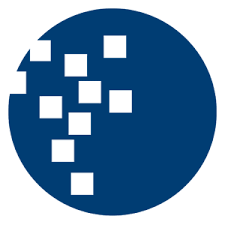
**LAPORAN CEH 2**

**DENARY**



**Anggota Kelompok:**

* Aldric Leonardo - 00000014152
* James Christian Wira - 00000014026
* Matthew Evans A. R. - 00000017942
* Vionie Laorensa - 00000013610

Tahun Ajaran 2018

**Daftar Isi**

[**I. Pendahuluan 1**](#_Toc532853265)

[**II. Penelitian Terkait 2**](#_Toc532853266)

[**III. Perancangan 3**](#_Toc532853267)

[**IV. Implementasi 9**](#_Toc532853268)

[**V. Uji Coba 18**](#_Toc532853269)

[**VI. Kesimpulan dan Saran 20**](#_Toc532853270)

1. Pendahuluan
   1. Latar Belakang Masalah

Dewasa ini perkembangan teknologi informasi sedang berkembang dengan sangat pesat. Dengan semakin berkembangnya teknologi informasi, semakin tinggi juga keperluan atas server yang ada di internet. Setiap server yang biasanya ada memiliki beberapa konfigurasi dari servis-servis yang ada di server tersebut. Servis-servis ini adalah layanan yang akan digunakan di server tersebut. Servis yang ada di suatu server dapat diketahui dengan menggunakan teknik scanning yang dilakukan pada port-port yang ada di suatu server. Dengan teknik scanning, suatu pihak yang melakukan scanning tersebut akan dapat mengetahui versi dari servis bahkan OS yang digunakan oleh server itu sendiri. Hal ini yang memulai attack di suatu server. Dengan mengetahui versi dari servis ataupun OS oleh server maka attacker akan dapat mencari dan mengetahui vulnerability yang ada di server tersebut. Servis dan OS yang ada di suatu server akan menjadi celah pintu masuk untuk attacker masuk dan menyerang server.

Penyerangan terhadap server tentu ingin dicegah oleh pemilik server. Diperlukan cara pencegahan agar attacker tidak dapat masuk ke server milik kita yaitu dengan menggunakan Intrusion Detection System (IDS). Salah satu cara yang dapat dilakukan dalam IDS adalah dengan membuat suatu server palsu, yang biasanya disebut honeypot, yang terlihat asli dan lebih menarik oleh attacker sehingga attacker akan menyerang server palsu tersebut. Dengan adanya server palsu ini, server yang asli dapat terjaga karena attacker akan masuk ke server palsu sebelum bahkan tahu ada server yang asli di sana. Server palsu ini perlu dibuat agar terlihat lebih vulnerable sehingga attacker akan langsung menyerang server palsu tersebut. Pada kesempatan ini, ingin dibuat suatu aplikasi yang dapat mempermudah pemilik server untuk membuat server palsu yang disebut honeypot tersebut dan membuat pemilik dapat tahu siapa saja yang telah masuk dan mengakses honeypot tersebut.

* 1. Rumusan Masalah

Rumusan masalah dari penelitian ini adalah sebagai berikut.

* + 1. Bagaimana merancang dan membangun aplikasi yang dapat mempermudah pembuatan honeypot?
    2. Bagaimana user aplikasi dapat mengetahui siapa saja yang melakukan akses ke honeypot yang telah dibuat tersebut?
  1. Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut.

* + 1. Aplikasi yang dibuat hanya dapat dijalankan di Sistem Operasi Linux.
    2. Aplikasi yang dibuat akan dapat membuat honeypot dengan melakukan konfigurasi di servis-servis umum saja ( seperti: FTP, HTTP, SMB, MySQL, SSH, RDP, SIP, SNMP, NTP, TFTP, TCP, Telnet, VNC ).
    3. Aplikasi memberitahukan siapa saja yang mengakses honeypot dengan memberitahukan IP apa saja yang melakukan akses ke honeypot.
  1. Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah sebagai berikut.

* + 1. Membangun aplikasi yang dapat mempermudah pembuatan honeypot.
    2. Memberitahukan user aplikasi siapa saja yang melakukan akses ke honeypot yang sudah dibuat.

Manfaat dari penelitian ini adalah sebagai berikut.

* + - 1. Mempermudah pemilik server untuk membuat honeypot sebagai IDS dari server yang dimiliki.
      2. Membuatu user aplikasi tahu lewat port mana attacker masuk. Hal ini dilakukan agar user tahu dan dapat melakukan tindakan pencegahan lainnya terhadap server.

1. Penelitian Terkait
   1. OpenCanary

OpenCanary adalah library yang ada di python yang dapat dijalankan untuk melakukan konfigurasi dari honeypot. OpenCanary akan dijalankan di virtual environment yang juga disediakan oleh python. Di virtual environment tersebut akan dibuat honeypot server yang disesuaikan dengan konfigurasi yang ditentukan dan dibuat di file konfigurasi OpenCanary ini. Bila hanya menggunakan library ini, user perlu melakukan konfigurasi secara manual terhadap file konfigurasi yang ada di OpenCanary tersebut. Bila dilakukan scan ke sistem dari server, maka akan menghasilkan hasil scanning sesuai dengan konfigurasi honeypot tersebut.

OpenCanary akan menghasilkan log file yang berupa record yang tiap recordnya adalah data JSON. Log file ini akan menampilkan source, target, dan juga yang dilakukan di honeypot tersebut. Log file ini dapat digunakan untuk mengetahui informasi dari attacker yang masuk ke honeypot tersebut.

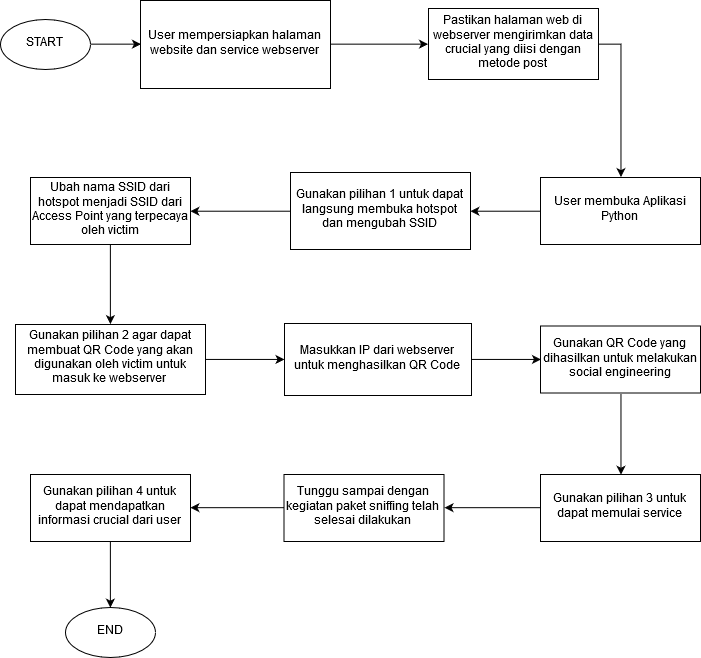
* 1. ELK Stack

ELK Stack adalah akronim untuk penggunaan tiga proyek open source yang digunakan untuk memvisualisasikan data log, yaitu Elasticsearch, Logstash, dan Kibana. Elasticsearch akan melakukan processing pada data JSON yang ada agar dapat diambil dan digunakan data-data yang penting saja. Logstash akan melakukan pipeline dari data Elasticsearch ke Kibana. Terakhir, di Kibana, data akan divisualisasikan sesuai dengan data dari Elasticsearch.

1. Perancangan
   1. Model Aplikasi
      * 1. Pengguna Hasil Penelitian:

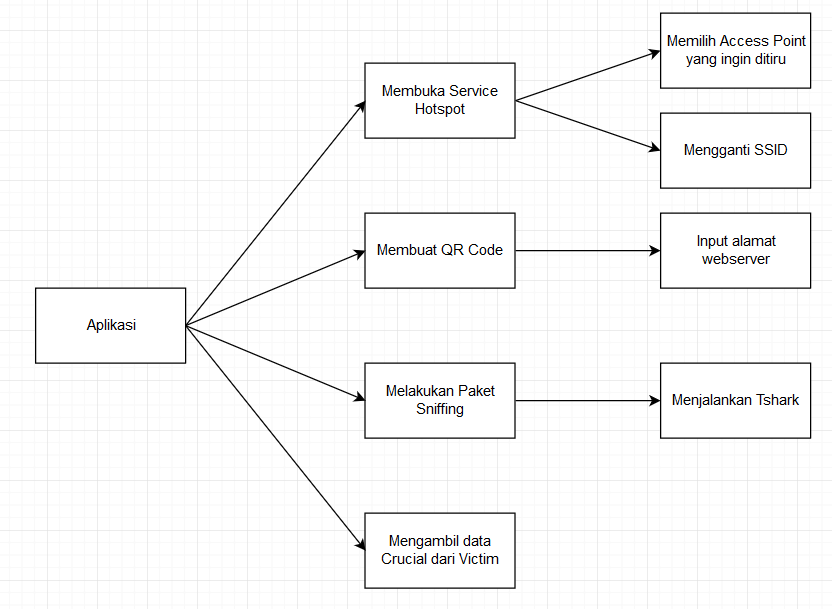
* Pemilik Server yang ingin membuat IDS Honeypot.
  + - 1. Spesifikasi Sistem:
* OS kali linux untuk menjalankan aplikasi.
* Hubungan ke jaringan untuk melakukan pembuatan server palsu.
* Aplikasi python yang telah dibuat untuk melakukan instalasi untuk menggunakan aplikasi dan menjalankan fungsi untuk membuat honeypot.
  1. Flow Chart
  2. Desain Aplikasi

Urutan *Attack* secara sederhana:



* *Attack* yang dilakukan adalah attack untuk menipu *victim* dengan memberikan suatu pancingan agar korban melakukan *scanning* QR *Code* untuk masuk ke *phising* *webserver* kita.
  + Pancingan yang digunakan harus exclusive dimiliki oleh masyarakat UMN, sehingga diberitahu dibutuhkan email dari *email* *server* UMN (*email student* dan *email* *lecturer*).
  + Di halaman di *webserver* akan ada halaman *login* SSO yang sudah mirip dengan halaman SSO yang asli (*Phising*).
* Di halaman tersebut, korban akan diminta untuk login SSO untuk dapat lanjut ke sesi selanjutnya untuk dapat berhasil mendapatkan apa yang telah dijanjikan dalam pancingan sebelumnya.
  + Halaman *web* hanya melakukan pengecekkan pada string *email* dengan label *username* untuk melakukan validasi apakah *email* yang dipakai itu adalah *email student, email lecturer*, atau bukan salah satu dari keduanya.
  + Bila validasi gagal, maka korban akan diminta untuk melakukan *input* SSO lagi dengan data yang benar.
  + Bila validasi berhasil, maka korban akan dilanjutkan ke sesi selanjutnya untuk mendapatkan apa yang dijanjikan dalam pancingan.
* Halaman *web* yang digunakan akan mengirimkan data yang diisi oleh korban, sehingga *attacker* akan berhasil mendapatkan informasi SSO dari korban.
  + Halaman web menggunakan service *Post* untuk mengirimkan data korban di *URL* yang digunakan.
* Data korban di URL akan ditangkap oleh *packet* *sniffer*. Data korban kemudian akan diexport ke dalam bentuk txt sehingga dapat diproses lagi.
  + Data SSO yang berupa *email* dan *password* kemudian akan disaring dari data di paket sniffer.
  + Dilakukan string matching URL yang memiliki elemen nama label dari form yang ada di halaman web phishing.
  + Data SSO kemudian akan di export ke dalam bentuk txt sehingga dapat digunakan lagi.

Desain Aplikasi:



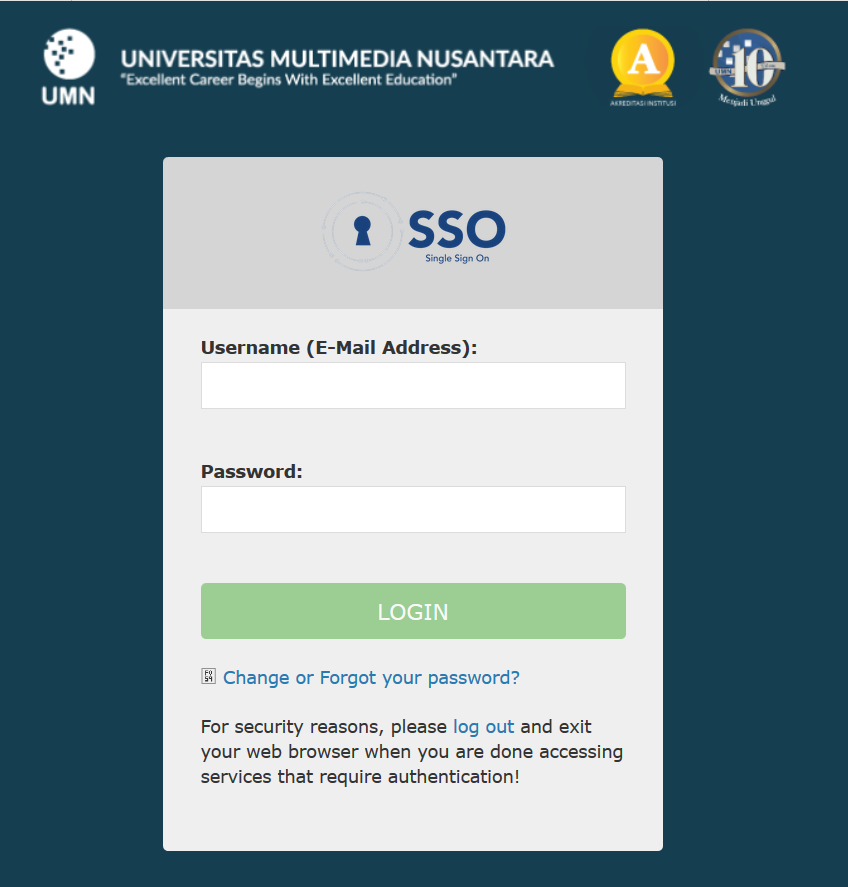
* Service membuka setting membuka Hotspot.
  + Service ini berfungsi untuk membuat Access Point Evil Twin untuk dapat menipu victim untuk terhubung dengan Access Point yang salah.
  + Aplikasi akan membuat suatu Access Point yang dapat menggantikan Access Point yang asli (Evil Twin) sehingga korban tetap dapat terhubung ke internet dengan Access Point ini.
  + Aplikasi akan dapat "menendang" orang yang terhubung dengan Access Point yang asli sehingga akan membuat korban terhubung secara paksa ke Access Point yang telah dibuat oleh aplikasi ini.
  + Di service ini, user akan dapat masuk ke jendela untuk membuka hotspot secara manual.
* Service membuat QR Code.
  + Service ini berfungsi untuk menghasilkan QR Code yang akan digunakan dalam social engineering untuk dapat membuat korban masuk ke webserver kita dan mengakses halaman Phising SSO yang akan mengirimkan isi dari yang telah diisi oleh korban.
* Service untuk melakukan Paket Sniffing.
  + Service ini berfungsi untuk melakukan paket sniffing menggunakan service wireshark / tshark.
  + Seluruh paket yang di-sniff kemudian akan disimpan dan di-export ke dalam bentuk txt.
  + URL yang dikunjungi setelah mengisi data di halaman phising oleh korban, akan dapat ditangkap dengan service ini untuk diolah kemudiannya karena halaman web mengunakan metode POST.
* Service untuk mengambil data dari korban.
  + Service ini adalah kelanjutan dan bagian dari service selanjutnya, dimana paket-paket yang telah di-sniff akan disaring sehingga hanya paket-paket yang bersangkutan yang akan digunakan di proses selanjutnya.
  + Terdapat dua elemen form yang akan diisi oleh korban saat melakukan login di halaman phising, yaitu username (email) dan password.
  + Akan dilakukan string matching untuk mengambil email dan password yang ada di URL paket tersebut.
  + Email dan password tersebut kemudian akan di-export ke dalam bentuk txt untuk kemudian dapat digunakan oleh attacker.

1. Implementasi

Implemetasi yang dibuat adalah dalam bentuk halaman phishing dan aplikasi untuk membuat Access Point dan mengambil data yang yang dikirimkan oleh halaman Phishing.

1. Halaman Phishing
2. Tampilan Awal

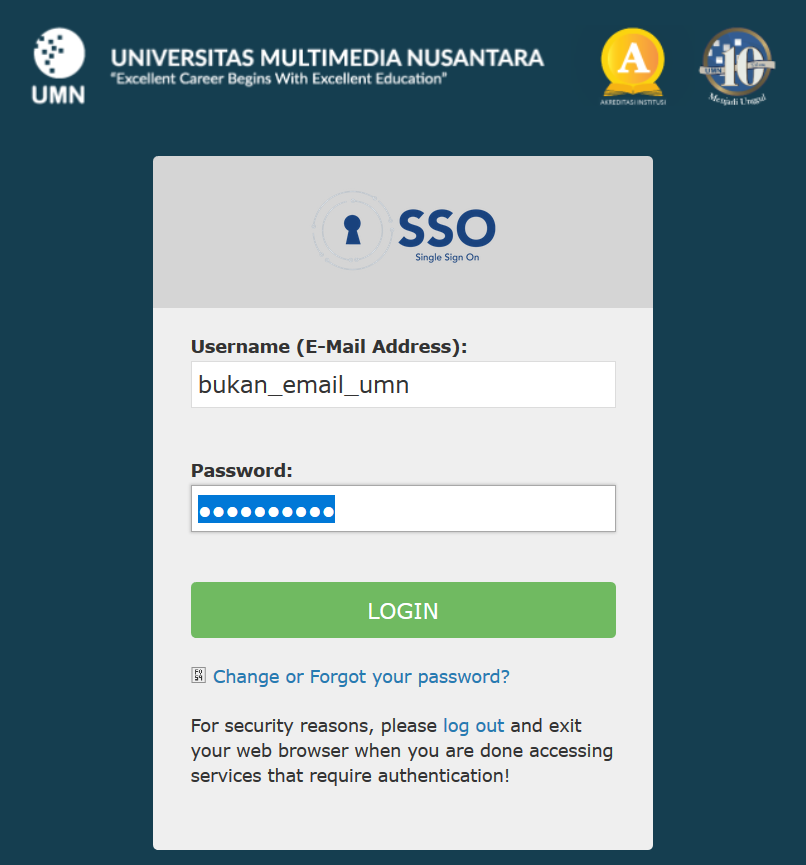
Tampilan HTML dari website diambil langsung dari SSO UMN yang asli, sehingga halaman web phising akan terlihat mirip dengan halaman yang asli.

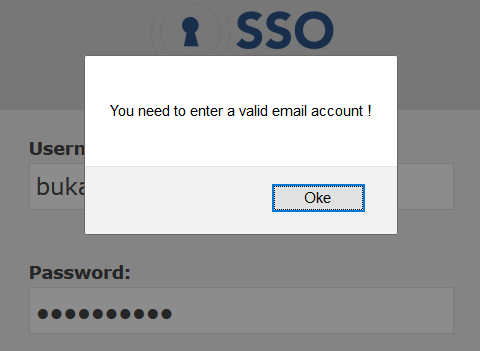


1. Validasi Input Victim

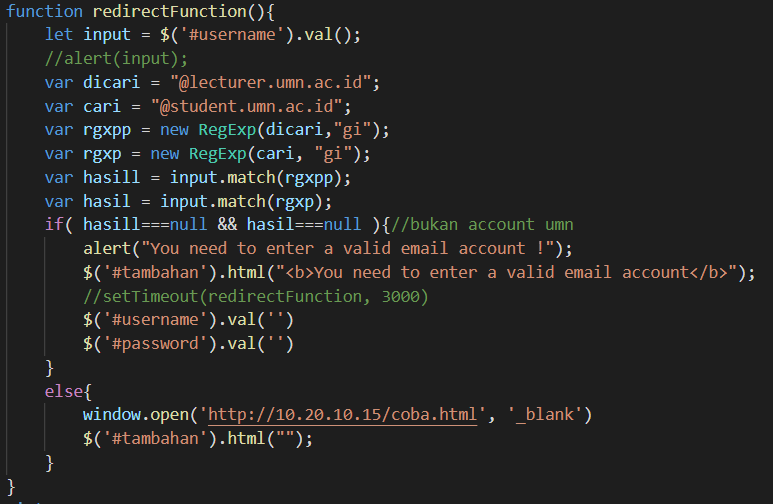
HTML akan melakukan validasi kepada input email dari form dari yang diinput dari user.

Bila input adalah email dari user yang bukan merupakan email dari email server umn maka akan muncul peringatan bahwa email yang dimasukkan salah.



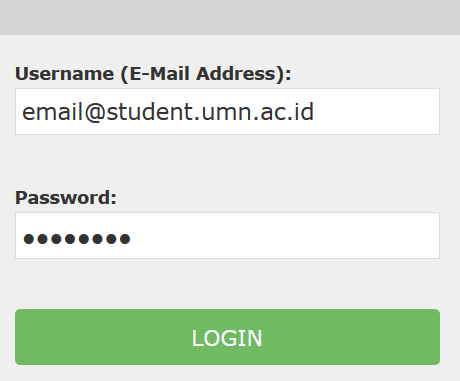


Dilakukan string matching di halaman web saat ditekan tombol LOGIN.



1. Redirect ke Halaman Bait

Bila input adalah email dari user yang dan merupakan email dari email server umn maka akan dilakukan redirecting ke halaman yang akan menampilkan hasil bait.

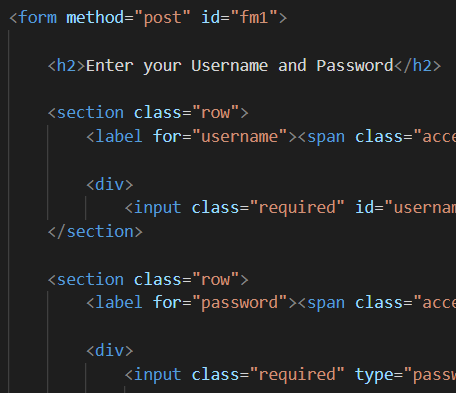




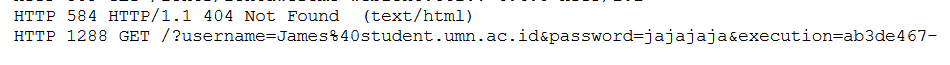
1. Pengiriman Data Input Victim

Data yang diinput di halaman phising kemudian akan dikirim ke access point.

Di HTML, digunakan method “POST” untuk dapat melakukan pengiriman data yang ada di form saat dilakukan submit.



Paket “POST” yang dikirim kemudian akan dapat diterima oleh tshark.

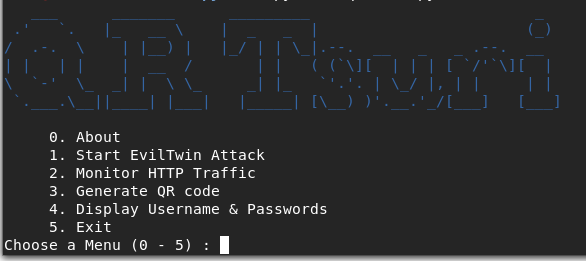


1. Aplikasi
2. Buka aplikasi python

Gunakan service python2 untuk membuka aplikasi. Pastikan requirement yang modul telah terinstall sebelum menjalankan aplikasi.



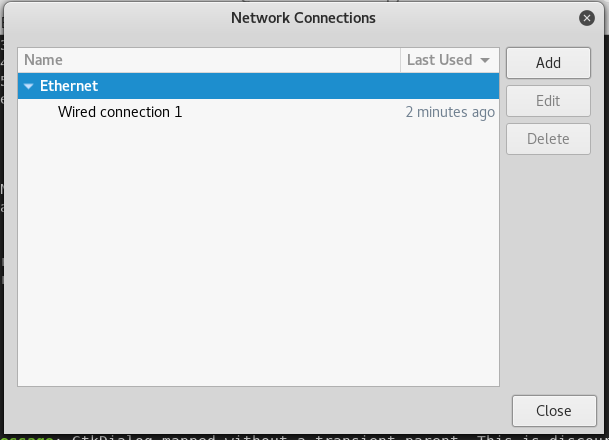
Akan muncul halaman utama untuk memasukkan input pilihan yang akan dilakukan.

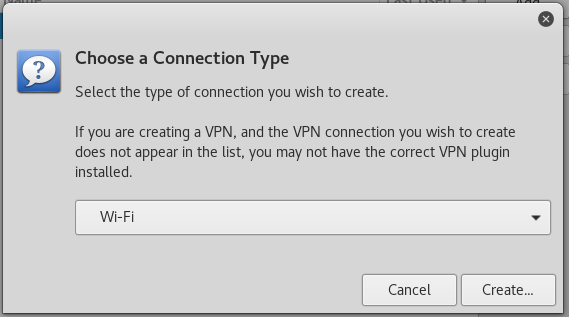


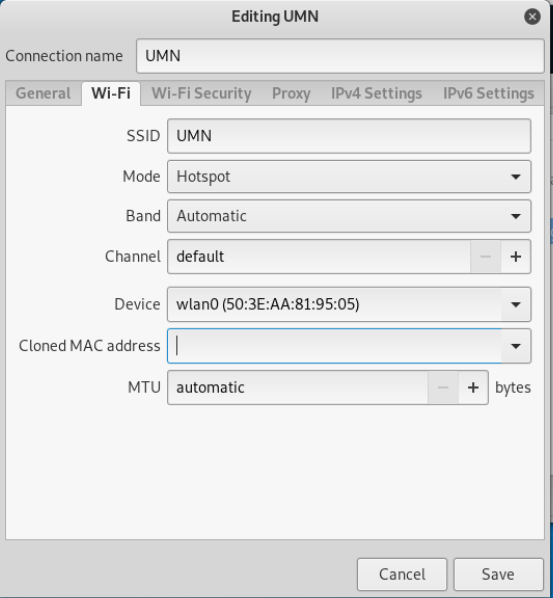
1. Memulai Attack Evil Twin

Input pilihan 1 pada menu utama untuk memulai attack EvilTwin yang akan menghasilkan Access Point yang palsu yang akan digunakan untuk attack.

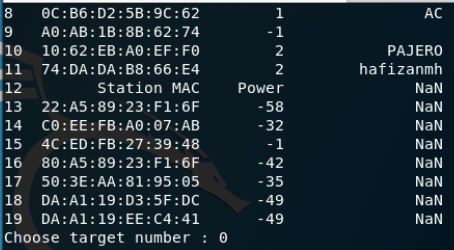
Pertama akan menjalankan Monitor mode untuk melihat AP yang berada disekitar yang akan menjadi target. Lalu akan ditampilkan list dari AP yang berada di sekitar User/Attacker. Kemudian User/Attacker akan diminta untuk membuat AP. Setelah itu User/Attacker akan memilih target yang akan di *Deauth* (Menghilangkan semua koneksi ke AP target).







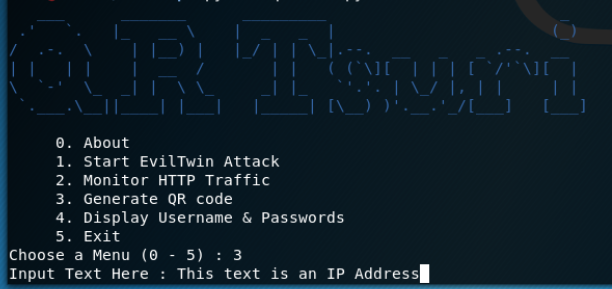
Untuk memilih target AP untuk di duplikat dan di-*deauth*.



1. Generate QR Code

Menggunakan input pilihan 3 untuk memulai pembuatan QRCode.

Dimasukkan string yang akan menjadi rujukan dari QRCode tersebut.

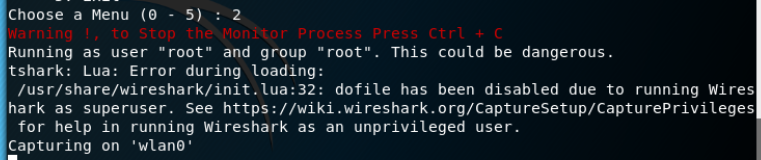


Hasil QRCode yang telah dibuat akan ada di directory yang sama dengan aplikasi python.

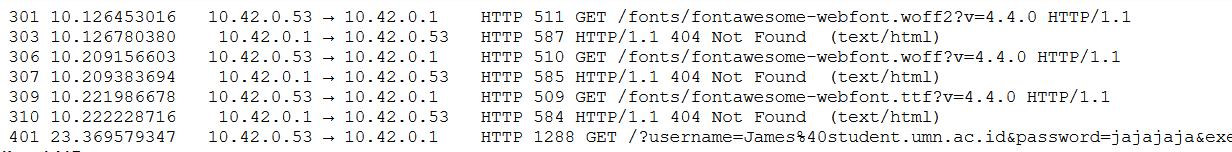


1. Memonitor Traffic HTTP

Gunakan input pilihan 2 agar dapat melakukan monitor ke traffiic HTTP yang melewati AP EvilTwin yang telah dibuat.

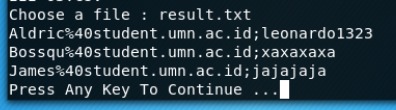


Saat monitor telah selesai maka akan melakukan extract file .txt dengan nama trc\_now untuk menyimpan semua traffic HTTP dari service tshark.



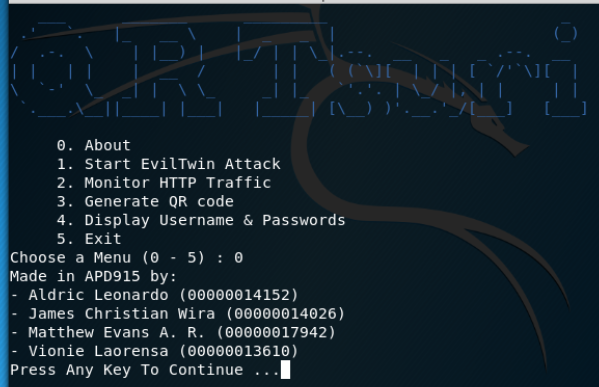
1. Menampilkan Username dan Password

Gunakan input pilihan 4 untuk menampilkan data yang penting dari traffic HTTP seperti username dan password.



1. About

Menu ini akan menampilkan tim pengembang aplikasi.

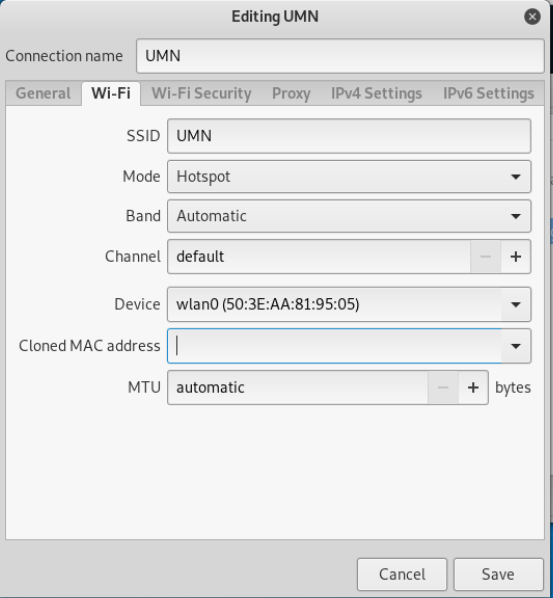


1. Uji Coba

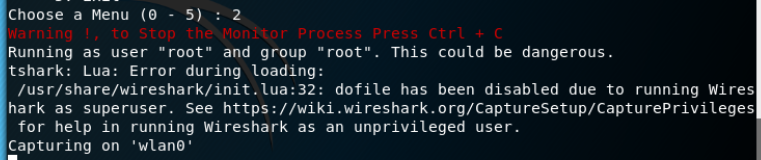
* Attacker pertama-tama akan membuat QRCode dengan menggunakan aplikasi python agar dapat di-scan oleh *victim* untuk dilakukan *redirect* ke halaman phising SSO UMN.



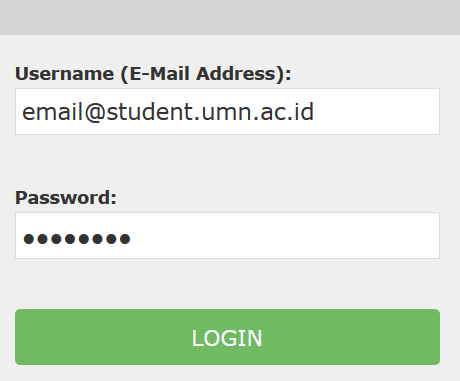
* Dengan menggunakan aplikasi python, attacker kemudian akan membuat Access Point palsu yang memiliki SSID yang sama dengan yang asli, yaitu UMN dan memaksa korban untuk berpindah ke access point yang palsu saat melakukan koneksi ke Access Point yang sebenarnya.



* Attacker kemudian perlu untuk melakukan monitoring traffic HTTP untuk dapat menangkap paket-paket yang lewat Access Point yang palsu.



* Victim kemudian menggunakan QRCode yang telah dihasilkan untuk di-scan dan masuk ke dalam halaman Phising.
* Di halaman Phising akan muncul halaman yang mirip dengan halaman SSO UMN sehingga victim akan melakukan kegiatan login ke halaman SSO UMN palsu tersebut.



* Saat tombol sumbit di-klik oleh victim, maka data input akan dikirimkan ke Access Point yang akan ditangkap oleh tshark.
* Attacker kemudian akan dapat menampilkan username dan password dari traffic yang telah didapatkan dari tshark.

1. Kesimpulan dan Saran

Kesimpulan akhir dari proyek akhir ini adalah wifi bukanlah merupakan teknologi yang sepenuhnya aman, ada banyak faktor yang dapat dieksploitasi dan dimanfaatkan dengan tidak semestinya oleh attacker terhadap teknologi wifi itu sendiri. Oleh karena itu, masyarakat harus lebih berhati – hati dalam melakukan koneksi terhadap wifi terutama yang tersedia di tempat umum. Masyarakat juga tentunya perlu meningkatkan kesadaran dan lebih membuka mata terhadap kemungkinan adanya social engineering yang dapat terjadi, supaya dapat terhindar dari kejahatan lebih lanjut.

Saran pembuatan untuk kedepannya adalah untuk dapat lebih mengoptimalisasi kembali aspek pembuatan access point dan flow proses secara keseluruhan, dapat memperbarui tampilan dari aplikasi agar lebih menarik dan lebih mudah untuk digunakan, melakukan pengembangan lebih lanjut dalam penerapan aplikasi sehingga tidak hanya berhenti pada tahap phishing saja tetapi dapat diteruskan dengan virus, system hacking, dan lain lain. Melakukan perluasan skala dari target atau victim ke level yang lebih tinggi sehingga aplikasi tidak hanya beredar di kalangan mahasiswa UMN saja, tetapi dapat menyebar ke masyarakat luas.