



3.6.4 Pengembangan Program *Reporting* Otomatis 54

3.7 PENGUJIAN PENELITIAN 59

3.7.1 Pengujian Fungsionalitas Program Otomasi..... 60

3.7.1.1 Pengujian program eksploitasi 60

3.7.1.2 Pengujian program *reporting* 60

3.7.2 Perbandingan Waktu Antara Metode Otomatis dengan Manual 61

3.7.2.1 Pengambilan Waktu dengan Metode Otomatis 61

3.7.2.2 Pengambilan Waktu dengan Metode Manual..... 61

3.7.2.3 Langkah *Penetration Testing* secara manual..... 62

BAB IV HASIL DAN PEMBAHASAN..... 63

4.1 HASIL PENGUJIAN FUNGSIONALITAS PROGRAM..... 63

4.1.1 Analisis Hasil Pengujian Program Eksploitasi 64

4.1.2 Analisis Hasil Pengujian Program *Reporting*..... 65

4.2 PERBANDINGAN ANTARA METODE OTOMATIS DENGAN MANUAL 68

BAB V PENUTUP 73

5.1 KESIMPULAN..... 73

5.2 SARAN..... 73

DAFTAR PUSTAKA 74

LAMPIRAN 76

Gambar 2. 1 Detail CVE-2023-30799 pada nvd.nist.gov	16
Gambar 3. 1 Gambar Alur Penelitian	20
Gambar 3. 2 Topologi Skenario Penyerangan	23
Gambar 3. 3 Flowchart run.py	24
Gambar 3. 4 Flowchart generate_report.py	26
Gambar 3. 5 <i>Import library</i> pada exploit.py	28
Gambar 3. 6 <i>Source Code</i> fungsi p32 dan class ROPgadget	29
Gambar 3. 7 <i>Source Code</i> ROPContext exploit.py	30
Gambar 3. 8 <i>Source Code</i> exploit.py	31
Gambar 3. 9 Fungsi yang ada pada <i>Class</i> ROPContext	32
Gambar 3. 10 <i>Source Code</i> fungsi load_from_file bagian pertama	32
Gambar 3. 11 <i>Source Code</i> fungsi load_from_file bagian kedua	33
Gambar 3. 12 <i>Source Code</i> fungsi print_info	34
Gambar 3. 13 <i>Source Code</i> fungsi save_to_db	34
Gambar 3. 14 <i>Source Code</i> fungsi load_from_db	35
Gambar 3. 15 <i>Source Code</i> fungsi pop3	35
Gambar 3. 16 <i>Source Code</i> write_32	36
Gambar 3. 17 <i>Source Code</i> write_string	37
Gambar 3. 18 <i>Source Code</i> set_ecx	37
Gambar 3. 19 <i>Source Code</i> set_ebx_uclibc_offset	38
Gambar 3. 20 <i>Source Code</i> set_edx_ebx	39
Gambar 3. 21 <i>Source Code</i> fungsi build_v1 bagian pertama	40
Gambar 3. 22 <i>Source Code</i> fungsi build_v1 bagian kedua	41
Gambar 3. 23 <i>Source Code</i> fungsi fingerprint_version	42
Gambar 3. 24 <i>Source Code</i> fungsi upload_bins	42
Gambar 3. 25 Fungsi send_ropchain pada exploit.py	44
Gambar 3. 26 Fungsi super_admin pada exploit.py	45
Gambar 3. 27 <i>Source Code</i> fungsi needs_json	46
Gambar 3. 28 Fungsi main pada exploit.py bagian pertama	47
Gambar 3. 29 Fungsi main pada exploit.py	48
Gambar 3. 30 <i>Parser</i> pada exploit.py	48
Gambar 3. 31 <i>Folder</i> db	49
Gambar 3. 32 <i>Import</i> pada program run.py	51
Gambar 3. 33 <i>Source Code</i> fungsi is_valid_ip	51
Gambar 3. 34 <i>Source Code</i> fungsi get_non_empty_input	52
Gambar 3. 35 <i>Source Code</i> fungsi sanitize_input	52
Gambar 3. 36 <i>Source Code</i> fungsi run_exploit	53
Gambar 3. 37 <i>Source Code</i> fungsi main pada run.py	53
Gambar 3. 38 <i>Import</i> pada generate_report.py	54
Gambar 3. 39 <i>Source Code</i> fungsi sanitize_text	55
Gambar 3. 40 <i>Source Code</i> fungsi add_header	55
Gambar 3. 41 <i>Source Code</i> fungsi create_cover_page	56
Gambar 3. 42 <i>Source Code</i> fungsi generate_report bagian pertama	57
Gambar 3. 43 <i>Source Code</i> fungsi generate_report bagian kedua	58
Gambar 3. 44 <i>Source Code</i> fungsi generate_report bagian ketiga	59
Gambar 3. 45 Pengujian program eksploitasi	60
Gambar 3. 46 Pengujian program reporting	60
Gambar 3. 47 Pengambilan Waktu dengan Metode Otomatis	61
Gambar 3. 48 Langkah <i>Penetration Testing</i> dengan Metode Manual	62
Gambar 4. 1 Hasil Pengujian Program Eksploitasi	64
Gambar 4. 2 Hasil Pengujian Program Reporting	65
Gambar 4. 3 Hasil Pengujian Program Reporting	66
Gambar 4. 4 Hasil Pengujian Program Reporting	67



Pengembangan program otomatisasi uji penetrasi CVE-2023-30799 pada perangkat jaringan mikrotik berbasis python
JOHNAN CHRISTIAN, Dr. Ronald Adrian S.T., M.Eng.
Universitas Gadjah Mada, 2024 | Diunduh dari <http://etd.repository.ugm.ac.id/>

Gambar 4.5 Perbandingan Waktu Metode Otomatis dengan Manual 68
Gambar 4.6 Perbandingan durasi yang digunakan tiap kegiatan *penetration testing* 70
Gambar 4.7 Menjalankan kegiatan *reconnaissance* secara manual..... 71



DAFTAR TABEL

Tabel 2. 1 Ringkasan hasil skenario pada jurnal penelitian 6
Tabel 2. 2 Tabel Ringkasan Jurnal Penelitian 8
Tabel 3. 1 Tabel Spesifikasi Laptop 18
Tabel 3. 2 Tabel Spesifikasi VirtualBox 18
Tabel 3. 3 Tabel Spesifikasi Kali Linux 19
Tabel 3. 4 Tabel Spesifikasi Python 19
Tabel 3. 5 Tabel Spesifikasi router 20
Tabel 4. 1 Tabel Uji Fungsionalitas 63
Tabel 4. 2 Perbandingan aktivitas saat melakukan kegiatan *penetration testing* 68
Tabel 4. 3 Perbandingan taktik yang dapat dilakukan metode manual dengan otomatis 71
Tabel 4. 4 Perbandingan hasil yang didapat dari metode otomatis dengan manual .. 72

Internet di Indonesia menggunakan perangkat dari vendor yang bernama MikroTik.

MikroTik adalah salah satu vendor perangkat yang umum digunakan untuk kebutuhan infrastruktur internet di Indonesia. Indonesia sendiri memiliki 230.000 perangkat MikroTik yang terpapar ke internet. Karena hal itu, Indonesia menduduki peringkat ketiga di dunia dalam hal penggunaan MikroTik pada infrastrukturnya ('*Observations on cyber threat activity and vulnerabilities in Indonesia, Malaysia, Philippines and Thailand: The Shadowserver Foundation*', 2023). MikroTik juga memiliki kerentanan atau *vulnerability* pada perangkatnya. Salah satunya CVE-2023-30799 yang bisa mengakses router dengan meningkatkan akses admin menjadi *super-admin* melalui antarmuka *winbox* atau layanan web HTTP. Kerentanan ini memungkinkan penyerang mengesekusi kode atau perintah melalui *backdoor*. Disebabkan oleh banyaknya perangkat MikroTik di Indonesia, diperlukan sebuah uji pengamanan efisien untuk CVE-2023-30799. Penelitian ini bertujuan untuk mengembangkan program eksploitasi dan *reporting* secara otomatis dengan menggunakan Python. Harapan penelitian ini agar dapat memperoleh solusi menghemat waktu dalam melakukan *security assessment* untuk CVE-2023-30799. Pengujian penelitian dibagi menjadi dua yaitu uji fungsionalitas dan uji perbandingan waktu antara metode manual dengan otomatis. Hasil penelitian menunjukkan bahwa program mampu dieksekusi tanpa kesalahan dan mampu menjalankan fungsi eksploitasi dan *reporting* dengan benar. Dengan diterapkannya program otomatisasi yang dibuat pada penelitian ini, proses *penetration testing* pada CVE-2023-30799 pada perangkat *router* MikroTik dapat dilakukan 79,1% lebih cepat daripada metode manual

Kata kunci: mikrotik, otomatisasi, *penetration testing*, keamanan siber

The Internet infrastructure in Indonesia relies extensively on devices provided by a vendor known as MikroTik. MikroTik stands as one of the prevalent equipment vendors catering to the internet infrastructure requirements within the Indonesian context. The nation boasts a staggering count of 230,000 MikroTik devices exposed to the internet. Consequently, Indonesia occupies the third position globally in terms of MikroTik device adoption for its infrastructure ('Observations on cyber threat activity and vulnerabilities in Indonesia, Malaysia, Philippines and Thailand: The Shadowserver Foundation', 2023). However, MikroTik devices are not immune to vulnerabilities. Notably, one such vulnerability identified is CVE-2023-30799, enabling unauthorized access to routers by elevating admin privileges to super-admin via the Winbox interface or HTTP web service. This vulnerability potentially allows attackers to execute arbitrary code or commands through a backdoor. Given the pervasive presence of MikroTik devices in Indonesia, there arises a necessity for an efficient security assessment concerning CVE-2023-30799. This study aims to develop an automated exploitation and reporting program utilizing Python. The research endeavors to streamline the time-consuming security assessment process associated with CVE-2023-30799, thus envisaging time-saving solutions. The research methodology involves two primary phases: functional testing and a comparative analysis of time efficiency between manual and automated methods. The findings indicate error-free execution of the program, proficiently carrying out exploitation functions and reporting accurately. By implementing the automation program devised in this study, the penetration testing process for CVE-2023-30799 on MikroTik router devices can be expedited by 79,1% compared to manual methods.

Keywords: mikrotik, automation, penetration testing, cyber security

BAB I

PENDAHULUAN

1.1 Latar belakang

Keamanan Informasi sangat penting pada era sekarang ini. Teknologi berkembang pesat dengan banyak inovasi dimana – mana. Perangkat Teknologi memiliki manfaat yang berfungsi untuk membantu aktivitas manusia. Semakin bermanfaat teknologi, semakin melekat teknologi itu dengan seseorang. Sebuah alat teknologi memiliki memori yang menyimpan sesuatu untuk mendukung fungsinya. Memori yang diisi dengan data pribadi seseorang akan lebih membantu seseorang tersebut dalam hal pribadinya. Tentunya teknologi bisa disalahgunakan oleh orang yang tidak bertanggung jawab. Oleh karena itu, diperlukan sebuah keamanan informasi yang dapat menjaga kerahasiaan dan keamanan dari data yang dimiliki oleh alat tersebut.

Internet di Indonesia menggunakan perangkat dari vendor yang bernama MikroTik. MikroTik adalah salah satu vendor perangkat yang umum digunakan untuk kebutuhan infrastruktur internet di Indonesia. Indonesia sendiri memiliki 230.000 perangkat MikroTik yang terpapar ke internet. Karena hal itu, Indonesia menduduki peringkat ketiga di dunia dalam hal penggunaan MikroTik di infrastrukturnya (*'Observations on cyber threat activity and vulnerabilities in Indonesia, Malaysia, Philippines and Thailand: The Shadowserver Foundation', 2023*). MikroTik juga memiliki kerentanan atau *vulnerability* pada perangkatnya. Salah satunya CVE-2023-30799 yang bisa mengakses *router* dengan meningkatkan akses admin menjadi *super-admin* melalui antarmuka *winbox* atau layanan web HTTP. Kerentanan ini memungkinkan penyerang mengeksekusi kode atau perintah melalui *backdoor*

Untuk mengamankan sebuah informasi, salah satu metode yang dapat dilakukan adalah evaluasi keamanan perangkat atau komputer dengan cara menstimulasikan serangan siber pada perangkat atau komputer yang disebut *penetration testing* (uji penetrasi). Seseorang yang bekerja pada bidang *penetration testing* disebut dengan *pentester*. *Penetration testing* yang sering disingkat dengan *pentest* adalah metode yang sering dilakukan oleh ahli keamanan siber ketika ingin memeriksa keamanan sebuah perangkat secara langsung.

Saat melakukan *penetration testing* terdapat referensi pada internet yang berisi tentang kerentanan (*vulnerability*) yang ada. Referensi ini disebut dengan CVE (*Common Vulnerabilities and Exposures*). CVE dibuat oleh perusahaan non-profit bernama MITRE dengan pembiayaan dari *US Division of Homeland Security* atau Divisi Keamanan Dalam Negeri Amerika Serikat. CVE menganalisa skor kerentanan dengan menggunakan *Common Vulnerability Scoring System* (CVSS). CVE digunakan sebagai glosarium keamanan informasi untuk digunakan sebagai panduan mengamankan perangkat teknologi informasi

Penetration testing memiliki berbagai metode. Salah satu metode *penetration testing* adalah *Binary Exploitation*. *Binary Exploitation* telah lama menjadi masalah keamanan siber utama dan terus menjadi ancaman keamanan yang serius bagi setiap individu dan organisasi di seluruh dunia. Meskipun sejumlah besar upaya didedikasikan untuk itu, mengatasi dan mencegah *Binary Exploitation* tetap menjadi masalah yang masih belum terpecahkan (Liu dkk., 2022).

Penelitian yang berjudul Pengembangan Program Otomatisasi Uji Penetrasi CVE-2018-14847 Pada Perangkat Jaringan MikroTik Berbasis Python dan Shodan bertujuan untuk melakukan pengembangan *penetration testing* dengan otomatis (*autopentest*) menggunakan *search engine* SCADA Shodan dengan CVE-2018-14847. Target dari *penetration testing* adalah router MikroTik dengan versi 6.42 kebawah. Hasil dari penelitian adalah sebuah laporan otomatis yang berisi *username* dan *password* yang ada pada perangkat router MikroTik. Otomatisasi yang dilakukan adalah melakukan *pentest* sekaligus melakukan *auto-reporting* hanya dengan menjalankan satu berkas kode. Penelitian juga menggunakan target *router* fisik yang memiliki IP publik yang didapatkan dengan menggunakan *search engine* SCADA Shodan. Kerentanan atau CVE yang digunakan pada penelitian ini sudah usang dan banyak perangkat MikroTik yang sudah tidak terpengaruh dengan kerentanan pada CVE-2018-14847 (Sanjaya. Y., 2023)

Dari permasalahan pada penelitian diatas, maka penulis berinisiatif untuk melakukan penelitian yang berjudul Implementasi Otomatisasi *Pentest* pada *Binary Exploitation* berbasis kali linux dengan CVE-2023-30799. Penelitian ini diharapkan dapat memperbaharui sebuah metode *pentest* yang kompatibel dengan versi 6.42 sampai 6.49 yang lebih baru dan lebih relevan pada masa sekarang ini. Program pada

penelitian diharapkan dapat mempermudah dan mempersingkat waktu dalam melakukan *penetration testing*. Program akan melakukan eksploitasi otomatis dan melaporkan bukti konsep (*proof of concept*) yang telah dilakukan sehingga respon yang diperlukan dapat segera dilakukan untuk mengamankan perangkat jaringan.

1.2 Rumusan Masalah

Berdasarkan penjelasan pada latar belakang, maka perumusan masalah yang diangkat adalah bagaimana cara mempersingkat waktu *penetration testing* dengan menggunakan program yang didalamnya termasuk proses eksploitasi dan pelaporan kerentanan CVE-2023-30799 pada perangkat jaringan MikroTik menggunakan metode otomatisasi berbasis Python.

1.3 Batasan Masalah

Penelitian proyek akhir ini memiliki beberapa batasan masalah, adapun batasan masalah tersebut adalah sebagai berikut:

1. Program yang dikembangkan berfokus pada proses eksploitasi, dan pelaporan secara otomatis untuk kerentanan CVE-2023-30799 pada perangkat jaringan MikroTik.
2. Perangkat jaringan MikroTik yang akan di uji merupakan perangkat jaringan MikroTik virtual, menggunakan arsitektur x86 dan diimplementasikan secara lokal, hal ini dikarenakan hasil eksplotasi adalah penciptaan *backdoor* yang hanya dapat dihapus atau dinonaktifkan jika perangkat router di-*restart* atau *reset*. Hal tersebut membahayakan bila menggunakan *physical router* karena menambah sebuah *backdoor access* pada sebuah perangkat.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk

1. Pengembangan sebuah program otomatisasi yang mampu melakukan eksploitasi, dan pelaporan secara otomatis untuk proses uji penetrasi kerentanan CVE-2023-30799 pada perangkat jaringan MikroTik
2. Penghematan waktu saat proses *penetration testing*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Pemberian solusi dan respon yang tepat untuk kerentanan CVE-2023-30799 pada perangkat MikroTik
2. Penghematan waktu pekerjaan *pentester* atau *network administrator* saat melakukan *penetration testing* CVE-2023-30799 pada perangkat *router* MikroTik
3. Pembuktian pada metode *penetration testing* secara otomatis yang dapat dilakukan pada *router* MikroTik

1.6 Sistematika Penelitian

Bagian sistematika penulisan memuat garis besar dari laporan proyek akhir yang dikerjakan dengan sistematika penelitian dibagi menjadi lima bab. Bagian tersebut adalah sebagai berikut:

BAB I PENDAHULUAN, berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan yang digunakan penulis pada laporan ini.

BAB II TINJAUAN PUSTAKA DAN DASAR TEORI, Dalam tinjauan pustaka dan dasar teori, penulis membahas penelitian mereka terhadap artikel dan jurnal tentang penelitian sebelumnya, yang digunakan sebagai teori pendukung dan sebagai dasar untuk penelitian mereka saat ini, proyek akhir.

BAB III METODE PENELITIAN, Metode penelitian mencakup bahan, peralatan, langkah-langkah penelitian, rancangan sistem, dan analisis data penelitian proyek akhir penulis.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN, Dalam bab hasil penelitian dan pembahasan, penulis membahas temuan, analisis, dan argumen tentang temuan penelitian proyek akhir.

BAB V PENUTUP, Bab penutup berisi kesimpulan terkait penelitian proyek akhir ini serta saran dari penulis untuk pengembangan penelitian selanjutnya

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian yang berjudul *Automated versus Manual Approach of Web Application Penetration Testing*. Penelitian tersebut membandingkan metode *penetration testing* secara manual atau otomatis. Penelitian dilakukan dengan skenario eksploitasi pada aplikasi web. Skenario yang dipakai pada penelitian adalah *clickjacking* yang digunakan untuk kerentanan *sameorigin*. Skenario *file upload* untuk kerentanan XSS. Skenario *sensitive files* untuk kerentanan *public domain*. Skenario terakhir adalah *business logic* untuk kerentanan *logic failure*. Hasil yang didapatkan pada penelitian tersebut bahwa dalam *penetration testing* secara manual lebih efektif dalam mendapatkan hasil eksploitasi yang lebih banyak tetapi dengan metode otomatis dapat melakukan sebuah eksploitasi yang mudah dengan cepat. Hasilnya adalah metode *penetration testing* secara otomatis atau manual memiliki kelebihan dan kekurangannya masing – masing pada skenario yang berbeda – beda (Singh dkk., 2020). Ringkasan dari hasil penelitian *Automated versus Manual Approach of Web Application Penetration Testing* dapat dilihat pada tabel 2.1.

Tabel 2. 1 Ringkasan hasil skenario pada jurnal penelitian

Skenario	Kerentanan	Manual	Otomatis
<i>Clickjacking</i>	<i>Sameorigin</i>	✓	✗
<i>File Upload</i>	XSS	✓	✗
<i>Sensitive Files</i>	<i>Public Domain</i>	✗	✓
<i>Business Logic</i>	<i>Logic Failure</i>	✓	✗

Penelitian yang berjudul Pengembangan Program Otomatisasi Uji Penetrasi CVE-2018-14847 Pada Perangkat Jaringan MikroTik Berbasis Python dan Shodan. Penelitian ini bertujuan untuk mempersingkat waktu dalam melakukan uji penetrasi pada perangkat MikroTik. Eksploitasi yang dihasilkan oleh penelitian ini adalah *username* dan *password* dari sebuah perangkat *router* MikroTik. Penelitian ini membuktikan bahwa dengan menggunakan otomatisasi pada uji penetrasi dapat mempermudah dan mempercepat pekerjaan seorang *pentester*. *Penetration testing* pada penelitian menggunakan tahap, *scanning* sebagai tahap *reconnaissance*, *exploit*, dan yang terakhir adalah tahap *reporting*. Semua tahap pada penelitian dilakukan secara otomatis. Tahap *reporting* yang dilakukan secara otomatis juga di-generate dengan ekstensi PDF (Antono, Y. 2023).

Penelitian yang berjudul *Binary Exploitation in Industrial Control Systems: Past, Present and Future*. Penelitian berfokus pada salah satu metode *penetration testing* yaitu *binary exploitation*. Skenario dilakukan dengan melibatkan sistem kontrol industri yang menjadi target eksploitasi. Penelitian membahas pertahanan berbasis deteksi dan bagaimana mitigasi dari metode *binary exploitation*. Hasil dari penelitian adalah deteksi dari pertahanan deteksi pada eksploitasi *binary exploitation*. Penelitian menyatakan bahwa *binary exploitation* adalah masalah yang serius dan sulit ditangani. *Binary exploitation* sulit ditangani karena serangannya langsung pada memori sehingga sulit serangan dikenali (Liu dkk., 2022).

Gambar 2. 1 Detail CVE-2023-30799 pada nvd.nist.gov	16
Gambar 3. 1 Gambar Alur Penelitian	20
Gambar 3. 2 Topologi Skenario Penyerangan	23
Gambar 3. 3 Flowchart run.py	24
Gambar 3. 4 Flowchart generate_report.py	26
Gambar 3. 5 <i>Import library</i> pada exploit.py	28
Gambar 3. 6 <i>Source Code</i> fungsi p32 dan class ROPgadget	29
Gambar 3. 7 <i>Source Code</i> ROPContext exploit.py	30
Gambar 3. 8 <i>Source Code</i> exploit.py	31
Gambar 3. 9 Fungsi yang ada pada <i>Class</i> ROPContext	32
Gambar 3. 10 <i>Source Code</i> fungsi load_from_file bagian pertama	32
Gambar 3. 11 <i>Source Code</i> fungsi load_from_file bagian kedua	33
Gambar 3. 12 <i>Source Code</i> fungsi print_info	34
Gambar 3. 13 <i>Source Code</i> fungsi save_to_db	34
Gambar 3. 14 <i>Source Code</i> fungsi load_from_db	35
Gambar 3. 15 <i>Source Code</i> fungsi pop3	35
Gambar 3. 16 <i>Source Code</i> write_32	36
Gambar 3. 17 <i>Source Code</i> write_string	37
Gambar 3. 18 <i>Source Code</i> set_ecx	37
Gambar 3. 19 <i>Source Code</i> set_ebx_uclibc_offset	38
Gambar 3. 20 <i>Source Code</i> set_edx_ebx	39
Gambar 3. 21 <i>Source Code</i> fungsi build_v1 bagian pertama	40
Gambar 3. 22 <i>Source Code</i> fungsi build_v1 bagian kedua	41
Gambar 3. 23 <i>Source Code</i> fungsi fingerprint_version	42
Gambar 3. 24 <i>Source Code</i> fungsi upload_bins	42
Gambar 3. 25 Fungsi send_ropchain pada exploit.py	44
Gambar 3. 26 Fungsi super_admin pada exploit.py	45
Gambar 3. 27 <i>Source Code</i> fungsi needs_json	46
Gambar 3. 28 Fungsi main pada exploit.py bagian pertama	47
Gambar 3. 29 Fungsi main pada exploit.py	48
Gambar 3. 30 <i>Parser</i> pada exploit.py	48
Gambar 3. 31 <i>Folder</i> db	49
Gambar 3. 32 <i>Import</i> pada program run.py	51
Gambar 3. 33 <i>Source Code</i> fungsi is_valid_ip	51
Gambar 3. 34 <i>Source Code</i> fungsi get_non_empty_input	52
Gambar 3. 35 <i>Source Code</i> fungsi sanitize_input	52
Gambar 3. 36 <i>Source Code</i> fungsi run_exploit	53
Gambar 3. 37 <i>Source Code</i> fungsi main pada run.py	53
Gambar 3. 38 <i>Import</i> pada generate_report.py	54
Gambar 3. 39 <i>Source Code</i> fungsi sanitize_text	55
Gambar 3. 40 <i>Source Code</i> fungsi add_header	55
Gambar 3. 41 <i>Source Code</i> fungsi create_cover_page	56
Gambar 3. 42 <i>Source Code</i> fungsi generate_report bagian pertama	57
Gambar 3. 43 <i>Source Code</i> fungsi generate_report bagian kedua	58
Gambar 3. 44 <i>Source Code</i> fungsi generate_report bagian ketiga	59
Gambar 3. 45 Pengujian program eksploitasi	60
Gambar 3. 46 Pengujian program reporting	60
Gambar 3. 47 Pengambilan Waktu dengan Metode Otomatis	61
Gambar 3. 48 Langkah <i>Penetration Testing</i> dengan Metode Manual	62
Gambar 4. 1 Hasil Pengujian Program Eksploitasi	64
Gambar 4. 2 Hasil Pengujian Program Reporting	65
Gambar 4. 3 Hasil Pengujian Program Reporting	66
Gambar 4. 4 Hasil Pengujian Program Reporting	67



Pengembangan program otomatisasi uji penetrasi CVE-2023-30799 pada perangkat jaringan mikrotik berbasis python
JOHNAN CHRISTIAN, Dr. Ronald Adrian S.T., M.Eng.
Universitas Gadjah Mada, 2024 | Diunduh dari <http://etd.repository.ugm.ac.id/>

Gambar 4.5 Perbandingan Waktu Metode Otomatis dengan Manual 68
Gambar 4.6 Perbandingan durasi yang digunakan tiap kegiatan *penetration testing* 70
Gambar 4.7 Menjalankan kegiatan *reconnaissance* secara manual..... 71

DAFTAR TABEL

Tabel 2. 1 Ringkasan hasil skenario pada jurnal penelitian 6
Tabel 2. 2 Tabel Ringkasan Jurnal Penelitian 8
Tabel 3. 1 Tabel Spesifikasi Laptop 18
Tabel 3. 2 Tabel Spesifikasi VirtualBox 18
Tabel 3. 3 Tabel Spesifikasi Kali Linux 19
Tabel 3. 4 Tabel Spesifikasi Python 19
Tabel 3. 5 Tabel Spesifikasi router 20
Tabel 4. 1 Tabel Uji Fungsionalitas 63
Tabel 4. 2 Perbandingan aktivitas saat melakukan kegiatan *penetration testing* 68
Tabel 4. 3 Perbandingan taktik yang dapat dilakukan metode manual dengan otomatis 71
Tabel 4. 4 Perbandingan hasil yang didapat dari metode otomatis dengan manual .. 72

Internet di Indonesia menggunakan perangkat dari vendor yang bernama MikroTik.

MikroTik adalah salah satu vendor perangkat yang umum digunakan untuk kebutuhan infrastruktur internet di Indonesia. Indonesia sendiri memiliki 230.000 perangkat MikroTik yang terpapar ke internet. Karena hal itu, Indonesia menduduki peringkat ketiga di dunia dalam hal penggunaan MikroTik pada infrastrukturnya ('*Observations on cyber threat activity and vulnerabilities in Indonesia, Malaysia, Philippines and Thailand: The Shadowserver Foundation*', 2023). MikroTik juga memiliki kerentanan atau *vulnerability* pada perangkatnya. Salah satunya CVE-2023-30799 yang bisa mengakses router dengan meningkatkan akses admin menjadi *super-admin* melalui antarmuka *winbox* atau layanan web HTTP. Kerentanan ini memungkinkan penyerang mengeksekusi kode atau perintah melalui *backdoor*. Disebabkan oleh banyaknya perangkat MikroTik di Indonesia, diperlukan sebuah uji pengamanan efisien untuk CVE-2023-30799. Penelitian ini bertujuan untuk mengembangkan program eksploitasi dan *reporting* secara otomatis dengan menggunakan Python. Harapan penelitian ini agar dapat memperoleh solusi menghemat waktu dalam melakukan *security assessment* untuk CVE-2023-30799. Pengujian penelitian dibagi menjadi dua yaitu uji fungsionalitas dan uji perbandingan waktu antara metode manual dengan otomatis. Hasil penelitian menunjukkan bahwa program mampu dieksekusi tanpa kesalahan dan mampu menjalankan fungsi eksploitasi dan *reporting* dengan benar. Dengan diterapkannya program otomatisasi yang dibuat pada penelitian ini, proses *penetration testing* pada CVE-2023-30799 pada perangkat *router* MikroTik dapat dilakukan 79,1% lebih cepat daripada metode manual

Kata kunci: mikrotik, otomatisasi, *penetration testing*, keamanan siber

The Internet infrastructure in Indonesia relies extensively on devices provided by a vendor known as MikroTik. MikroTik stands as one of the prevalent equipment vendors catering to the internet infrastructure requirements within the Indonesian context. The nation boasts a staggering count of 230,000 MikroTik devices exposed to the internet. Consequently, Indonesia occupies the third position globally in terms of MikroTik device adoption for its infrastructure ('Observations on cyber threat activity and vulnerabilities in Indonesia, Malaysia, Philippines and Thailand: The Shadowserver Foundation', 2023). However, MikroTik devices are not immune to vulnerabilities. Notably, one such vulnerability identified is CVE-2023-30799, enabling unauthorized access to routers by elevating admin privileges to super-admin via the Winbox interface or HTTP web service. This vulnerability potentially allows attackers to execute arbitrary code or commands through a backdoor. Given the pervasive presence of MikroTik devices in Indonesia, there arises a necessity for an efficient security assessment concerning CVE-2023-30799. This study aims to develop an automated exploitation and reporting program utilizing Python. The research endeavors to streamline the time-consuming security assessment process associated with CVE-2023-30799, thus envisaging time-saving solutions. The research methodology involves two primary phases: functional testing and a comparative analysis of time efficiency between manual and automated methods. The findings indicate error-free execution of the program, proficiently carrying out exploitation functions and reporting accurately. By implementing the automation program devised in this study, the penetration testing process for CVE-2023-30799 on MikroTik router devices can be expedited by 79,1% compared to manual methods.

Keywords: mikrotik, automation, penetration testing, cyber security

BAB I

PENDAHULUAN

1.1 Latar belakang

Keamanan Informasi sangat penting pada era sekarang ini. Teknologi berkembang pesat dengan banyak inovasi dimana – mana. Perangkat Teknologi memiliki manfaat yang berfungsi untuk membantu aktivitas manusia. Semakin bermanfaat teknologi, semakin melekat teknologi itu dengan seseorang. Sebuah alat teknologi memiliki memori yang menyimpan sesuatu untuk mendukung fungsinya. Memori yang diisi dengan data pribadi seseorang akan lebih membantu seseorang tersebut dalam hal pribadinya. Tentunya teknologi bisa disalahgunakan oleh orang yang tidak bertanggung jawab. Oleh karena itu, diperlukan sebuah keamanan informasi yang dapat menjaga kerahasiaan dan keamanan dari data yang dimiliki oleh alat tersebut.

Internet di Indonesia menggunakan perangkat dari vendor yang bernama MikroTik. MikroTik adalah salah satu vendor perangkat yang umum digunakan untuk kebutuhan infrastruktur internet di Indonesia. Indonesia sendiri memiliki 230.000 perangkat MikroTik yang terpapar ke internet. Karena hal itu, Indonesia menduduki peringkat ketiga di dunia dalam hal penggunaan MikroTik di infrastrukturnya (*'Observations on cyber threat activity and vulnerabilities in Indonesia, Malaysia, Philippines and Thailand: The Shadowserver Foundation'*, 2023). MikroTik juga memiliki kerentanan atau *vulnerability* pada perangkatnya. Salah satunya CVE-2023-30799 yang bisa mengakses *router* dengan meningkatkan akses admin menjadi *super-admin* melalui antarmuka *winbox* atau layanan web HTTP. Kerentanan ini memungkinkan penyerang mengeksekusi kode atau perintah melalui *backdoor*

Untuk mengamankan sebuah informasi, salah satu metode yang dapat dilakukan adalah evaluasi keamanan perangkat atau komputer dengan cara menstimulasikan serangan siber pada perangkat atau komputer yang disebut *penetration testing* (uji penetrasi). Seseorang yang bekerja pada bidang *penetration testing* disebut dengan *pentester*. *Penetration testing* yang sering disingkat dengan *pentest* adalah metode yang sering dilakukan oleh ahli keamanan siber ketika ingin memeriksa keamanan sebuah perangkat secara langsung.

Saat melakukan *penetration testing* terdapat referensi pada internet yang berisi tentang kerentanan (*vulnerability*) yang ada. Referensi ini disebut dengan CVE (*Common Vulnerabilities and Exposures*). CVE dibuat oleh perusahaan non-profit bernama MITRE dengan pembiayaan dari *US Division of Homeland Security* atau Divisi Keamanan Dalam Negeri Amerika Serikat. CVE menganalisa skor kerentanan dengan menggunakan *Common Vulnerability Scoring System* (CVSS). CVE digunakan sebagai glosarium keamanan informasi untuk digunakan sebagai panduan mengamankan perangkat teknologi informasi

Penetration testing memiliki berbagai metode. Salah satu metode *penetration testing* adalah *Binary Exploitation*. *Binary Exploitation* telah lama menjadi masalah keamanan siber utama dan terus menjadi ancaman keamanan yang serius bagi setiap individu dan organisasi di seluruh dunia. Meskipun sejumlah besar upaya didedikasikan untuk itu, mengatasi dan mencegah *Binary Exploitation* tetap menjadi masalah yang masih belum terpecahkan (Liu dkk., 2022).

Penelitian yang berjudul Pengembangan Program Otomatisasi Uji Penetrasi CVE-2018-14847 Pada Perangkat Jaringan MikroTik Berbasis Python dan Shodan bertujuan untuk melakukan pengembangan *penetration testing* dengan otomatis (*autopentest*) menggunakan *search engine* SCADA Shodan dengan CVE-2018-14847. Target dari *penetration testing* adalah router MikroTik dengan versi 6.42 kebawah. Hasil dari penelitian adalah sebuah laporan otomatis yang berisi *username* dan *password* yang ada pada perangkat router MikroTik. Otomatisasi yang dilakukan adalah melakukan *pentest* sekaligus melakukan *auto-reporting* hanya dengan menjalankan satu berkas kode. Penelitian juga menggunakan target *router* fisik yang memiliki IP publik yang didapatkan dengan menggunakan *search engine* SCADA Shodan. Kerentanan atau CVE yang digunakan pada penelitian ini sudah usang dan banyak perangkat MikroTik yang sudah tidak terpengaruh dengan kerentanan pada CVE-2018-14847 (Sanjaya. Y., 2023)

Dari permasalahan pada penelitian diatas, maka penulis berinisiatif untuk melakukan penelitian yang berjudul Implementasi Otomatisasi *Pentest* pada *Binary Exploitation* berbasis kali linux dengan CVE-2023-30799. Penelitian ini diharapkan dapat memperbaharui sebuah metode *pentest* yang kompatibel dengan versi 6.42 sampai 6.49 yang lebih baru dan lebih relevan pada masa sekarang ini. Program pada

penelitian diharapkan dapat mempermudah dan mempersingkat waktu dalam melakukan *penetration testing*. Program akan melakukan eksploitasi otomatis dan melaporkan bukti konsep (*proof of concept*) yang telah dilakukan sehingga respon yang diperlukan dapat segera dilakukan untuk mengamankan perangkat jaringan.

1.2 Rumusan Masalah

Berdasarkan penjelasan pada latar belakang, maka perumusan masalah yang diangkat adalah bagaimana cara mempersingkat waktu *penetration testing* dengan menggunakan program yang didalamnya termasuk proses eksploitasi dan pelaporan kerentanan CVE-2023-30799 pada perangkat jaringan MikroTik menggunakan metode otomatisasi berbasis Python.

1.3 Batasan Masalah

Penelitian proyek akhir ini memiliki beberapa batasan masalah, adapun batasan masalah tersebut adalah sebagai berikut:

1. Program yang dikembangkan berfokus pada proses eksploitasi, dan pelaporan secara otomatis untuk kerentanan CVE-2023-30799 pada perangkat jaringan MikroTik.
2. Perangkat jaringan MikroTik yang akan di uji merupakan perangkat jaringan MikroTik virtual, menggunakan arsitektur x86 dan diimplementasikan secara lokal, hal ini dikarenakan hasil eksplotasi adalah penciptaan *backdoor* yang hanya dapat dihapus atau dinonaktifkan jika perangkat router di-*restart* atau *reset*. Hal tersebut membahayakan bila menggunakan *physical router* karena menambah sebuah *backdoor access* pada sebuah perangkat.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk

1. Pengembangan sebuah program otomatisasi yang mampu melakukan eksploitasi, dan pelaporan secara otomatis untuk proses uji penetrasi kerentanan CVE-2023-30799 pada perangkat jaringan MikroTik
2. Penghematan waktu saat proses *penetration testing*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Pemberian solusi dan respon yang tepat untuk kerentanan CVE-2023-30799 pada perangkat MikroTik
2. Penghematan waktu pekerjaan *pentester* atau *network administrator* saat melakukan *penetration testing* CVE-2023-30799 pada perangkat *router* MikroTik
3. Pembuktian pada metode *penetration testing* secara otomatis yang dapat dilakukan pada *router* MikroTik

1.6 Sistematika Penelitian

Bagian sistematika penulisan memuat garis besar dari laporan proyek akhir yang dikerjakan dengan sistematika penelitian dibagi menjadi lima bab. Bagian tersebut adalah sebagai berikut:

BAB I PENDAHULUAN, berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan yang digunakan penulis pada laporan ini.

BAB II TINJAUAN PUSTAKA DAN DASAR TEORI, Dalam tinjauan pustaka dan dasar teori, penulis membahas penelitian mereka terhadap artikel dan jurnal tentang penelitian sebelumnya, yang digunakan sebagai teori pendukung dan sebagai dasar untuk penelitian mereka saat ini, proyek akhir.

BAB III METODE PENELITIAN, Metode penelitian mencakup bahan, peralatan, langkah-langkah penelitian, rancangan sistem, dan analisis data penelitian proyek akhir penulis.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN, Dalam bab hasil penelitian dan pembahasan, penulis membahas temuan, analisis, dan argumen tentang temuan penelitian proyek akhir.

BAB V PENUTUP, Bab penutup berisi kesimpulan terkait penelitian proyek akhir ini serta saran dari penulis untuk pengembangan penelitian selanjutnya

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian yang berjudul *Automated versus Manual Approach of Web Application Penetration Testing*. Penelitian tersebut membandingkan metode *penetration testing* secara manual atau otomatis. Penelitian dilakukan dengan skenario eksploitasi pada aplikasi web. Skenario yang dipakai pada penelitian adalah *clickjacking* yang digunakan untuk kerentanan *sameorigin*. Skenario *file upload* untuk kerentanan XSS. Skenario *sensitive files* untuk kerentanan *public domain*. Skenario terakhir adalah *business logic* untuk kerentanan *logic failure*. Hasil yang didapatkan pada penelitian tersebut bahwa dalam *penetration testing* secara manual lebih efektif dalam mendapatkan hasil eksploitasi yang lebih banyak tetapi dengan metode otomatis dapat melakukan sebuah eksploitasi yang mudah dengan cepat. Hasilnya adalah metode *penetration testing* secara otomatis atau manual memiliki kelebihan dan kekurangannya masing – masing pada skenario yang berbeda – beda (Singh dkk., 2020). Ringkasan dari hasil penelitian *Automated versus Manual Approach of Web Application Penetration Testing* dapat dilihat pada tabel 2.1.

Tabel 2. 1 Ringkasan hasil skenario pada jurnal penelitian

Skenario	Kerentanan	Manual	Otomatis
<i>Clickjacking</i>	<i>Sameorigin</i>	✓	✗
<i>File Upload</i>	XSS	✓	✗
<i>Sensitive Files</i>	<i>Public Domain</i>	✗	✓
<i>Business Logic</i>	<i>Logic Failure</i>	✓	✗

Penelitian yang berjudul Pengembangan Program Otomatisasi Uji Penetrasi CVE-2018-14847 Pada Perangkat Jaringan MikroTik Berbasis Python dan Shodan. Penelitian ini bertujuan untuk mempersingkat waktu dalam melakukan uji penetrasi pada perangkat MikroTik. Eksploitasi yang dihasilkan oleh penelitian ini adalah *username* dan *password* dari sebuah perangkat *router* MikroTik. Penelitian ini membuktikan bahwa dengan menggunakan otomatisasi pada uji penetrasi dapat mempermudah dan mempercepat pekerjaan seorang *pentester*. *Penetration testing* pada penelitian menggunakan tahap, *scanning* sebagai tahap *reconnaissance*, *exploit*, dan yang terakhir adalah tahap *reporting*. Semua tahap pada penelitian dilakukan secara otomatis. Tahap *reporting* yang dilakukan secara otomatis juga di-generate dengan ekstensi PDF (Antono, Y. 2023).

Penelitian yang berjudul *Binary Exploitation in Industrial Control Systems: Past, Present and Future*. Penelitian berfokus pada salah satu metode *penetration testing* yaitu *binary exploitation*. Skenario dilakukan dengan melibatkan sistem kontrol industri yang menjadi target eksploitasi. Penelitian membahas pertahanan berbasis deteksi dan bagaimana mitigasi dari metode *binary exploitation*. Hasil dari penelitian adalah deteksi dari pertahanan deteksi pada eksploitasi *binary exploitation*. Penelitian menyatakan bahwa *binary exploitation* adalah masalah yang serius dan sulit ditangani. *Binary exploitation* sulit ditangani karena serangannya langsung pada memori sehingga sulit serangan dikenali (Liu dkk., 2022).

Tabel 2. 2 Tabel Ringkasan Jurnal Penelitian

No	Judul Penelitian	Tahun	Metode Penelitian		Metode	Tujuan
			BinEx	AutoPen		
1	Singh N., Meherhomji V., Chandavarkar B., (2020) <i>Automated versus Manual Approach of Web Application Penetration Testing</i>	2020		✓	Penelitian melakukan <i>penetration testing</i> pada sebuah <i>website</i> dengan empat skenario untuk empat kerentanan. Skenario adalah <i>clickjacking</i> , <i>File upload</i> , <i>sensitive files</i> , <i>Business logic</i>	Tujuan penelitian adalah membandingkan metode otomatis dengan manual menggunakan empat buah skenario yang berbeda. Hasil yang didapatkan berupa keuntungan dan kelemahan dari masing – masing metode
2	Liu Q, Bao K, Hagenmeyer V., (2022) <i>Binary Exploitation in Industrial Control Systems: Past, Present and Future</i>	2022	✓		Penelitian membuat sebuah pertahanan untuk <i>penetration testing</i> dengan metode <i>binary exploitation</i> . Pertahanan yang dibuat adalah teknik mendeteksi serangan <i>binary exploitation</i> menggunakan berbagai metode. Deteksi yang dilakukan adalah mendeteksi kode yang berjalan, <i>rop scan</i> , dan mendeteksi anomali pada program	Tujuan penelitian adalah menemukan mitigasi yang tepat untuk serangan <i>binary exploitation</i> . Mitigasi dilakukan dengan memeriksa anomali pada kode. Menemukan deteksi tahap pertama yang dapat mencegah serangan <i>binary exploitation</i> merusak lebih lagi