Implementasi *Password Stealing Attack* Terhadap *Saved Password* Pada *Browser* Komputer Menggunakan Digispark Attiny85

Farid Akram[1)]
(1) Badan Siber dan Sandi Negara, farid.akram@bssn.go.id

Abstrak

Password merupakan metode autentikasi yang paling umum digunakan. Namun, terdapat permasalahan dalam penggunaannya, pengguna harus mengingat password miliknya secara terus-menerus. Solusi dari permasalahan tersebut yaitu dengan menggunakan password manager. Saat ini, sudah terdapat password manager yang terintegrasi dengan browser komputer. Namun, sayangnya keamanan pada password manager tersebut tidak sepenuhnya melindungi data pribadi pengguna karena password yang disimpan pada password manager tersebut akan tersimpan dalam suatu file. Hal itu dapat dimanfaatkan pihak jahat untuk mencuri file saved passsword tersebut dengan melakukan password stealing attack menggunakan malicious software (malware). Diantara sekian banyaknya jenis malware, terdapat malware yang dapat dijalankan dengan menggunakan microcontroller universal serial bus (USB). USB interface merupakan bidang yang masih mungkin untuk diserang karena firmware perangkat USB tidak bisa dideteksi oleh perangkat lunak antivirus. Oleh karena itu, pada penelitian ini akan dilakukan implementasi password stealing attack untuk mencuri file saved password menggunakan perangkat microcontroller USB serta memberi pengetahuan terkait dampak yang disebabkan oleh serangan tersebut. Microcontroller USB yang digunakan pada penelitian ini adalah Digispark Attiny85 Tahap implementasi yang dilakukan meliputi penentuan fungsi, pembuat program, melakukan uji coba, dan menganalisis dampak. Setelah dilakukan implementasi dan ujicoba password stealing attack pada Digispark Attiny85, didapatkan hasil bahwa password stealing attack dapat diimplementasikan pada Digispark Attiny85 dengan bukti tercurinya file saved password dan seluruh fungsi pada program berjalan dengan semestinya.

Kata kunci : Digispark, *password, password stealing attack, malicious software*

Abstract

Passwords are the most commonly used authentication method. However, there are problems with their use as users must remember their passwords constantly. The solution to this problem is to use a password manager. Currently, there is a password manager that is integrated with computer browsers. However, the security measures in password managers do not fully protect the user's personal data, and passwords stored in them are vulnerable to exploitation by malicious parties who can steal files stored with these passwords through password stealing attacks using malware. Among the many types of malware, there is malware that can be run using a universal serial bus (USB) microcontroller. The USB interface remains a potential area for attack since the firmware of the USB device cannot be detected by antivirus software. Therefore, this study seeks to implement a password stealing attack that leverages a USB device microcontroller to steal files stored with passwords and provide insight into the impact of the attack. The Digispark Attiny85 USB microcontroller is used in this research. The implementation phase includes function testing, programming, testing, and impact analysis. After implementing and testing the password stealing attack on Digispark Attiny85, the researchers found that the attack could be executed on Digispark Attiny85. The evidence showed that the file stored with the password was stolen, and all the functions of the program ran correctly.

Keywords: Digispark, *password, password stealing attack, malicious software*

1. PENDAHULUAN

Keamanan telah menjadi perhatian utama sejak internet menjadi salah satu faktor yang mendukung kehidupan masyarakat seperti proses jual beli, pendidikan dan proses bisnis lainnya [1]. Aspek mendasar dari keamanan yaitu melindungi data dari akses yang tidak sah. Metode yang paling umum digunakan yaitu dengan menggunakan *password* [1]. *Password* telah menjadi suatu metode yang paling umum digunakan untuk proses autentikasi pengguna [2]. Meskipun saat ini sudah terdapat teknik autentikasi lain seperti *smartcard* dan *biometric*, sistem *password* kemungkinan besar akan tetap digunakan mengingat masalah keamanan, kemudahan pengguna, privasi, dan keandalan dari pendekatan lainnya [3–5]. Namun, terdapat tantangan besar dalam

penggunaan *password* yaitu setelah *password* dibuat, pengguna diharuskan untuk mengingat *password* miliknya tersebut secara terus-menerus. Disisi lain, manusia mempunyai kecenderungan untuk melupakan *password* milik mereka, terlebih lagi jika *password* tersebut tidak sering digunakan oleh mereka.

Oleh karena itu, untuk menjawab tantangan tersebut, beberapa perusahaan-perusahaan teknologi mulai membuat *password manager*. *Password manager* merupakan perangkat lunak yang dapat digunakan pengguna untuk menyimpan informasi berharga dan sensitif mulai dari *username*, *password*, dan informasi kredensial lainnya dengan aman [6]. Sekarang ini, sudah terdapat *password manager* yang sudah terintegrasi dengan *browser* komputer yang memudahkan pengguna dalam penyimpanan *password* dan informasi kredensial lainnya. *Browser* komputer yang sudah terintegrasi dengan *password manager* diantaranya yaitu Google Chrome dan Mozilla Firefox. Google Chrome menempati tempat pertama sebagai *browser* yang paling banyak digunakan pada *platform desktop* sedangkan Mozilla Firefox menempati tempat kedua [7].

Di sisi lain, terdapat permasalahan pada *password manager* yang terintegrasi dengan *browser* komputer yaitu *password*, *username*, dan informasi kredensial lainnya disimpan dalam suatu file dimana file tersebut juga disimpan pada perangkat pengguna [8]. *Password* dan *username* yang tersimpan tersebut biasa disebut dengan *saved password*. Hal tersebut dapat memungkinkan penyerang untuk mengambil *file saved password* dengan melakukan *password stealing attack*. *Password stealing attack* merupakan teknik yang dimanfaatkan oleh penyerang untuk mendapatkan *password* dari korban agar penyerang mendapatkan hak akses dengan menggunakan bantuan *malicious software* (*malware*) sebagai alat bantu penyerangan [9]. *Malware* adalah program komputer yang dirancang untuk membuat efek berbahaya dan tidak diinginkan pada perangkat korban [10]. *Malware* dianggap menjadi salah satu dari sejumlah ancaman yang berbahaya bagi pengguna internet dan diantara banyaknya jenis *malware*, terdapat *malware* yang dapat dijalankan dengan menggunakan *microcontroller Universal Serial Bus* (USB) [11].

Microcontroller merupakan sebuah *mini computer* yang dibuat dalam bentuk *integrated circuit* dan dirancang untuk melakukan tugas tertentu serta menjalankan aplikasi tunggal [12]. Dalam *microcontroller* biasanya sudah terdapat CPU (*Central Processing Unit*), RAM (*Random Access Memory*), ROM (*Read Only Memory*), dan *port* I/O (*Input/Output*) [12]. Salah satu contoh dari *microcontroller* USB adalah Digispark. Digispark merupakan arduino *board* berukuran kecil dan memiliki performa yang baik. Digispark mirip dengan arduino dalam pemrograman, tetapi memiliki harga yang murah, ukuran yang lebih kecil, dan memiliki USB *connector* [13]. USB *interface* merupakan salah satu bidang yang berbahaya dan masih memungkinkan untuk diserang karena *firmware* perangkat USB tidak bisa dideteksi oleh *antivirus* yang memungkinkan perangkat untuk dilakukan aktivitas berbahaya pada mesin *host* [14]. Contohnya, USB *flashdrive* dapat mendaftarkan dirinya sebagai perangkat lain seperti keyboard sehingga memungkinkan USB tersebut dapat menjalankan *script* berbahaya.

Penelitian Hansen Edrick Harianto dan Dennis Gunawan (2019) menggunakan *microcontroller* USB Rubber Ducky untuk mengambil dan menganalisis *password* Wi-Fi dimana tingkat keberhasilan mencapai 92.28% [15]. Selain itu, berdasarkan hasil analisis *Data Breach Investigation Report* (DBIR) yang dilakukan oleh Verizon, dari sekian banyaknya *data breach* yang terjadi, sekitar 20% disebabkan oleh *malware* [16]. Berdasarkan uraian diatas, pada penelitian ini akan dilakukan implementasi *password stealing attack* pada Digispark Attiny85 untuk mengambil *saved password browser* komputer. Penelitian ini bertujuan untuk membuktikan *Password Stealing Attack* dapat diimplementasikan pada Digispark Attiny85. Setelah mengetahui hasil implementassi *Password Stealing Attack* pada *browser* komputer, diharapkan penelitian ini dapat menjadi bahan pertimbangan untuk pengembangan keamanan pada sistem *browser* komputer dan sistem *antivirus* serta pengembangan pada serangan yang menggunakan *microcontroller* USB.

2. LANDASAN TEORI

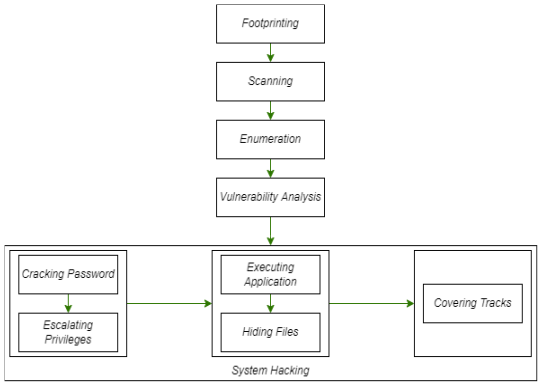
Bagian Landasan Teori mencantumkan teori-teori yang terkait dengan pembahasan , yaitu: *system hacking*, *malicious software* (malware), Digispark, dan *password stealing attack*.

2.1. *System Hacking*

System hacking adalah aktivitas yang dilakukan oleh seorang penyerang untuk mengambil alih sebuah sistem. Sebelum melakukan *system hacking*, penyerang akan melakukan *reconnaissance* dan *scanning* terhadap korban untuk mencari informasi yang akan dibutuhkan pada proses *system hacking*. Terdapat tiga langkah utama dalam melakukan *system hacking* yaitu *gaining access*, *maintaining access*, dan *clearing logs* seperti pada Gambar 2.1 [17]. Pada tahap *gaining access*, penyerang akan melakukan kegiatan yang bertujuan untuk mendapatkan hak akses pada perangkat korban agar penyerang dapat melakukan segala kegiatan *hacking* pada perangkat tersebut. Tahap *gaining access* terbagi menjadi dua bagian yaitu *cracking password* dan *escalating privileges* [17]. *Cracking password* merupakan metode yang bertujuan untuk mendapatkan hak akses ke sistem target dengan menyamar sebagai pengguna yang sah dan akan dilanjutkan dengan *escalating*

privileges untuk menjadikan penyerang sebagai *administrator* pada perangkat korban [17].

Pada tahap *maintaining access*, penyerang akan melakukan kegiatan yang bertujuan untuk mempertahankan hak akses terhadap perangkat korban. Pada umumnya, *maintaining access* dibagi menjadi dua bagian yaitu *executing application* dan *hiding files* [17]. *Executing application* merupakan aktivitas utama penyerang dalam penyerangan terhadap korban. Proses *executing application* dapat dilakukan dengan menggunakan *malware* seperti *trojan*, *spyware*, *backdoor*, atau *keylogger*. Proses selanjutnya yaitu *hiding files*. *Hiding files* merupakan aktivitas yang dilakukan oleh penyerang untuk menyembunyikan *file* atau data apa saja yang sudah diambil penyerang agar tidak terdeteksi oleh sistem keamanan korban. Selanjutnya, tahap terakhir adalah *covering track*. Pada tahap *covering tracks*, penyerang akan menghapus seluruh jejak bahwa telah terjadi sebuah penyerangan pada perangkat korban. Proses *covering tracks* dilakukan dengan menghapus *file* yang berhubungan dengan penyerangan yang telah dilakukan [17].



Gambar 1 *System Hacking*

2.2. Malicious Software

Malicious Software (*Malware*) adalah program komputer yang dirancang untuk membuat efek berbahaya dan tidak diinginkan pada perangkat korban. *Malware* dianggap menjadi salah satu dari banyak ancaman yang berbahaya bagi pengguna internet. Penyerang biasanya merancang *malware* dengan tujuan dan fungsi tertentu yang sesuai dengan kebutuhan penyerang. Saat sudah diaktifkan, *malware* dapat menyebar melalui internet dan menyebabkan kerusakan pada sistem operasi. *Malware* memanfaatkan kerentanan yang ada pada perangkat dan sistem operasi untuk mengeksploitasi data. Terkadang penyerang juga menggunakan teknik *social engineering* dalam menarik minat pengguna untuk menjalankan *malware* pada perangkatnya. Pada umumnya *malware* dikategorikan sebagai berikut [18]:

a. *Viruses*
Malware berjenis virus biasanya disembunyikan pada program lain yang terlihat tidak berbahaya dan biasanya virus melakukan tindakan berbahaya seperti menghancurkan data.

b. *Worms*
Malware ini diberi nama *worms* karena kemampuannya untuk menyusup melalui jaringan. *Worms* mereplikasi dirinya tetapi tidak menyembunyikan dirinya ke dalam program lain seperti yang cenderung dilakukan virus. *Worms* bergerak di sepanjang koneksi jaringan untuk mencari perangkat yang rentan untuk diinfeksi.

c. *Trojans*
Trojans merupakan jenis *malware* yang menyamar sebagai program atau *utility* yang tidak berbahaya untuk mengelabui korban agar korban mau memasangnya. *Malware* ini biasanya membawa fungsi destruktif tersembunyi yang diaktifkan saat *malware* dijalankan.

d. *Spyware*
Fungsi utama *spyware* adalah *malware* yang digunakan untuk memantau aktivitas yang korban lakukan pada perangkatnya, menggunakan ataupun tidak menggunakan internet lalu mengirimkan informasi yang didapatkan ke pihak ketiga tanpa sepengetahuan korban.

e. *Rootkits*
Setelah *malware* terpasang pada sistem, sangat penting untuk *malware* tetap tersembunyi untuk terhindar dari *antivirus*. *Rootkit* memungkinkan penyembunyian ini dengan memodifikasi sistem operasi *host* sehingga *malware* dapat disembunyikan dari korban.

f. *Backdoors*
Backdoors adalah metode untuk melewati prosedur autentikasi normal yang biasanya melalui koneksi ke jaringan seperti internet. Setelah sistem yang ada sudah disusupi satu atau lebih *backdoor*, maka di masa yang akan datang penyerang dapat menyusupi *malware* jenis lainnya tanpa disadari korban.

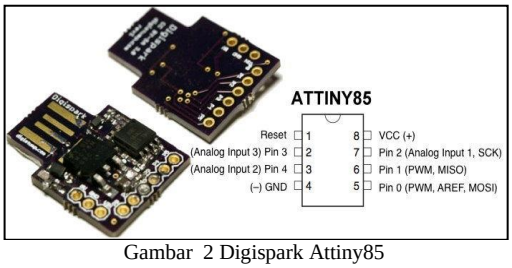
2.3. Digispark

A. Spesifikasi Digispark

Digispark merupakan board development microcontroller berbasis Attiny85 yang mirip dengan Arduino line. Namun, hal yang membedakannya adalah Digispark lebih murah, lebih kecil, dan lebih powerful. Berikut merupakan spesifikasi dari Digispark [19]:

1. Support untuk Arduino IDE 1.0+ (OSX/Win/Linux)
2. Sumber daya melalui USB atau External source
3. On-board 500ma 5V regulator
4. Built-in USB
5. 6 I/O Pin
6. 8k Flash Memory
7. I2C dan SPI

8. Pulse With Modulation (PWM)
9. Analog to Digital Converter (ADC)
10. Power LED dan Test/Status LED



Gambar 2 Digispark Attiny85

B. Library Digispark

Digispark ditenagai dengan Atmel Attiny MCU dan sudah memiliki *library* khusus untuk Digispark itu sendiri [19]. Library tersebut adalah DigiKeyboard.h. Perintah-perintah dasar yang digunakan pada penelitian ini sebagai berikut.

1. DigiKeyboard.sendKeyStroke
   DigiKeyboard.sendKeyStroke digunakan untuk memberikan perintah *input keyboard key* ataupun fungsi *modifier* seperti *enter*, *control*, dan *shift*.
2. DigiKeyboard.delay
   DigiKeyboard.delay digunakan untuk menunda perintah selanjutnya dalam hitungan *milisecond*.
3. DigiKeyboard.print
   DigiKeyboard.print digunakan untuk menuliskan sebuah kata atau kalimat yang diinginkan.

2.4. *Password Stealing Attack*

Password stealing attack merupakan teknik yang dimanfaatkan oleh penyerang untuk mendapatkan *password* dari korban dengan tujuan agar penyerang mendapatkan hak akses korban. *Password stealing attack* dibagi menjadi tiga jenis [9], yaitu:

1. *Password Stealing Program Attack*
   Password stealing program attack merupakan sebuah teknik penyerangan dengan menggunakan *program* yang sudah dibuat oleh penyerang yang akan mengambil *file* atau data yang berisi *username* dan *password* korban. Program *keylogger* dan Trojan *redirectors* merupakan contoh dari *password stealing program attack*.
2. *Phishing Attack*
   Dalam *phishing attack*, penyerang berusaha memperoleh informasi korban dengan berpura-pura sebagai pihak yang bertanggung jawab. Contohnya, seorang penyerang membuat *website* palsu dan kemudian mengirim *email* ke calon korban untuk membujuk korban mengakses *website* palsu tersebut serta memasukan informasi-informasi pribadi kedalamnya.
3. *Shoulder Surfing Attack*
   Shoulder surfing attack merupakan metode pengamatan langsung yang bertujuan untuk mendapatkan informasi tertentu, biasanya dilakukan oleh penyerang di tempat umum yang ramai. Alat yang biasanya digunakan untuk serangan ini adalah kamera mini.

3. METODE PENELITIAN

Metode yang digunakan pada penelitian Tugas Akhir ini adalah metodologi *System Development Life Cycle* (SDLC). SDLC sendiri merupakan sebuah metode dan pedoman yang digunakan untuk membangun suatu sistem informasi. Metode SDLC yang digunakan pada penelitian ini memiliki 4 tahap, yaitu *planning, analysis, design, dan implementation* [21].

3.1. *Planning* (Perencanaan)

Tahapan ini menentukan bagaimana program akan dibuat dan memahami alasan program tersebut dibuat berdasarkan ide penelitian yang dimiliki yaitu implementasi *password stealing attack* pada Digispark Attiny85 untuk mengambil *saved password* pada *browser* komputer. Selain itu, pada tahapan ini juga akan diberikan gambaran umum dari *password stealing attack* dan akan ditampilkan lingkungan implementasi yang berkaitan dengan *password stealing attack* dengan menggunakan Digispark Attiny85. Tahap *planning* dilakukan dengan mengidentifikasi permasalahan yang ada sebagai dasar pembuatan program dan mengumpulkan referensi serta literatur yang berhubungan dengan penelitian.

3.2. *Analysis* (Analisis)

Pada tahap *analysis*, peneliti akan menganalisis kebutuhan fungsional dan non-fungsional, merencanakan program yang akan diimplementasikan, dan desain program yang akan diimplementasikan pada Digispark Attiny85 untuk mengambil *saved password* pada *browser* komputer. Analisis ini dilakukan dengan tujuan untuk menjelaskan fungsi apa saja yang akan dibangun pada program. Hasil dari tahap ini berupa gambaran mengenai implementasi *password stealing attack* terhadap *saved password* pada *browser* komputer pada Digispark Attiny85 yang selanjutnya akan digunakan dalam tahap desain.

3.3. *Design* (Desain)

Pada tahap ini akan dibuat rancangan *password stealing attack* yang akan diimplementasikan pada Digispark Attiny85 untuk mengambil data *saved password* pada *browser* komputer. Pemodelan rancangan akan menggunakan diagram alir untuk dapat memvisualisasikan proses yang akan terjadi pada rancangan *password stealing attack* yang akan diimplementasikan pada Digispark Attiny85. Berdasarkan spesifikasi perangkat yang dijabarkan oleh penelitian Benjamin Cannoles dan Ahmad Ghafarian (2017), program yang dibangun pada

penelitian ini akan memiliki fungsi utama yaitu, membuka dan menutup *command prompt* atau *terminal*, menghidupkan dan mematikan *firewall*, membuka direktori tempat *file saved saved password*, terhubung ke *server* menggunakan FTP, mengirimkan *file saved password* yang sudah didapatkan, menghapus DNS *history* pada *command prompt* dan *terminal*, serta menghapus *command history* pada *command prompt* dan *terminal*.

3.4. *Implementation* (Implementasi)

Pada tahap *implementation*, akan dilakukan pembangunan program *password stealing attack* yang akan diimplementasikan pada Digispark Attiny85 berdasarkan rancangan yang dibuat sebelumnya pada tahap *design*. Setelah program dibangun maka akan dilakukan pengujian pada program tersebut. Pengujian program dilakukan dengan menggunakan pendekatan *white box testing. White box testing* merupakan pengujian dari segi *design* dan *source code*, apakah kedua hal tersebut dapat menghasilkan fungsi, *input*, dan *output* yang sesuai [22]. Teknik yang digunakan pada penelitian ini adalah teknik *path testing*. *Path testing* merupakan teknik komprehensif yang menguji setiap jalur program dan memastikan jalur program tersebut dapat berjalan sesuai dengan semestinya [23].

4. HASIL DAN PEMBAHASAN

4.1. Analisis Kebutuhan Program (*Analysis*).

Proses analisis dilakukan untuk mengetahui kebutuhan dari program yang akan dibangun. Pada bagian analisis ini akan dijelaskan terkait kebutuhan fungsional dan kebutuhan non-fungsional program *password stealing attack* yang dibangun.

a. Analisis Kebutuhan Fungsional Program

Kebutuhan fungsional merupakan kebutuhan yang berkaitan langsung dengan program yang akan dibangun dan merupakan kebutuhan yang keberadaannya sangat penting dalam program yang akan dibangun [21]. Kebutuhan fungsional *password stealing attack* dapat dilihat pada Tabel 1.

Tabel 1 Kebutuhan Fungsional

No	Kebutuhan Fungsional
1.	Perangkat memiliki *switch button* dan fungsi *switch command* untuk mengganti perintah yang akan dijalankan
2.	Program *password stealing attack* memiliki fungsi membuka dan menutup *command prompt* sebagai *administrator* dan *terminal* sebagai *super user*
3.	Program *password stealing attack* memiliki fungsi menonaktifkan dan mengaktifkan kembali *firewall*
4.	Program *password stealing attack* memiliki fungsi berganti ke direktori *file saved saved password*
5.	Program *password stealing attack* memiliki fungsi terhubung ke *server* menggunakan FTP
6.	Program *password stealing attack* memiliki fungsi mengirimkan *file saved password* yang sudah didapatkan
7.	Program *password stealing attack* memiliki fungsi menghapus DNS *history* pada *command prompt* dan *terminal*
8.	Program *password stealing attack* memiliki fungsi menghapus *command history* pada *command prompt* dan *terminal*

b. Analisis Kebutuhan Non-Fungsional Program

Kebutuhan non-fungsional merupakan kebutuhan-kebutuhan yang dibutuhkan untuk membuat program secara optimal. Namun, kebutuhan ini memiliki peran pendukung untuk fungsi program yang akan dijalankan [21]. Oleh karena itu, kebutuhan non-fungsional harus disesuaikan dengan kebutuhan fungsional agar dapat mendukung kinerja program yang dibuat. Kebutuhan non-fungsional *password stealing attack* dapat dilihat pada Tabel 2.

Tabel 1 Kebutuhan Non-Fungsional

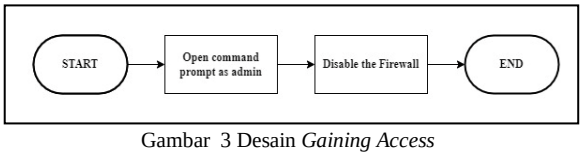
No	Kebutuhan Non-Fungsional
1.	Program *password stealing attack* dibangun menggunakan Arduino IDE.
2.	Program menggunakan Digispark Attiny85 sebagai media implementasi.
3.	Lingkungan uji coba menggunakan program operasi Windows dan Linux ubuntu yang memiliki aplikasi *browser* Google Chrome dan Mozilla Firefox.

4.2. Perancangan Program (*Design*)

Pada tahap ini akan disajikan tahapan-tahapan proses yang akan dilakukan oleh program *password stealing attack* berdasarkan tahap perencanaan dan analisis yang telah dilakukan. Tahapan proses tersebut dibagi menjadi 3 bagian, yaitu:

a. Gaining access

Pada langkah ini, penyerang akan mencoba mendapatkan akses fisik ke komputer korban untuk untuk mencuri *saved password* di browser computer korban. Setelah penyerang mendapatkan akses fisik ke komputer, penyerang harus memilih program untuk sistem operasi mana yang akan dijalankan. Setelah itu, penyerang akan menyuntikkan Digispark Attiny85 ke komputer korban. Seperti yang diilustrasikan pada Gambar 3, metode untuk mendapatkan akses terdiri dari membuka *command prompt* sebagai *administrato*r di Windows dan terminal sebagai *superuser* di Ubuntu serta menonaktifkan firewall untuk melakukan eskalasi hak istimewa.

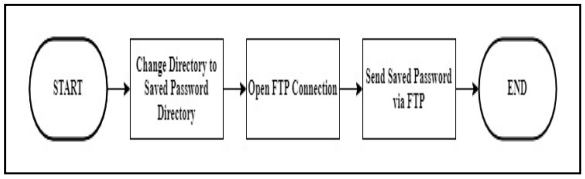


Gambar 3 Desain *Gaining Access*

Comand Prompt dan terminal harus dijalankan sebagai *administrator* atau *superuser* karena perintah tertentu memerlukan hak istimewa administrator untuk dieksekusi. Menonaktifkan firewall diperlukan untuk menghindari perlindungan fitur-fitur tertentu seperti File Transfer Protocol (FTP). Tanpa keamanan dari firewall, segala jenis tindakan terkait internet tidak akan difilter, dan lebih mudah bagi penyerang untuk melanjutkan ke metode berikutnya.

b. Execute program

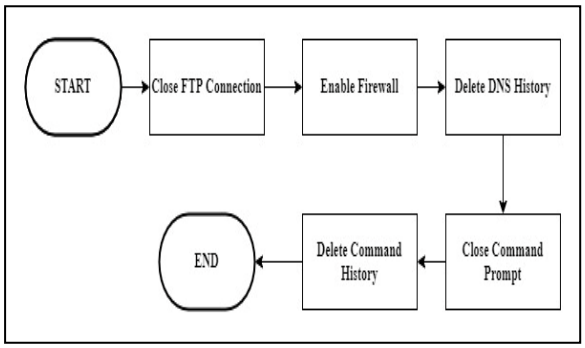
Langkah-langkah berikutnya yaitu program akan membuka direktori *saved password*, membuka koneksi FTP, dan mengirimkan *saved password* ke web server, seperti yang diilustrasikan pada Gambar 4. Web server di sini membantu menyimpan semua *saved password* yang diperoleh. Sebelum mengirimkan *saved password*, penyerang harus mengetahui nama *saved password*. Misalnya, *saved password* pada Mozilla Firefox biasanya disimpan dengan nama key4.db. Setelah penyerang mendapatkan *saved password*, itu akan dikirimkan melalui koneksi FTP.



Gambar 4 Desain *Execute Program*

c. Clearing logs

Pada langkah terakhir, program akan menghapus atau menutupi setiap jejak di komputer korban sehingga terlihat seperti tidak ada yang terjadi pada komputer korban, seperti yang diilustrasikan pada Gambar 5. Tahap ini terdiri dari menutup koneksi FTP, mengaktifkan firewall, menghapus Domain Name System (DNS) *history*, menghapus *comand history*, dan menutup *comand prompt* atau terminal.



Gambar 5 Desain Clearing Logs

4.3. Implementasi (*Implementation*)

Pada tahap ini program yang sudah dibuat berdasarkan *design* yang sudah ditentukan akan diimplementasikan pada Digispark Attiny85 melalui Arduino IDE. Gambar 6 menunjukkan potongan *source code* yang siap untuk diimplementasikan pada Digispark Attiny85 menggunakan Arduino IDE.

Gambar 6 Potongan *Source Code*

4.4. Pengujian Program (*Testing*)

Setelah tahap implementasi, maka akan dilanjutkan ke tahap uji coba program. Pengujian dilakukan untuk memastikan bahwa proses implementasi sudah dilakukan sebagaimana mestinya, program berjalan sebagaimana mestinya, dan menghindari munculnya *error* pada program. Pendekatan yang digunakan pada penelitian ini yaitu pendekatan *white box testing* dan teknik yang digunakan yaitu teknik *path testing*.

Dalam proses uji coba, digunakan lingkungan uji coba yang merupakan lingkungan untuk menguji program yang sudah dibuat. Adapun perangkat-perangkat yang digunakan dalam lingkungan-lingkungan tersebut akan dijelaskan pada Tabel 3.

Tabel 3 Spesifikasi Perangkat Pengujian

No	Informasi	Keterangan
1.	Perangkat	Dell AIO Optiplex 7480
2.	Prosesor	10th *Generation* Intel® *Core*™ i7-10700
3.	RAM	16 GB
4.	Sistem Operasi	*Virtual Machine* Linux Ubuntu 22.04 *Virtual Machine* Linux Ubuntu 20.04 *Virtual Machine* Windows 7 Pro 64-bit *Virtual Machine* Windows 10 Pro 64-bit
5.	*Harddisk*	1000 GB

Pengujian dilakukan sebanyak 4 kali pada masing-masing system operasi dengan total pengujian sebanyak 16 kali pengujian. Setelah dilakukan pengujian, maka didapatkan hasil bahwa program berjalan dengan semestinya dan tidak terdapat error. Hasil pengujian dapat dilihat pada Tabel 4.

Tabel 4 Hasil Pengujian

No	Aktivitas	Frekuensi		
		Uji	Hasil	*Error*
1	*Choose System Operation*	16	16	0
2	*Open CMD as Administrator on Windows*	16	16	0
3	*Disable Firewall on Windows*	16	16	0
4	*Change to Mozilla Firefox Profile Directory on Windows*	16	16	0
5	*Open FTP Connection on Windows*	16	16	0
6	*Login FTP Connection on Windows*	16	16	0
7	*Upload File Saved Password on Windows*	16	16	0
8	*Close FTP Connection on Windows*	16	16	0
9	*Change to Google Chrome Profile Directory on Windows*	16	16	0
10	*Enable Firewall on Windows*	16	16	0
11	*Delete DNS History on Window*	16	16	0
12	*Delete Command History on Windows*	16	16	0
13	*Close CMD on Windows*	16	16	0
14	*Open Terminal as Super User on Linux Ubuntu*	16	16	0
15	*Disable Firewall on Linux Ubuntu*	16	16	0
16	*Change to Mozilla Firefox Profile Directory on Linux Ubuntu*	16	16	0
17	*Open FTP Connection on Linux Ubuntu*	16	16	0
18	*Login FTP Connection on Linux Ubuntu*	16	16	0
19	*Upload File Saved Password on Linux Ubuntu*	16	16	0
20	*Close FTP Connection on Linux Ubuntu*	16	16	0
21	*Change to Google Chrome Profile Directory on Linux Ubuntu*	16	16	0
22	*Enable Firewall on Linux Ubuntu*	16	16	0
23	*Delete DNS History on Linux Ubuntu*	16	16	0
24	*Delete Command History on Linux Ubuntu*	16	16	0
25	*Close CMD on Linux Ubuntu*	16	16	0

5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat diperoleh kesimpulan sebagai berikut:

1. Tahap implementasi *password stealing attack* pada Digispark Attiny85, yaitu:
   a) Penentuan fungsi-fungsi apa saja yang dibutuhkan program.
   b) Pembuatan program *password stealing attack.*
   c) Melakukan *upload* program *password stealing attack* ke Digispark Attiny85.
   d) Melakukan uji coba terhadap fungsi-fungsi yang ada, antara lain:
      1. Berganti jenis program sesuai dengan sistem operasi korban.
      2. Membuka dan menutup *command prompt* atau *terminal.*
      3. Menonaktifkan dan mengaktifkan kembali *firewall*.
      4. Berpindah ke direktori Mozilla Firefox dan Google Chrome.
      5. Membuka koneksi FTP dan mengirim *file saved password* yang didapatkan.
      6. Menghapus DNS *history* dan *command history.*
2. Program yang telah dibuat dapat diimplementasikan pada Linux Ubuntu dengan versi 20.04 dan 22.04 serta Windows dengan versi 7 dan 10.
3. Berdasarkan dari hasil implementasi dan uji coba program *password stealing attack* pada Digispark Attiny85 yang telah dilakukan, peneliti menyimpulkan bahwa implementasi *password stealing attack* pada Digispark Attiny85 untuk mencuri *saved password browser* berhasil dilakukan. Hal tersebut didasari dari hasil proses uji coba, dimana setiap fungsi yang dibutuhkan sudah berjalan dengan semestinya dan tujuan akhir dari program *password stealing attack* yaitu mencuri *file saved password* sudah terpenuhi.

REFERENSI

[1] G. Hu, "On password strength: A survey and analysis," *Stud. Comput. Intell.*, vol. 721, no. January, pp. 165–186, 2018, doi: 10.1007/978-3-319-62048-0_12.

[2] T. Khodadadi, A. K. M. M. Islam, S. Baharun, and S. Komaki, "Evaluation of recognition-based graphical password schemes in terms of usability and security attributes," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 6, pp. 2939–2948, 2016, doi: 10.11591/ijece.v6i6.11227.

[3] A. P. Sabzevar and A. Stavrou, "Universal multi-factor authentication using graphical passwords," *SITIS 2008 - Proc. 4th Int. Conf. Signal Image Technol. Internet Based Syst.*, pp. 625–632, 2008, doi: 10.1109/SITIS.2008.92.

[4] P. Shi, B. Zhu, and A. Youssef, "A PIN entry scheme resistant to recording-based shoulder-surfing," *Proc. - 2009 3rd Int. Conf. Emerg. Secur. Information, Syst. Technol. Secur. 2009*, pp. 237–241, 2009, doi: 10.1109/SECURWARE.2009.43.

[5] L. Catuogno and C. Galdi, "Graphical Passwords," vol. 4, pp. 111–128, 2012, doi: 10.4018/978-1-4666-0978-5.ch006.

[6] P. Gasti and K. B. Rasmussen, "On the security of password manager database formats," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7459 LNCS, pp. 770–787, 2012, doi: 10.1007/978-3-642-33167-1_44.

[7] S. GlobalStats, "Desktop Browser Market Share Indonesia," *October*, 2021. https://gs.statcounter.com/browser-market-share/desktop/indonesia (accessed Nov. 06, 2021).

[8] J. B. Billa, A. Nawar, M. M. H. Shakil, and A. K. Das, "PassMan: A New Approach of Password Generation and Management without Storing," *2019 7th Int. Conf. Smart Comput. Commun. ICSCC 2019*, pp. 1–5, 2019, doi: 10.1109/ICSCC.2019.8843591.

[9] V. S and K. Palanivel, "A Survey on Password Stealing Attacks and Its Protecting Mechanism," *Int. J. Eng. Trends Technol.*, vol. 19, no. 4, pp. 223–226, 2015, doi: 10.14445/22315381/ijett-v19p239.

[10] M. Faizal, A. Razak, N. Badrul, R. Salleh, and A. Firdaus, "The rise of ' malware ': Bibliometric analysis of malware study," *J. Netw. Comput. Appl.*, vol. 75, pp. 58–76, 2016, doi: 10.1016/j.jnca.2016.08.022.

[11] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *J. Comput. Syst. Sci.*, vol. 80, no. 5, pp. 973–993, 2014, doi: 10.1016/j.jcss.2014.02.005.

[12] A. Hussain, M. Hammad, K. Hafeez, and T. Zainab, "Programming a Microcontroller," *Int. J. Comput. Appl.*, vol. 155, no. 5, pp. 21–26, 2016, doi: 10.5120/ijca2016912310.

[13] Electronic-web.com, "INTRODUCTION TO DIGISPARK – A SMALLER, CHEAPER AND POWERFUL ARDUINO BOARD," 2018. https://www.electronics-lab.com/introduction-digispark-smaller-cheaper-powerful-arduino-board/ (accessed Nov. 06, 2021).

[14] B. Cannoles and A. Ghafarian, "Hacking experiment using USB rubber ducky scripting," *IMCIC 2017 - 8th Int. Multi-Conference Complexity, Informatics Cybern. Proc.*, vol. 2017-March, no. 2, pp. 73–78, 2017.

[15] H. E. Harianto and D. Gunawan, "Wi-Fi password stealing program using USB rubber ducky," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 2, pp. 745–752, 2019, doi: 10.12928/TELKOMNIKA.V17I2.11775.

[16] Verizon Business, "Data Breach Investigations Report (DBIR) 2021," *Trends*, pp. 1–62, 2021, [Online]. Available: rp_data-breach-investigations-report-2013_en_xg.pdf.

[17] I. P. Specialist, "CEH V10 EC - COUNCIL CERTIFIED ETHICAL HACKER," 2018.

[18] A. O. Eze and C. C. E, "Malware Analysis and Mitigation in Information Preservation," *J. Comput. Eng.*, vol. 20, no. 4, pp. 53–62, 2018, doi: 10.9790/0661-2004015362.

[19] Digistump, "Digispark USB Development Board," 2015. http://digistump.com/products/1 (accessed Nov. 07, 2021).

[20] Instructables.com, "Digispark DIY: the Smallest USB Arduino : 9 Steps (with Pictures)." https://www.instructables.com/Digispark-DIY-The-smallest-USB-Arduino/ (accessed Nov. 18, 2021).

[21] Dennis, Wixom, and Roth, *System Analysis & Design 5th Edition*. United States of America: John Wiley & Sons, Inc., 2012.

[22] Rosa A. S and M. Shalahuddin, "Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek," *Informatika*, vol. 4, 2016.

[23] V. P. Katiyar and S. Patel, "White-Box Testing Technique for Finding Defects," vol. 8, no. 7, 2019, [Online]. Available: http://worldwidejournals.co.in/index.php/gjra/article/view/4883.