OPTIMASI KEAMANAN *WEBSERVER* RUMAH SAKIT UMUM DAERAH PALEMBANG BARI (rsudbari.palembang.go.id)

Ahmad Redho Rivai[1]. Fatoni[2]. Taqrim Ibadi[3]
Jalan Jenderal Ahmad Yani No.12 Palembang
Pos-el : redho.rivai@yahoo.com[1],
Fatoni@binadarma.ac.id[2], Taqrimibadi@binadarma.ac.id[3]

Abstrack Perkembangan teknologi telah berkembang dengan pesat, perbedaan teknologi yang dulu dan sekarang sudah jauh berbeda. Dengan adanya perkembangan teknologi ini akan sangat membantu memberikan kemudahan salah satunya informasi, dengan menggunakan akses internet semua informasi dapat ditemukan. Perkembangan teknologi *websever* ini sudah banyak dipakai hampir seluruh instansi-instansi penting, salah satunya Rumah Sakit Umum Daerah Palembang Bari. Dengan menggunakan teknologi informasi berbasis *webserver* ini semua informasi yang ada di Rumah Sakit Umum Daerah Palembang Bari dapat diketahui tanpa perlu datang langsung ke lokasi. Sistem *webserver* biasanya tidak luput dari kerentanan atau celah, oleh karena itu dibutuhkan pengamanan pada sistem tersebut. dengan menggunakan metode *action research* serta dibantu oleh teknik *Confidentiality, Integrity, Availability (CIA),* Optimasi dan *Penetration Testing (pentest),* diharapkan dapat membantu dalam mengoptimalkan *webserver.* Dengan menggunakan ketiga teknik tersebut, serta dibantu oleh beberapa *script firewall,* telah berhasil melakukan optimasi yang membuat *webserver* tersebut dapat bekerja secara optimal serta tidak memiliki celah seperti sebelumnya.
Kata Kunci : *Webserver, Internet, Confidentiality, Integrity, Availability (CIA), Optimasi, Penetration Testing (pentest), Script Firewall*

1. PENDAHULUAN

Perkembangan teknologi telah berkembang pesat, perbedaan antara teknologi dulu dan sekarang telah jauh berbeda. Dengan adanya perkembangan teknologi ini, orang mampu mencari informasi dengan sangat mudah dan cepat. Untuk melakukan pencarian suatu informasi cukup dengan menggunakan internet, dengan cara memasukan alamat suatu situs di aplikasi *browser* dengan benar, maka informasi yang dicari dapat ditemukan.

Perkembangan teknologi situs *website* ini, nampaknya sudah menjadi kebiasaan masyarakat umum untuk mempermudah semua aktifitas mereka, sehingga hampir seluruh instansi penting seperti pemerintah menggunakan teknologi situs *website,* salah satunya yaitu Rumah Sakit Umum Daerah Palembang Bari dengan *website* http://rsudpbari.palembang.go.id. *Website* ini untuk mempermudah masyarakat umum mendapatkan informasi tentang Rumah Sakit Umum Daerah Palembang Bari tanpa perlu datang langsung ke rumah sakit tersebut. Namun perlu kita ketahui bahwa situs *website* yang digunakan tersebut di latar belakangi oleh *server* sebagai perangkat kerasnya. *Webserver* yang kita gunakan tersebut tidak menutup kemungkinan terdapat celah-celah yang menjadi ancaman bagi *administrator webserver* tersebut.

Berdasarkan informasi yang didapat dari Rumah Sakit Umum Daerah Palembang Bari, bahwa *website* yang mereka gunakan sering terjadi manipulasi tampilan *(Web Deface).* Dan dari data *scanning* tersebut dapat diketahui bahwa manipulasi tampilan *(Web Deface)* terjadi dikarenakan terdapat serangan *SQL* (*Structured Query Language Injection),* hal ini menandakan

bahwa *website* yang ada di rumah sakit tersebut lemah.

Keretanan yang ditimbulkan oleh *website* tersebut melalui kesalahan pada *database*, jenis serangannya yaitu *SQL (Strutured Query Language Injection),* Tidak hanya *SQL injection* yang dapat terjadi pada setiap *website* akan tetapi salah satunya yaitu serangan menggunakan teknik *arbitrary upload.* Dengan semakin banyaknya jenis-jenis serangan pada *website*, tentu ini menjadi pekerjaan tambahan bagi seorang *administrator* suatu *website*, dikarenakan mereka harus menjaga agar sistem tetap aman dan sistem tersebut dapat berjalan sesuai fungsinya, serangan yang terjadi pada *website* memang harus diperhatikan, akan tetapi perlu diketahui *server* pun harus perlu diperhatikan agar tidak memiliki kerentanan, salah satunya yaitu kerentanan melalui *port* yang terlalu banyak dibuka sehingga dapat menimbulkan para *attacker* untuk mencoba melakukan pengujian pada sistem tersebut. Dalam beberapa hal yang disebutkan diatas yaitu berupa kerentanan yang ditimbulkan melalui *website* maupun *server* tentu saja sangat perlu didukung pengamanan yang baik yaitu dengan menambahkan beberapa jenis *firewall* yang juga berfungsi untuk meningkatkan keamanan pada *webserver*.

Data yang dihasilkan malalui *scanning* menggunakan *nmap* terhadap *website* Rumah Sakit Umum Daerah Palembang Bari yakni terdapat serangan (*Structured Query Language Injection),* untuk menutupi celah serangan (*Structured Query Language Injection)* pada *server* seperti ancaman manipulasi tampilan *(web deface),* pengambilan hak akses pada *server* dan lain-lain, maka digunakan teknik *Confidentiality, Integrity, Availability (CIA)* teroptimalisasinya *webserver* tersebut, lalu untuk melakukan uji coba *webserver* yang sudah teroptimalisasi dengan teknik *Confidentiality, Integrity, Availability (CIA)* maka digunakan pula teknik *Penetrastion Testing* untuk menguatkan bahwa *webserver* tersebut sudah teroptimalisasi.

2. METODOLOGI

2.1. Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini menggunakan metode tindakan atau *action research.* menurut Masyhuri dan Zainudin (2009) "*Action research* adalah penelitian untuk mengembangkan keterampilan-keterampilan baru atau cara pendekatan baru atau cara pendekatan baru untuk memecahkan masalah di dunia kerja atau dunia terapan lain. Adapun tahapan penelitian yang merupakan siklus dari *action research* ini yaitu :

1. Melakukan diagnosa *(Diagnosing),* Melakukan identifikasi masalah-masalah pokok yang ada guna menjadi dasar dalam melakukan penelitian pada suatu organisasi sehingga dapat memberikan perubahan lebih baik. Pada tahap ini peneliti melakukan diagnosa terhadap parameter yaitu terdapat sebuah serangan pada situs *website* Rumah Sakit Umum Daerah Palembang BARI. Maka akan dilakukan sebuah optimasi dari segi keamanan *webserver* sehingga sistem tersebut dapat lebih baik.
2. Membuat rencana tindakan *(Action Planning),* Peneliti memahami masalah yang ada kemudian dilanjutkan dengan menyusun

rencanan tindakan dalam menyelesaikan masalah pada *webserver* sehingga diharapkan mampu menutupi kerentanan yang ada pada *webserver* Rumah Sakit Umum Daerah Palembang BARI.

3. Melakukan Tindakan *(Action Taking),* Peneliti melakukan tindakan disertai impementasi rencana yang telah dibuat dengan melakukan pengamanan pada *webserver* sehingga dapat menutupi kerentanan terhadap celah-celah pada *webserver* tersebut. Selanjutnya akan dilakukan optimasi agar sistem dapat berjalan lebih baik dari sebelumnya dan terakhir dilakukan pengujian kembali dengan menggunakan teknik *penetration testing (pentest)* untuk mengetahui sistem tersebut sudah tidak memiliki kerentanan.
4. Melakukan Evaluasi *(Evaluation),* peneliti melakukan evaluasi hasil dari kelemahan-kelemahan yang ada pada *webserver* dalam bentuk laporan,
5. Pembelajaran *(Learning),* Setelah melakukan analisis yang dianggap sudah cukup, kemudian peneliti mendapatkan rapot tentang kelemahan-kelemahan pada *webserver* tersebut. Selanjutnya dapat memberikan masukan akan pentingnya pengamanan pada sebuah sistem *webserver*.

2.2. Metode Pengumpulan Data

Data yang akan diolah untuk melakukan penelitian ini di dapatkan dengan studi *liberator* dan observasi serta melakukan wawancara untuk mendapatkan data yang lebih memadai, adapun metode pengumpulan data, yaitu sebagai berikut:

1. Studi kepustakaan *(Literature)*

   Data yang diperoleh guna mencapai tujuan penelitian ini yaitu melakukan studi kepustakaan dalam mencari bahan seperti buku, skripsi dan juga mencari beberapa jurnal,ebook yang ada diinternet sesuai dengan objek yang dieliti.
2. Penelitian *(Observasi)*

   Data yang dikumpulkan yaitu melihat secara langsung dari objek yang diteliti dan juga mengetahui informasi-informasi yang didapat dari tempat objek dan juga kerentanan yang ada di situs-situs pemerintahan yang ada di internet.
3. Wawancara

   Untuk mendukung dalam penelitian ini, maka dilakukan tahap wawancara kepada ketua SIM RS guna mendapatkan informasi sebagai pendukung data penelitian dan juga mengetahui kendala-kendala yang ada didalam sistem *webserver* yang mereka bangun.

2.3. Macam-macam penyerang

Berikut adalah macam-macam penyerang keamanan *situs web* serta motif mereka menyerang menurut (Richard Pangalila, 2012:13) adalah :

1. *White Hat Hacker*

   Menurut (Hidayat&Sopyan,2007:7) *White hat hacker, hacker* dengan topi putih, adalah tokoh-tokoh yang mengagumkan dari segi pencapaian teknis dan filosofis mereka yang turut mengembangkan budaya *hacker* di dunia. Ini adalah tokoh-tokoh yang ikut mendorong banyak revolusi dalam dunia komputer dan teknologi informasi. *White hat hacker* sendiri banyak membantu instansi dan perusahaan maupun negara dengan memberi

tahu kelemahan-kelemahan sistem yang mereka miliki.

2. *Black Hat Hacker*

   Black hat hacker ini adalah kebalikan dari *white hat hacker*. Ini adalah orang yang suka menembus suatu jaringan komputer dengan tujuan buruk seperti mengambil data/informasi dan bahkan merusak data/informasi tersebut.*Black hat hacker* ini biasanya juga disebut *cracker*.

3. *Cyber Criminal*

   Tujuan utamanya mencuri uang korbannya dengan menggunakan malware. *Cyber criminal* biasanya terdiri dari kelompok yang besar. Macam-macam cara dan objek yang dilakukan untuk melancarkan aksi jahatnya. Misalnya, memakai kartu kredit milik korban, memanipulasi akun bank, mencuri identitas seperti kata sandi korbannya.

4. *Spammer*

   Ini adalah orang-orang yang menyebarkan spam ke suatu aplikasi berbasis *web*. Tujuannya macam-macam, ada yang hanya iseng maupun yang mengiklankan sesuatu secara ilegal lewat *spam* tersebut.

5. *Advanced Persistent Threat (APT) agent*

   Tujuan mereka ialah mencuri *property* intelektual sebuah perusahaan untuk mendapatkan uang dengan cara cepat. Mereka menjiplak ide dari perusahaan lain dan menjual informasi yang mereka dapat ke perusahaan lain.

6. Mata-mata perusahaan

   Hampir mirip seperti *APT agent*, namun bedanya mereka tidak terorganisir seperti *APT agent*.

7. *Hacktivist*

   Hacker tipe ini melancarkan aksinya dengan latar belakang politik, agama, lingkungan hingga keyakinan. Biasanya mereka mempermalukan lawannya atau mengobrak-abrik situs mereka.

8. *Cyber warriors*

   Cyber warrior berperan dalam perang *cyber* di mana suatu wilayah suatu negara melawan wilayah negara lain dengan tujuan akhir melumpuhkan kemampuan militer melawan.

9. *Rogue Hackers*

   Mereka melakukan aksi *hacking* hanya untuk membuktikan kemampuan mereka, menyombongkan diri ke teman atau hanya karena merasa tertantang dalam melakukan aksi ilegal. Aksi mereka memang menggangu namun mereka tidak menggangu maupun merusak bisnis orang lain.

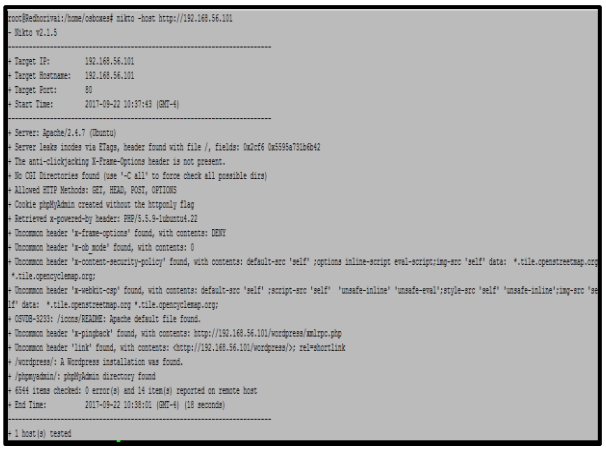
3. Hasil

3.1 Mengamankan Celah Pada *Webserver*

Scanning menggunakan nikto untuk mendapatkan informasi mengenai *webserver* yang menjadi target, dengan meggunakan perintah sebagai berikut :

```
#nikto –host http://192.168.56.101/
```

Fungsi dari perintah diatas yaitu mengetahui berapa banyak jumlah pada *host* dan mengetahui beberapa informasi secara umum mengenai *header* pada *webserver* tersebut seperti pada gambar dibawah ini.

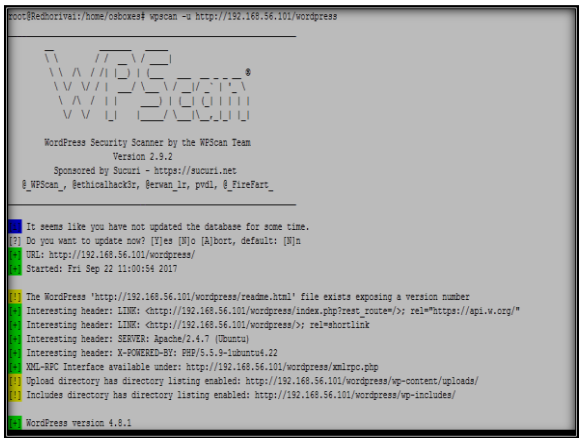


Gambar 1. *scanning* menggunakan *Nikto*

Tidak banyak informasi yang bisa didapatkan dari hasil *scanning* dengan menggunakan *tool nikto,* informasi yang diperoleh berupa *phpmyadmin, wordpress* dan *header* . Perlu diketahui bahwa *phpmyadmin* merupakan salah satu informasi penting yang bisa didapat oleh para *attacker*. Setelah mendapatkan beberapa informasi dari *nikto,* dapat diketahui bahwa situs *website* pada Rumah Sakit Umum Daerah Palembang BARI, menggunakan CMS *(Content Management System)* yaitu *wordpress.* Dengan mengetahui hal tersebut, maka dilanjutkan dengan melakukan *scanning* di situs *website* tersebut dengan menggunakan *tools wp scanning. Wordpress scanning* atau yang sering dipanggil yaitu *wpscan* merupakan *tools open source* untuk mengetahui informasi yang ada pada situs *website wordpress* berupa informasi secara umum. Dengan mengetikan perintah pada terminal yaitu :

```
#wpscan –u http://192.168.56. 101/wordpress
```

Perintah diatas berfungsi untuk mengetahui informasi mengenai situs *wordpress* berupa versi *wordpress, apache* dan *server.*



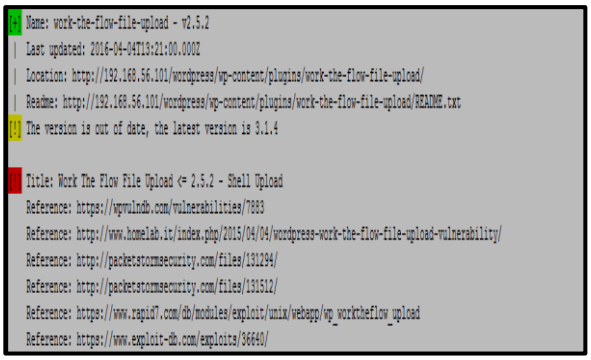
Gambar 2. *Scanning* Menggunakan *Wpscan*

Informasi yang diperoleh berupa jenis *Webservice* yang dipakai yaitu *apache* dengan versi 2.4.7, *PHP (personal home page)* dan menggunakan sistem operasi jenis *ubuntu.* diketahui juga bahwa versi *wordpress* yaitu 4.8.1.

Untuk mendapatkan informasi yang lebih dalam guna mengetahui celah keamanan pada *wordpress* tersebut, selanjutkan akan dilakukan *scanning plugin. Tools* yang dipakai tetap menggunakan *wpscan* akan tetapi dengan menambahkan beberapa perintah sehingga dapat dihasilkan beberapa jenis *plugin* yang dipakai pada *wordpress* tersebut. Berikut ini merupakan perintah untuk melakukan *scanning plugin* yaitu:

```
#wpscan –u http://192.168.56.101/wordpress –enumerate vp
```

Perintah diatas berfungsi untuk mengetaui jumlah *plugin* yang dipasang pada *website wordpress* dan juga dapat mengetahui kerentanan pada *plugin* tersebut.



Gambar 3. *Scanning Plugin*

Dapat diketahui *plugin* yang dipakai pada *wordpress* tersebut terdapat peringatan yaitu adanya *shell upload* dengan jenis *plugin work the flow file upload* dan juga beberapa referensi tentang cara melakukan *shell upload* pada *wordpress* yang menggunakan *plugin* tersebut. Hal inilah yang dapat dilakukan oleh *attacker* untuk melakukan *penetration testing (pentest)* dengan upaya melakukan penerobosan pada *website wordpress* tersebut.

Dengan merubah nama-nama direktori khususnya pada *plugin work the file upload* sehingga akan menyulitkan para *attacker* untuk melakukan pencarian lokasi kerentanan tersebut.

Wp-content/plugins/work-the-flow-file-upload/public/assets/jQuery-File-Upload-9.5.0/server/php

Dapat kita ketahui ini adalah direktori *default* pada *plugin work the flow file* yang memiliki kerentanan, tentu saja para *attacker* dengan mudah masuk melalui celah tersebut. Oleh karena itu cara pertama adalah melakukan penggantian nama pada direktori-direktori untuk menyulitkan para *attacker* mencari lokasi *plugin* nya. Penulis melakukan perubahan nama direktori sebagai berikut.

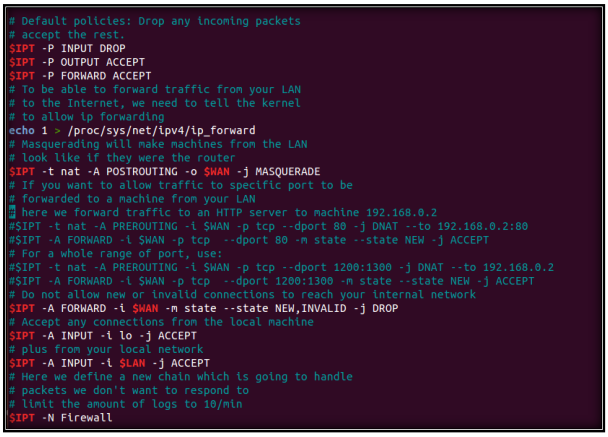
Wp-content/plugins/W0rk_The_Fl0w_File_Upl0ad/publiC/AssetS/jQuery-File-Upl0ad/server/php

Dengan mengganti beberapa kata yang akan menyulitkan para *attacker* untuk melakukan pencarian direktori yang memilki kerentanan. Beberapa nama direktori yang telah berubah dapat memberikan dampak yang lebih baik untuk memberikan kesulitan kepada para *attacker*.

3.2. Optimasi *Webserver*

3.2.1. Menambahkan *Firewall*

Berdasarkan hasil dari analisis sebelumnya, tidak terdapatnya *firewall* khususnya untuk menghindakan serangan dengan menggunakan teknik *ddos*, selanjutnya penulis melakukan evaluasi pada sistem yang mereka bangun, dengan melakukan penambahan *script* untuk anti *ddos*, dan menambahkan *script Iptables* untuk kembali meningkatkan keamanan pada sistem tersebut, berikut *script* yang ditambahkan oleh penulis kedalam sistem *webserver* tersebut.



Gambar 4. *Script* Anti *DDOS*

Pada gambar diatas merupakan *script* yang penulis gunakan untuk membuat anti *ddos* dengan menggunakan *script* sederhana, sehingga dapat membuat sistem lebih tangguh dalam mengantisipasi serangan dari *ddos*. *Script* anti

ddos juga biasanya sudah disediakan oleh konsultan IT untuk membantu para admin dalam meningkatkan keamanan pada sistem tersebut. Setelah selesai melakukan penambahan *script* anti *ddos* tersebut, penulis kembali melakukan penambahakn *script IPtables* agar sistem tersebut mempunyai tingkat keamanan secara optimal.

No	Tindakan	Keterangan
1.	*Script Firewall* Anti *DDOS*	Menambahkan *script firewall* yang berfungsi untuk anti *ddos*
2.	*Script Iptables*	Menambahkan *script iptables*
3.	*Mod Security*	a. Mengaktifkan *secrule engine* b. Mengaktifkan *secrule upload* c. Mengaktifkan *secrule debug log* d. Mengaktifkan *secrule audit log*

Gambar 5. Hasil Optimasi

3.3. *Penetration Testing Webserver*

Pada tahap ini untuk melihat sebuah data yang diperoleh hasil pada tahap *pentest* sebelumnya terhadap *website*, maka dilanjutkan dengan melakukan *pentest* pada *server* nya. Dapat diketahui bahwa terdapat kerentanan pada *FTP (file transfer protocol)* dan *SSH (secure shell)* yaitu mudah ditebak *username* maupun *password* dengan menggunakan teknik *bruteforce.* Penulis kembali melakukan pengujian dalam kedua hal tersebut dengan melakukan teknik *brutefoce* yang dibantu oleh *tools* yang ada di *linux bacbox* dan juga dibantu sebuah *file* berupa kombinasi *password* umum yang telah di unduh.

Gambar 6. Melakukan Pengunduhan *Wordlist*

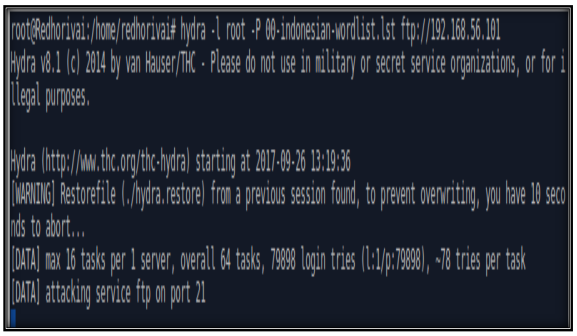
Setelah *wordlist* yang telah diunduh selesai, maka akan dilanjutkan teknik *bruteforce* dengan menggunakan *tools hydra* yang telah tersedia di *linux backbox*. *Tool hydra* sendiri merupakan *tool* yang disediakan secara gratis, bahkan para pakar IT selalu menggunakan *tool hydra* untuk membantu mereka dalam melakukan pengujian pada sistem yang ada.

2. *Privilege Escalation*

   Privilege escalation merupakan tahap konsep *penetration testing (pentest)* selanjutnya, yang berfungsi untuk melakukan *cracking password* dan juga untuk melihat ketangguhan dalam kombinasi *password* yang ada. Berikut ini merupakan perintah *hydra* pada terminal:

```
#hydra –l user –P wordlist ftp://ip-target
```

   Perintah *hydra* diatas yaitu melakukan *brutefoce* pada *FTP (file transfer protocol)* dengan dibantu oleh *wordlist* serta alamat *IP* yang dituju berikut merupakan hasil dari *bruteforce:*

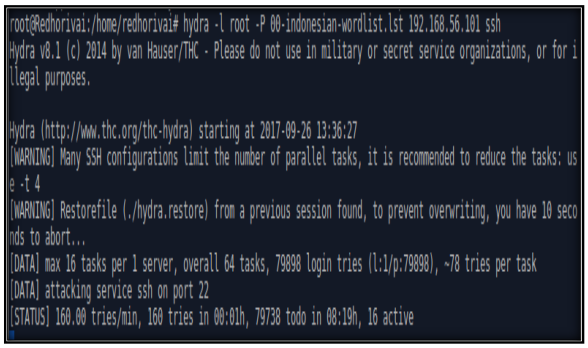


Gambar 7. *Brutefoce FTP* menggunakan *Hydra*

Berdasarkan gambar diatas akan dijelaskan sedikit mengenai maksud dari perintah pada terminal tersebut yaitu penulis melakukan *brutefoce* pada *username root* dengan ditambah perintah –P singkatan dari *password* yang telah disediakan dalam bentuk teks serta alamat *ftp* yang menjadi target. Setelah itu tinggal menunggu apakah dalam *wordlist* tersebut terdapat *password* yang dipakai oleh *admin*, tentu saja proses *bruteforce* ini memerlukan waktu yang cukup lama dikarenakan *tools* tersebut akan melakukan tebakan *password* sampai semua kombinasi *password* yang ada di *wordlist* tersebut telah dicoba semuanya. Tidak jauh halnya dengan melakukan *brutefoce* pada *ssh (secure shell)* dengan sedikit merubah perintah pada terminal sebagai berikut :

```
#hydra –l root –P wordlist ip-target ssh
```

Konsep dari perintah diatas sama seperti yang dijelaskan pada *bruteforce ftp* sebelumnya, hanya saja setelah *ip* target ditambahkan perintah *ssh* sehingga didapatkan hasil seperti gambar dibawah ini :



Gambar 8. *Bruteforce SSH* Menggunakan *Hydra*

Seperti yang dijelaskan sebelumnya, konsep dari *tools* ini sama saja yaitu akan selalu mencoba semua kombinasi *password* dalam bentuk teks yang dinamai yaitu *wordlist.* Sehingga waktu yang dilakukan untuk *bruteforce ssh* tersebut bisa lama tergantung ketepatan dalam tebakan *password* didalam *wordlist* tersebut.

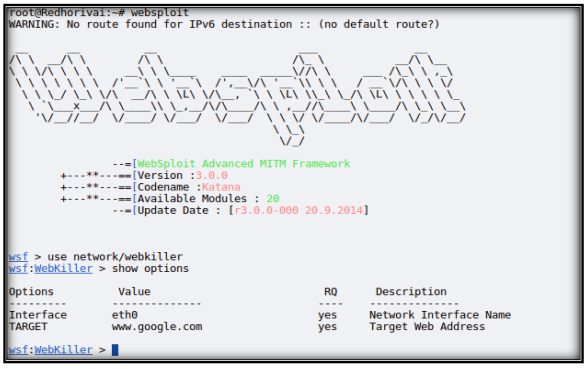
3. *Denial Of Service*

   Setelah melakukan teknik *bruteforce* dengan menggunakan *tool hydra*, selanjutnya penulis mencoba menguji ketahanan sistem dengan menggunakan teknik *ddos attack. DDOS (Distributed Denial Of Service Attack)* sebuah usaha yang membuat suatu sumber daya computer menjadi tidak bisa dipakai oleh *user* nya, dengan menggunakan ribuan *zombie system* yang menyerang secara bersamaan yang bertujuan untuk membuat sebuah *webserver* atau layananan *online* tidak bisa melakukan pekerjaan dengan *effisien* atau bahkan mati sama sekali untuk sementara waktu atau selama-lamanya. Berikut ini merupakan 5 tipe dasar *ddos attack :*

1. Penggunaan berlebihan sumber daya komputer berupa *bandwith, disk space,* atau *processor.*
2. Gangguan terhadap informasi konfigurasi, seperti informasi *routing.*
3. Gangguan terhadap informasi status, misalnya memaksa me-*reset TCP session.*

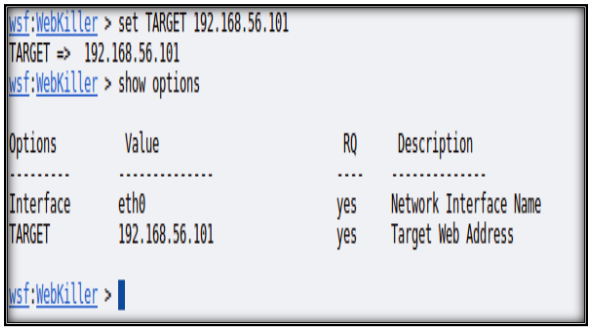
4. Menghalang-halangi media komunikasi antara computer dengan user sehingga mengganggu komunikasi.
5. Gangguan terhadap komponen-komponen fisik pada *network*.

Dari apa yang dijelaskan mengenai 5 dasar tipe *ddos attack*, maka dari itu penulis melakukan teknik tersebut dalam upaya melihat ketangguhan dalam sistem tersebut. *Websploit* adalah salah satu aplikasi untuk melakukan serangan *ddos* dapat melakuka serangan dalam skala cukup besar, *websploit* juga merupakan aplikasi yang bisa di *download* secara gratis sehingga para konsultan IT dapat menggunakannya dalam upaya menguji ketangguhan sistem yang mereka bangun. Berikut merupakan perintah dalam melakukan serangan *ddos* dengan menggunakan *websploit* dengan menggunakan sistem operasi *backbox* .



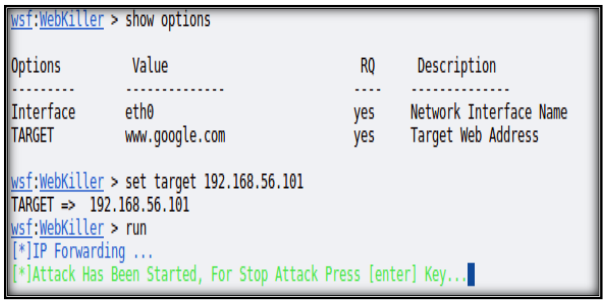
Gambar 9. *Websploit*

Gambar diatas merupakan tampilan pada *tools websploit* dengan mengetik perintah *websploit* pada terminal *bacbox*, selanjutnya pada perintah *use network/webkiller* merupakan metode yang dipakai didalam penggunaan *websploit.* Sebelum melakukan penyerangan dengan menggunakan *ddos* diharapkan periksa *ip* target dengan melakukan perintah *show options.* Dapat diketahui bahwa pada bagian *show options* memiliki dua cara yaitu dapat mengunci target melalui *ip address* maupun alamat *website*, sehingga *tool* ini memang sangat memudahkan para *attacker* untuk melakukan pengujian ketahanan sistem tersebut dari serangan *ddos.* Penulis menggunakan *ip address* dalam mencoba melakukan serangan *ddos,* berikut merupakan tampilan serangan *ddos* menggunakan *websploit.*



Gambar 10. Mengunci Target *DDOS*

Setelah target terkunci dengan menggunakan *ip address* selanjutnya tinggal menjalankan perintah *run* pada terminal, sehinga *websploit* dapat langsung menjalankan tugasnya untuk melancarkan serangan *ddos* pada sistem yang telah dijadikan target, akan tetapi untuk bagian pilihan dalam mengunci target serangan khususnya menggunakan *ip address* , *attaker* harus mengetahui *interface* yang digunakan oleh sistem tersebut, sehingga pada saat *tool websploit* dijalankan tidak mengalami kesalahan dalam target yang dipilih. Berikut tampilan serangan *ddos* setelah dijalankan.



Gambar 11. *DDOS* Menggunakan *Websploit* dijalankan

Setelah menjalankan *ddos* tersebut, tentu saja *webserver* akan mendapatkan ribuan serangan dari *zombie-zombie* dalam membanjiri jaringan tersebut, sehingga *webserver* tersebut mengalami penurunan akses atau lambat dalam mengakses *website* maupun *server* pada sistem. Sebenarnya dalam melakukan serangan menggunakan *ddos* tidak hanya tersedia melalui *script* atau *tool* yang ada pada *linux,* akan tetapi untuk sistem operasi *window* juga tersedia macam-macam aplikasi untuk melakukan serangan *ddos* seperti *Loic (Low Orbit Ion Cannon), Hoic (High Orbit Ion Cannon), Hulk, UDP Flooder, Rudy (R-U-Dead-Yet), ToR's Hammer, Pyloris, OWASP Switchblade, Davoset, DDOSIM.* Berikut ini hasil *penetration testing (pentest)* terhadap *server* sebagai berikut :

3.3. Evaluasi

Berdasarkan hasil dari *Vulnerability Assesment* dengan menggunakan beberapa aplikasi dan teknik *confidentiality, integrity, availability (CIA),* optimasi dan *penetration testing* untuk melakukan penutupan celah, serta melakukan optimasi agar sistem dapat bekerja secara optimal, dan melakukan pengujian ulang agar mengetahui apakah sistem sudah tidak memiliki celah seperti sebelumnya. Berdasarkan hasil dan pembahasan yaitu adanya kerentanan pada *website* khususnya *plugin work the flow file upload,* penulis berhasil melakukan *exploit* berupa file *upload* dengan menggunakan teknik *bypass* yang didapat pada proses penelitian. Dengan adanya kerentanan ini penulis melapor kepada pimpinan SIM-RS yang bertanggung jawab atas situs *web* rsudpbari.palembang.go.id. agar nantinya cepat dilakukan penindakan dalam upaya melakukan penutupan celah pada *plugin* tersebut. Selain pada *website,* peneliti juga menemukan konfigurasi *anonymous file transfer protocol (ftp)* yang aktif, sehingga setiap *user* mampu melakukan login menggunakan *user anonymous* serta mengetahui kombinasi *password* yang lemah, yang rentan terhadap serangan *bruteforce.*

Kerentanan berikutnya adalah tidak adanyan *firewall* yang membantu untuk mengoptimalkan sistem tersebut untuk bekerja lebih baik dan terhidar dari serangan seperti *DDOS (distributed denial of service)* yang dapat membuat sistem tersebut menurun dalam segi akses. Dalam keseluruhan *webserver* tersebut sebenarnya sudah memiliki penanganan yang baik pada tema *website,* serta telah melakukan pembaruan versi *website* dalam upaya serangan seperti *SQL injection* yang merupakan teknik serangan yang sangat berbahaya yang dapat merusak *databases* suatu *webserver.* Dengan menambahkan beberapa *firewall anti ddos, iptables,* serta melakukan instalasi *mod security,* diharapkan sistem tersebut dapat bekerja secara optimal, dan juga mampu melakukan *management* data keluar maupun masuk secara lebih baik.

3.4. Pembelajaran

Hasil yang didapat pada tahap melakukan tindakan, yaitu terdapatnya celah pada *plugin* tema *work the flow file upload.* tidak hanya kerentanan pada *plugin,* akan tetapi lemahnya kombinasi *password* dan juga tidak adanya beberapa *firewall* yang dapat membuat sistem tersebut dapat bekerja secara optimal. Pengelola

sistem harus selalu memperhatikan sistem yang mereka kelola, dimana seorang *admin* sangat perlu melakukan pembaruan pada *plugin,* menambahkan beberapa *script firewall* dan juga selalu melakukan pengujian pada sistem yang mereka jaga. *Plugin* yang ada pada *website* harus diperhatikan khususnya dalam memperbarui dan juga seorang admin harus dapat mengetahui apakah *plugin* yang mereka pasang pada *website* terdapat kerentanan, tentu saja sangat dianjurkan untuk melakukan pengujian sistem secara berkala agar sistem tersebut dapat dilakukan perbaikan kembali ketika pada saat pengujian terdapat kerentanan. Tidak hanya selalu melakukan pembaruan pada *plugin* akan tetapi diperlukan untuk selalu menambah jenis-jenis *script firewall* agar sistem tersebut dapat berjalan secara optimal dan tidak mengalami penurunan pada saat mengakses sistem tersebut. Dengan melakukan beberapa tahap tadi, diharakan dapat memberikan pembelajaran untuk selalu memperhatikan sistem dan selalu menjaga sistem tersebut. Berikut ini merupakan beberapa rekomendasi pencegahan yang dapat dilakukan untuk menjaga *webserver* tersebut :

1. Selalu melalukan perbaikan secara rutin baik pada *website* maupun pada *server* sehingga *webserver* dapat terjaga dari serangan para *attacker.*
2. Jangan menggunakan *plugin* dan tema yang gratisan, dan sebelum melakukan *instalasi* baik pada *plugin* maupun tema diharapkan mencari terlebih dahulu apa *plugin* dan tema tersebut memiliki kerentanan.
3. Melakukan *chmod* pada folder yang ada di *server* untuk memberikan kebatasan *user* dalam mengakses suatu direktori dan dokumen.
4. Melakukan perubahan nama direktori pada *plugin* agar para *attacker* tidak bisa melakukan *bypass file upload.*
5. Memberikan kombinasi *password* yang baik, serta memperhatikan *port* yang terbuka, jika ada beberapa *port* yang tidak diperlukan untuk diakses lebih baik dilakukan penutupan.

Menambahkan beberapa *script firewall* untuk meningkatkan keamanan pada *webserver* sehingga *webserver* tersebut mampu menjalankan tugasnya secara optimal.

4. Simpulan

Berdasarkan hasil dari penelitian dan pembahasan yang telah diuraikan pada bab sebelumnya, dalam penelitian yang berjudul Optimasi Keamanan *Webserver* Rumah Sakit Umum Daerah Palembang BARI (rsudpbari.palembang.go.id), maka dapat disimpulkan :

1. Teknik pengamanan *Confidentiality, Integrity, Availability (CIA)* dapat membantu secara signifikan pada situs *website* rsudpbari.palembang.go.id
2. Beberapa *script firewall* anti *DDOS (Distributed Denial Of Service Attack)* dan *script iptables* dapat mengoptimalkan *webserver* dari serangan *DDOS (Distributed Denial Of Service Attack).*
3. Lemahnya kombinasi *password* baik menggunakan *username SSH (secure shell)* serta *FTP (File Transfer Protocol).*

4. Teknik Pengujian *Arbitrary Upload* pada *plugin work the flow file upload* berhasil dilakukan khusunya pada *bypass file upload*.

Hasil penelitian ini memberikan kontribusi saran mengoptimalkan keamanan *webserver*

DAFTAR RUJUKAN

Ferdiansyah, 2013. "***Penetration Testing* Menggunakan *SQL Injection"*** Palembang, Jurnal Penelitian.

Pangalila, Ricard 2015. "**Uji Celah Pengamanan *Penetration Testing (pentest)"*** Yogyakarta penerbit : CEH.

Gurpreet Chief Dhillion. 2013. "***The CIA Strikes Back: Redefining Confidentiality, Integrity and Availability in Security"*** USA, penerbit : Jissec.

Arief. M. Rudyanto. 2011. "**Pemograman *Web* Dinamis Menggunakan PHP & *MySQL"*** Yogyakarta. penerbit : Andi.

Masyhuri dan Zainudin 2009. "**Metodologi Penelitian Pendekatan Praktis dan Aplikasi"** Bandung, Penerbit : PT. Refika Aditama.

Efvy Zem. 2014. "**Buku Sakti Hacker"** Yogyakarta, Penerbit : Mediakita

Lolly Amalia Abudullah. 2008. "**Keamanan *Web Server* Menggunakan IPv6"** Penerbit : Departemen Komunikasi dan Informatika.