RANCANG BANGUN APLIKASI ANALISIS STANDAR KEAMANAN *WEBSITE* DENGAN METODE *SCANNING VULNERABILITY* MENGGUNAKAN ALGORITMA *QUEUE TASK*

Ardi Mardiana[1], Qudrat Nurfajar Yasin Sutisna[2]
[1]*Jurusan Teknik Informatika, Fakultas Teknik, Universitas Majalengka*
Jl. KH. Abdul Halim No. 103 Kabupaten Majalengka Telp 0233 281496
ardimardiana@gmail.com
[2] *Jurusan Teknik Informatika, Fakultas Teknik, Universitas Majalengka*
Jl. KH. Abdul Halim No. 103 Kabupaten Majalengka Telp 0233 281496
fajarpunya99@gmail.com

ABSTRAK

Menurut laporan dari *Security Incident Response Team on Internet Infrastructure / Coordinator Center* (Id - SIRTII / CC) terdapat 205.502.159 serangan pada akhir tahun 2017, total dari seluruh aktivitas *malware* yang terdeteksi, sebanyak 37,72% berkaitan dengan serangan DOS, 20,93% merupakan *exploit*, 18% adalah trojan atau berkaitan dengan aktivitas trojan, 15% tercatat sebagai *bad unknown* dan sisanya tercatat sebagai *adware, shell code,* cnc, *misc attack*, *network scan*, dan *web application attack*.

Tools yang dibuat seperti halnya *tools Katoolin*, yaitu salah satu *repository* dari Sistem Operasi Kali Linux yang digunakan untuk *pentest* keamanan *website*, *tool* yang telah dibuat ini di dalamnya terdapat beberapa kategori untuk *Web Application Attack*, terutama pada *tool* yang dibuat buat lebih cepat untuk proses *scanning* karena bisa menggunakan *multiple scanning website* atau *url* dan dapat berjalan di *platform* Linux dan Windows, dengan menggunakan algoritma antrian (*queue task*), saat memasukan *url website* dalam jumlah yang banyak maka *url* tersebut di simpan terlebih dahulu ke dalam *file temporary* kemudian setelah itu *tools* akan membaca isi *file temporary* apakah terdapat *url / web* yang akan di analisa keamanannya jika terdapat *url / web* di dalamnya maka pada baris pertama langsung di eksekusi dan memulai menganalisa *url* tersebut setelah selesai maka *url* yang ada pada *file temporary* dihapus dan dipindahkan ke dalam *file log* atau disebut sebagai *history file* dimana *file* ini merupakan isi dari *url* atau *website* yang sudah di analisa sebelumnya dan hasil akhir atau disebut sebagai *report scanning*, hasil dari analisa masuk ke dalam *file log*, setelah semua selesai menyimpan dan *scanning* maka untuk baris selanjutnya akan dianalisa kembali dan seterusnya sehingga tidak terdapat *url* / *website* di dalam *file temporary* dan *tools* akan berhenti untuk menganalisa.

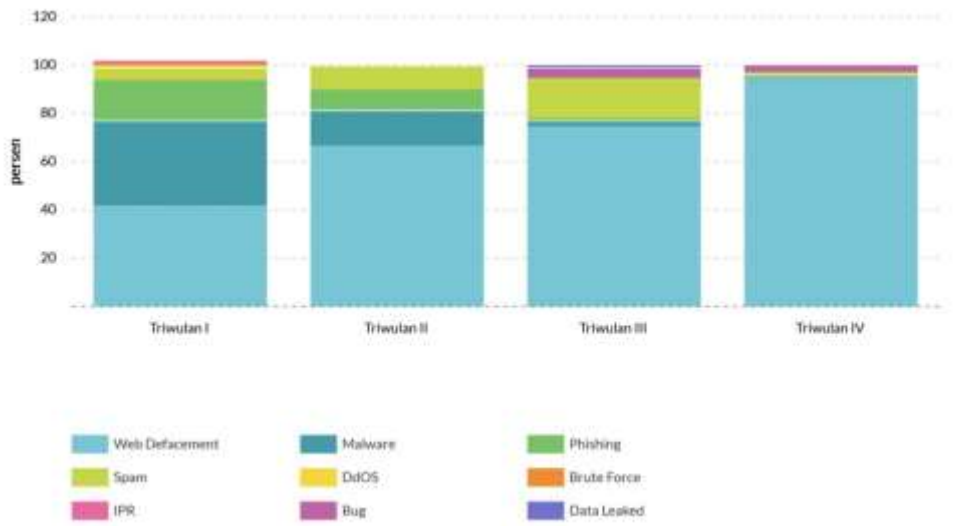
Dalam proses *scanning* terdapat beberapa proses kembali seperti menganalisa *port* yang terbuka, melakukan pencarian *url-url* di dalamnya dengan memanfaatkan pencarian *google* secara otomatis atau disebut sebagai *auto dorking*, melakukan analisa penggunaan *framework* / cms yang digunakan, mengalisa *form login*, dan lain sebagainya.

Kata Kunci: ***Hacking Tools, Pentest Tools, Web Security,*** **Pemrograman** ***Ruby, Scanning Vulnerability***

1. Latar Belakang

Indonesia pada saat ini mengalami salah satu masalah yang serius dalam kejahatan yaitu *Cybercrime*. *Cybercrime* adalah tindak kriminal yang dilakukan dengan menggunakan tekno*log*i komputer sebagai alat kejahatan utama. Di Negara-negara maju kasus kejahatan seperti ini juga marak tidak hanya terjadi seperti di Amerika dan Eropa namun juga di Negara berkembang yang ada di Asia dan Afrika.

Terutama pada *website* pemerintahan Indonesia adalah *website* yang akhir akhir ini banyak jadi target serangan kejahatan di dunia maya, Menurut laman Techno Okezone *website* pemerintahan Indonesia adalah *website* yang sangat mudah untuk diretas, berikut adalah grafik insiden peretasan *website* pemerintahan Indonesia pada tahun 2016



Gambar 1 Grafik peretasan pada *website* pemerintahan Indonesia

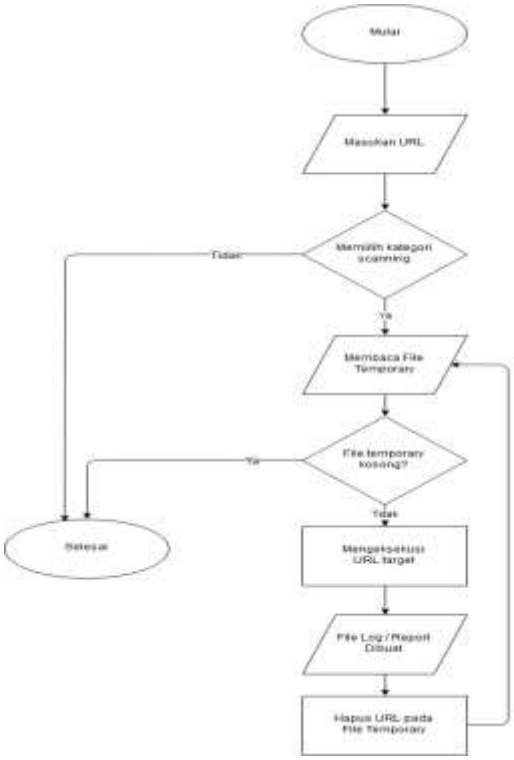
Seperti pada gambar diatas adalah grafik yang paling banyak yaitu *web defacement* yaitu dengan mengganti laman depan *website* dengan meninggalkan pesan-pesan dari seorang peretas, dan bukan hanya situs pemerintahan yang menjadi target lain adalah *website marketplace* seperti bukalapak, tokopedia, sehingga yang akan di incar adalah berupa kartu kredit, saldo / uang yang ada di situs tersebut adalah incaran para penjahat dunia maya.

Banyaknya peretasan pada situs pemerintahan karena masih banyak situs pemerintahan yang menggunakan cms yang sifatnya publik seperti *wordpress, drupal, joomla* dan lain sebagainya, padahal cms tersebut bisa dikatakan aman apabila seorang *developer* mampu dan memahami cara mengamankannya, dan banyak penggunaan *plugin* yang tidak *up to date*. Yang banyak digunakan oleh para penjahat dunia maya adalah dengan cara *random* atau bisa memanfaatkan pencarian *Google* atau disebut sebagai *Google Dork*, *SQL Injection, Local File Inclusion, tool* yang paling banyak digunakan seperti *Katoolin, Acunetix, Havij*.

2. Perancangan

a. Flowchart

Berikut adalah gambar alur *tools scanning* yang dibuat:



Gambar 2 *Flowchart Tools Scanning*

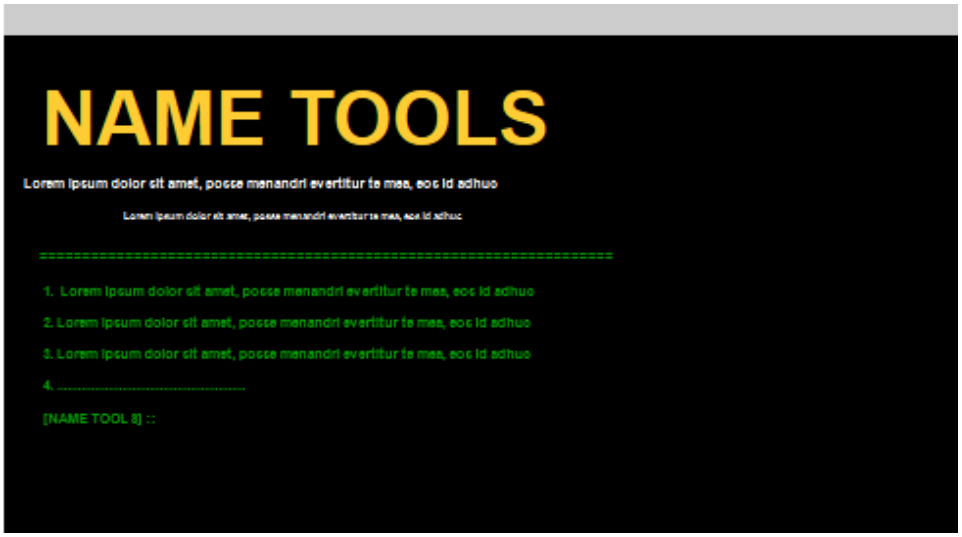
Penjelasan alur:

1. **Mulai**, Memulai menjalankan *tool*
2. **Masukan *URL***, pada tahapan ini user memasukan *url* pada *file temporary* dalam jumlah banyak dan dipisahkan atau disebut sebagai delimiter antara *url* pertama, kedua dan seterusnya dengan newline atau mengetikan *enter* untuk mengisi baris kedua
3. **Memlilih Kategori *Scanning***, pada tahap ini user memilih kategori *scanning* dengan mengetikan angka berdasarkan angka yang ada pada saat muncul pilihan kategori *scanning* jika tidak memilih dari angka yang ada maka proses akan selesai.
4. **Membaca *File temporary*,** pada tahap ini *tool* membaca *url* yang tersimpan di *file temporary*,
5. ***File temporary* Kosong?**, proses pengecekan jika *url* yang terdapat di *file* temporer kosong maka *tool* akan berhenti dan jika terdapat *url* maka lanjut ke tahap selanjutnya
6. **Mengeksekusi *URL* Target,** *tool* melakukan penetrasin terhadap *url* yang sedang berjalan
7. ***File Log* / Re*port* Dibuat,** Setelah selesai *url* di eksekusi maka, *file log* / history / re*port* akan di buat sesuai hasil dari proses sebelumnya
8. **Hapus *URL* pada *File* Temporer,** *URL* akan dihapus pada *file* temporer dan menaikan baris kedua ke baris pertama untuk di eksekusi kembali dan kembali ke proses tahapan ke-4.

b. Rancangan Tampilan

Berikut adalah beberapa rancangan pada tampilan *tool*s yang dibuat:

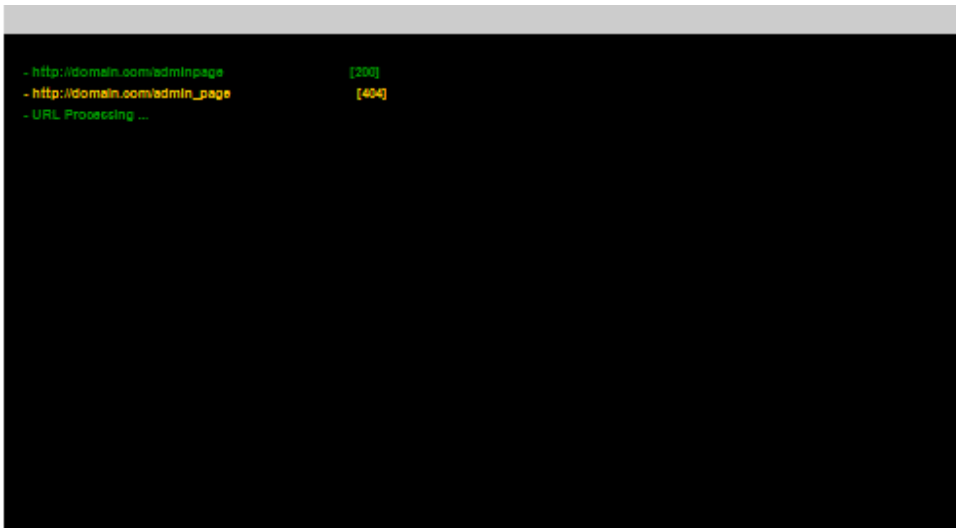
1. Tampilan Utama



Gambar 3 Tampilan Utama (*Main View*)

Gambar diatas adalah gambar rancangan untuk tampilan saat dijalankan, dan terdapat beberapa pilihan kategori dengan mengetikan nomor yang tersedia

2. Tampilan Saat Memproses



Gambar 4 Tampilan Saat *Tool* Berjalan

Gambar diatas adalah gambar rancangan saat *tool* sedang memproses

3. Pembahasan

Pada tahap pembahasan yaitu menjelaskan dan membahas tampilan dan hasil dari *tool* yang dibuat berikut adalah pembahasan-pembahasan yang akan di jelaskan:

Gambar 5 Tampilan Utama

Gambar di atas merupakan gambar tampilan utama saat *tool* mulai dijalankan dan terdapat beberapa pilihan nomor kategori

Gambar 6 *Port Scanner*

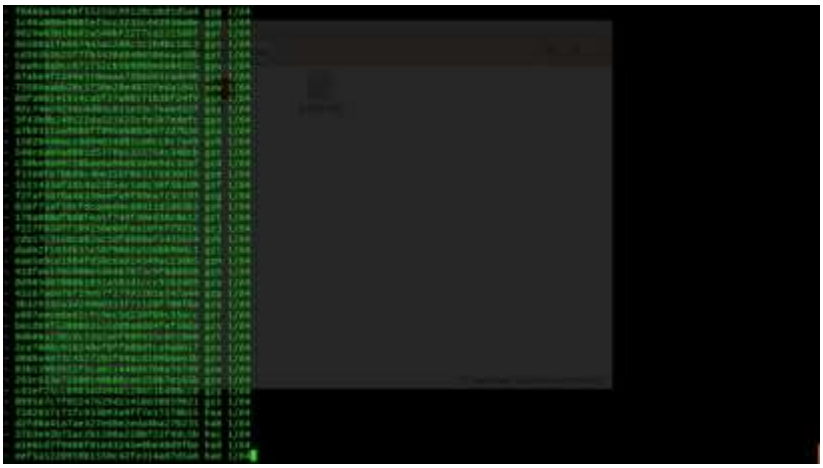
Gambar diatas merupakan tampilan saat mencari *port* yang terbuka pada *website* tujuan

Gambar 7 WP *Plugin Scanner*

Gambar diatas merupakan tampilan saat mencari *plugin* yang terinstal pada *website* yang menggunakan CMS *Wordpress*

Gambar 8 *Cpanel Dictionary*

Gambar diatas merupakan tampilan saat melakukan otomatis *log*in ke CPANEL (*Control Panel*)



Gambar 9 MD5 *Decrypter*

Gambar diatas merupakan tampilan saat mendeskripsi *password* dengan *type* MD5

Gambar 10 *Reverse IP* / Domain

Gambar diatas merupakan tampilan saat mencari *website client* pada *server* target yang dituju

4. KESIMPULAN

Berdasarkan hasil analisa yang dilakukan terhadap penggunaan aplikasi yang dibuat, maka dapat diambil kesimpulan sebagai berikut:

1. Dengan adanya *Multiple Process* maka *scanning web* bisa lebih cepat, untuk membantu *developer* mencari sebuah kesalahan atau *bug* pada *web* aplikasi yang dibuatnya, ditambah jika menjalankan aplikasi *scanning* ini dijalankan pada VPS (*Virtual Private Server*).
2. Bahasa pemrograman *ruby* tidak berjalan hanya pada satu sistem operasi, namun bahasa pemrograman *ruby* bisa dijalankan di semua sistem operasi.
3. Aplikasi *scanning* yang dibuat mudah digunakan, karena tidak terlalu banyak *argument* saat aplikasi dijalankan, karena dengan memakai *argument* saat aplikasi dijalankan, banyak lupa akan perintah atau *argument* yang harus ditambahkan.
4. Aplikasi yang dibuat telah menggunakan *versioning control* dengan git untuk melakukan *update souce code* jika terdapat pembaruan

5. Saran

Aplikasi yang dibuat masih banyak kekurangan yang dapat dikembangkan kembali, di antaranya:

1. Aplikasi yang dibuat hanya berfokus pada *scan* kerentanan *web*, dan untuk selanjutnya bisa ditambahkan seperti *Wireles Attack, Sniffing & Spoofing* dan lain sebagainya.
2. Kategori *scan* masih sedikit yang dibuat, dalam hal *scan web* tentunya banyak sekali teknik dan metode yang digunakan, maka dari itu pembuatan aplikasi ini dilakukan secara bertahap.

DAFTAR PUSTAKA

[1] Suryayusra, *Analisis Web Vulnerability pada Portal Pemerintahan Kota Palembang menggunakan Acunetix Vulnerability,Thesis*, 2014.

[2] *Maharani, MiaZattu, Henry Rossi Andrian, Setia Juli Irzal Ismail, Analisis Keamanan Website Menggunakan Metode Scanning Dan Perhitungan Metrics, Laporan Penelitian Internal Universitas Telkom, Bandung.*

[3] Stewart, Bruce, 2001, An Interview With The Creator Of Ruby, http://www.linuxdevcenter.com/pub/a/linux/2001/11/29/ruby.html,diakses pada tanggal 03 April 2018

*[4] Mandarnesia, 2017, Laporan Serangan Siber Sepanjang 2017,*http://mandarnesia.com/6831-2/ *diakses pada tanggal 04 April 2018*

[5] Debian, 2017, Ruby Programs Versus Python 3, https://benchmarksgame-team.pages.debian.net/benchmarksgame/compare/ruby.html diakses pada tanggal 04 April 2018

[6] Desmuflihah, Kustin Ayuwuragil, 2017, Hacker Ungkap Alasan Situs Pemerintahan Mudah Diretas, https://techno.okezone.com/read/2017/02/21/207/1624345/hacker-ungkap-alasan-situs-pemerintah-mudah-diretas, diakses pada tanggal 10 April 2018

[7] Persadha, Pratama, 2015, Penyebab Situs Pemerintahan Mudah Di Bobol, https://inet.detik.com/security/d-3106283/ini-penyebab-situs-pemerintah-mudah-dibobol, Diakses pada tanggal 10 April 2018

[8] Lokadata, 2016, Persentase Insiden Respon Domain go.id, https://lokadata.beritagar.id/chart/preview/persentase-insiden-respon-domain-go-id-1489890322, diakses pada tanggal 11 April 2018

[9] Kustiawan, Irwan, Vulnerability Assessment Terhadap SITU - Akademik Universitas Pasundan, Thesis (Skripsi(S1)), 2017