah Jakarta*, *12*(1), 1–7. Retrieved from https://jurnal.umj.ac.id/index.php/just-it/article/view/13596

[5] Faisal, I., Handoko, D., & Putra, H. (2024). *Penerapan Metode Rule Based Dalam Mendeteksi Serangan Multi Attack Pada Network Attached Storage*. Retrieved from https://doi.org/10.62712/juktisi.v2i3.138

[6] Hidayat, D., & Ramli, R. (2023). Mengoptimalkan Pencegahan Serangan Brute Force pada Linux melalui Penerapan Metode Aplikasi IDS Snort. *JiTEKH*, *11*(2), 57–61. https://doi.org/10.35447/jitekh.v11i2.764

[7] Natanegara, T., Muhyidin, Y., & Singasatia, D. (2023). Implementasi Honeypot Cowrie Dan Snort Sebagai Alat Deteksi Serangan Pada Server. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(3), 1871–1877. https://doi.org/10.36040/jati.v7i3.6989

[8] Febrian, R. A., Muhyidin, Y., & Singasatia, D. (2024). Analisis Penyerangan Bruteforce Terhadap Secure Shell (Ssh) Menggunakan Metode Penetration Testing. *Scientica: Jurnal Ilmiah Sains Dan Teknologi*, *2*(11), 151–162. https://jurnal.kolibi.org/index.php/scientica/article/view/2762

[9] Nirsal, Rusmala, & Syafriadi. (2020). Desain dan Implementasi Sistem Pembelajaran Berbasis E-Learning pada Sekolah Menengah Pertama Negeri 1 Pakue. D'computare: Jurnal Ilmiah Teknologi Informasi Dan Ilmu Komputer, 10(1). https://doi.org/10.30605/dcomputare.v10i1.24

[10] Shuda Syaifah. (2023). Penerapan Bukti Lulus Uji Elektronik Dalam Pengujian Kendaraan Bermotor Berdasarkan Permenhub Nomor Pm 19 Tahun 2021 Pasal 64 Ayat 1 Menurut Perspektif Siyasah Idariyyah (Studi Kasus Pada Upt Pengujian Kendaraan Bermotor Kota Dumai). Retrieved from http://repository.uin-suska.ac.id/id/eprint/73040

[11] Gede, I., Putra, K. S., Aranta, A., Suta Wijaya, P., & Gede Andika, I. (2023). Rancang Bangun Aplikasi Transliterasi Aksara Bali Menjadi Huruf Latin Menggunakan Metode Rule Based Pada UTF-16 Berbasis Android. *RESISTOR, Vol. 6 No*. https://doi.org/10.31598

[12] Tazkiya Ramadhoni, A., Hartami Santi, I., & Kirom, S. (2022). Penerapan Algoritma Brute Force Pada Aplikasi Sidayko Berbasis Android. *Jurnal Mnemonic*, *5*(1), 1–8. https://doi.org/10.36040/mnemonic.v5i1.4233
[13] H. Hidayat. (2019). Implementasi Algoritma Brute Force dalam Glosarium Sosiologi Berbasis Web. STMIK AKAKOM YOGYAKARTA. http://eprints.akakom.ac.id/id/eprint/8347
[14] Rahmawati, Y., Adi Pribadi, I. ., & Heningtyas, Y. (2021). Penerapan Algoritma Brute Force pada Menu Search Website "Calonku" dalam Rangka Pemilu Berbasis Web. *Jurnal Pepadun*, 2(1), 60–70. https://doi.org/10.23960/pepadun.v2i1.36
[15] Manalu, A. S., & Sitanggang, S. S. (2019). Perancangan Dan Implementasi Private Cloud Storage Dengan Owncloud Pada Jaringan Lokal Menggunakan Virtualbox. *Journal of Computer Networks, Architecture, and High-Performance Computing*, *1*(2), 60–71. https://doi.org/10.47709/cnahpc.v1i2.244
[16] Wiyono, P., & Maslan, A. (2021). Perancangan Private Cloud Computing Menggunakan Owncloud. *Computer and Science Industrial Engineering (COMASIE)*, *05*(02). https://ejournal.upbatam.ac.id/index.php/comasiejournal/article/view/3904
[17] Dwiyatno, S. (2020). Analisis Monitoring Sistem Jaringan Komputer Menggunakan Software Nmap. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, *7*(2), 108–115. https://doi.org/10.30656/prosisko.v7i2.2522
[18] Sudirman, D., & Akma Nurul Yaqin. (2021). Network Penetration dan Security Audit Menggunakan Nmap. *SATIN - Sains Dan Teknologi Informasi*, *7*(1), 32–44. https://doi.org/10.33372/stn.v7i1.702
[19] Oracle VM VirtualBox. (n.d.). Available at: https://www.oracle.com/. [Accessed 1 Juni 2024]
[20] Novita, R. T., Gunawan, I., Marleni, I., Grasia, O. G., & Valentika, M. N. (2021). Analisis Keamanan Jaringan Wifi Menggunakan Wireshark. *JES (Jurnal Elektro Smart)*, *1*(1), 10–12. Retrieved from https://www.sttrcepu.ac.id/jurnal/index.php/jes/article/view/159
[21] Hasbi, M., & Saputra, N. R. (2021). Analisis Quality of Service (Qos) Jaringan Internet Kantor Pusat King Bukopin Dengan Menggunakan Wireshark. *Universitas Muhammadiyah Jakarta*, *12*(1), 1–7. https://doi.org/10.24853/justit.12.1.%25p
[22] Fuada, Z. (2023). Penerapan Keamanan Jaringan Menggunakan Sistem Snort dan Honeypot sebagai Pendeteksi dan Pencegah Malware. *UIN Ar-Raniry Fakultas Tarbiyah Dan Keguruan*. Retrieved from https://repository.ar-raniry.ac.id/id/eprint/36133
[23] Purnama, T., Muhyidin, Y., & Singasatia, D. (2023). Implementasi Intrusion Detection System (Ids) Snort Sebagai Sistem Keamanan Menggunakan Whatsapp Dan Telegram Sebagai Media Notifikasi. *JURNAL ILMIAH TEKNOLOGI INFORMASI DAN KOMUNIKASI (JTIK) VOL*, *14*(2), 358–369. https://doi.org/10.51903/jtikp.v14i2.726