Merkurius : Jurnal Riset Sistem Informasi dan Teknik Informatika
Volume 3, Nomor 1, Tahun 2025
e-ISSN: 3031-8912; p-ISSN: 3031-8904; Hal 169-177
DOI: https://doi.org/10.61132/merkurius.v3i1.620
Available online at: https://journal.arteii.or.id/index.php/Merkurius

Keamanan Kernel Linux : Pendekatan Hardening dan Perlindungan terhadap Serangan Eksploitasi

Zalfa Dewi Zahrani[1*], Novianto Andi Hardiansyah[2], Elkin Rilvani[3]
[1,2,3]Universitas Pelita Bangsa, Indonesia
**Email: [1]zalfadewizahrani@gmail.com, [2]andihardiansyahh05@gmail.com, [3]elkin.rilvani@pelitabangsa.ac.id*

Alamat: Jl. Inspeksi Kalimalang No.9, Cibatu, Cikarang Sel., Kabupaten Bekasi, Jawa Barat 17530
*Korespondensi penulis: zalfadewizahrani@gmail.com**

***Abstract**. Linux kernel security is a critical aspect of ensuring the integrity and stability of operating systems. Vulnerabilities like Dirty COW (CVE-2016-5195) illustrate how exploitative threats can severely impact systems, particularly those that are not regularly updated. This study analyzes the working mechanism of Dirty COW, its impact, and mitigation strategies based on Linux kernel hardening techniques, including the use of security modules like SELinux and AppArmor, as well as the Address Space Layout Randomization (ASLR) technique. Through attack simulations and mitigation evaluations, the findings emphasize the importance of regularly applying kernel patches to maintain system security. This study aims to provide practical guidance for enhancing Linux kernel resilience against exploitation attacks.*

***Keywords**:Dirty COW, Mitigation, Hardening, SELinux, ASLR.*

Abstrak. Keamanan kernel Linux adalah salah satu aspek penting dalam menjamin integritas dan stabilitas sistem operasi. Kerentanan seperti Dirty COW (CVE-2016-5195) menunjukkan bagaimana ancaman eksploitatif dapat memengaruhi sistem secara serius, terutama pada perangkat yang tidak diperbarui. Kajian ini menganalisis mekanisme kerja Dirty COW, dampaknya, dan langkah mitigasi berbasis hardening kernel Linux, termasuk penggunaan modul seperti SELinux, AppArmor, dan teknik Address Space Layout Randomization (ASLR). Simulasi serangan dan evaluasi langkah mitigasi menunjukkan pentingnya penerapan patch kernel secara berkala untuk menjaga keamanan sistem. Kajian ini bertujuan memberikan panduan praktis untuk meningkatkan ketahanan kernel Linux terhadap serangan eksploitasi.

Kata kunci: Dirty COW, mitigasi, hardening, SELinux, ASLR.

1. LATAR BELAKANG

Keamanan sistem operasi merupakan komponen penting dalam menjamin integritas, kerahasiaan, dan ketersediaan data pada perangkat komputasi. Linux, sebagai salah satu sistem operasi yang dominan di server, perangkat IoT, dan superkomputer, menghadapi tantangan konstan dari berbagai ancaman keamanan. Salah satu ancaman serius adalah eksploitasi kernel, yang dapat memberikan akses tak terbatas kepada penyerang terhadap sistem.

Dirty COW (Copy-On-Write) adalah salah satu kerentanan terkenal dalam kernel Linux, pertama kali ditemukan pada tahun 2007 namun baru diungkap secara publik pada tahun 2016. Kerentanan ini (CVE-2016-5195) memungkinkan penyerang untuk meningkatkan hak akses mereka ke level root dengan memanfaatkan race condition dalam

Received: Desember 05, 2024; Revised: Desember 23, 2024; Accepted: Januari 04, 2025; Published: Januari 06, 2025

mekanisme pengelolaan memori. Dampaknya sangat signifikan, terutama pada sistem yang tidak memperbarui kernel secara berkala.

Untuk mengatasi ancaman seperti Dirty COW, pendekatan hardening pada kernel Linux menjadi sangat relevan. Teknik-teknik seperti mitigasi berbasis patch, penggunaan modul keamanan seperti SELinux dan AppArmor, serta mekanisme tambahan seperti Address Space Layout Randomization (ASLR) dapat membantu mengurangi risiko eksploitasi. Kajian ini akan membahas pendekatan hardening secara mendalam, dengan fokus pada penanganan Dirty COW sebagai studi kasus, untuk memberikan gambaran strategi perlindungan yang efektif terhadap serangan eksploitasi.

2. KAJIAN TEORITIS

Linux, sebagai sistem operasi open-source yang banyak digunakan, menawarkan fleksibilitas tinggi, tetapi juga menghadapi ancaman yang signifikan. Salah satu ancaman tersebut adalah kerentanan kernel, yang dapat dimanfaatkan oleh penyerang untuk mendapatkan kontrol penuh atas sistem.

Dirty COW, dengan identifikasi CVE-2016-5195, adalah salah satu contoh kerentanan kernel Linux yang paling terkenal. Kerentanan ini ditemukan pada mekanisme Copy-On-Write (COW), yang merupakan bagian dari pengelolaan memori. COW dirancang untuk meningkatkan efisiensi dengan memungkinkan proses untuk berbagi halaman memori yang sama, sampai salah satu proses mencoba memodifikasi halaman tersebut. Namun, Dirty COW mengeksploitasi race condition dalam mekanisme ini, memungkinkan penyerang untuk melakukan eskalasi hak akses. Dalam skenario ini, file yang seharusnya hanya dapat dibaca dapat dimodifikasi oleh penyerang untuk mendapatkan hak akses yang lebih tinggi.

Untuk mengatasi ancaman seperti Dirty COW, berbagai teknik hardening kernel telah dikembangkan. Patch kernel adalah langkah pertama dan paling penting untuk memperbaiki kerentanan yang ditemukan. Dengan memperbarui kernel ke versi terbaru yang telah diperbaiki, eksploitasi Dirty COW dapat dicegah. Selain itu, modul keamanan seperti Security-Enhanced Linux (SELinux) dan AppArmor memberikan kontrol yang lebih granular atas akses proses ke sumber daya sistem. Modul ini bekerja dengan menerapkan kebijakan keamanan berbasis konteks, sehingga membatasi ruang lingkup operasi proses.

Teknik mitigasi lainnya termasuk Address Space Layout Randomization (ASLR), yang mengacak lokasi memori proses untuk menyulitkan penyerang dalam memprediksi

lokasi yang rentan. ASLR bekerja dengan memindahkan lokasi stack, heap, dan library ke alamat memori yang berbeda setiap kali program dijalankan. Meskipun ASLR tidak secara langsung mencegah eksploitasi Dirty COW, teknik ini dapat menambah lapisan kesulitan bagi penyerang.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan deskriptif kualitatif dengan metode studi kasus untuk menganalisis kerentanan Dirty COW pada kernel linux serta langkah mitigasi berbasis hardening. Pendekatan ini dipilih untuk memberikan gambaran mendalam mengenai mekanisme serangan, dampaknya, dan efektivitas mitigasi yang diterapkan. Fokus penelitian terletak pada analisis eksploitasi Dirty COW (CVE-2016-5195) melalui simulasi teknis dan tinjauan literatur, di mana penelitian ini mengevaluasi dampak kerentanan terhadap sistem Linux serta langkah-langkah mitigasi yang mencakup penerapan patch, penggunaan modul keamanan, dan teknik hardening lainnya.

Data dalam penelitian ini dikumpulkan melalui tiga metode utama: tinjauan literatur yang mencakup analisis publikasi ilmiah, dokumentasi kernel linux, dan laporan keamanan terkait Dirty COW; simulasi teknis yang melibatkan pengujian serangan Dirty COW di lingkungan virtual menggunakan distribusi linux yang rentan, dan analisis dokumentasi untuk memahami implementasi patch dan teknik mitigasi lainnya. Setelah data diperoleh, analisis dilakukan secara deskriptif untuk memahami mekanisme kerja kerentanan Dirty COW serta eksploitasi yang dihasilkan, efektivitas patch kernel dalam menangani kerentanan tersebut, serta performa dan keamanan sistem setelah penerapan teknik hardening seperti SELinux, AppArmor, dan ASLR.

4. HASIL DAN PEMBAHASAN

Pengaruh dirty cow terhadap kernel

Diungkapkan pada bulan Oktober 2016, kerentanan Dirty COW telah ada dalam kode inti sistem operasi populer tersebut sejak tahun 2007. Pada saat pengungkapan tersebut, kerentanan ini memengaruhi sebagian besar distribusi Linux dan versi kernel. Distribusi yang terpengaruh meliputi:

Table 1. Beberapa versi kernel yang terpengaruh

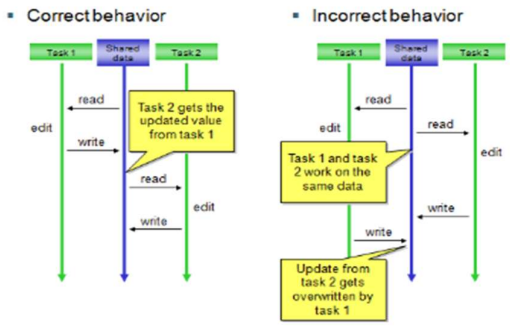
CentOS Linux 7.x	CentOS Linux 6.x	CentOS Linux 5.x
Debian Linux sid	Debian Linux jessie	Debian Linux wheezy
Ubuntu Linux yakkety	Ubuntu Linux trusty	Ubuntu Linux precise(LTS 12.04
Red Hat Enterprise Linux 7.x	Red Hat Enterprise Linux 6.x	Red Hat Enterprise Linux 5.x
SUSE Linux Enterprise 11	SUSE Linux Enterprise 12	Debian Linux stretch

Kerentanan tersebut mengganggu kernel Linux sejak versi 2.6.22 (yang dirilis pada September 2007), tetapi telah ditambal dalam versi 4.8.3, 4.7.9, 4.4.26, dan yang lebih baru dari kernel Linux. Meskipun demikian, ada banyak bukti bahwa Dirty COW digunakan oleh para penjahat di alam liar untuk mengambil alih server yang belum ditambal, meskipun pelaksanaan eksploitasi itu sendiri tidak meninggalkan jejak dalam log server web. Ia juga berfungsi sebagai komponen dalam malware Android yang ditemukan di alam liar.

Cara kerja eksploitasi Dirty COW

Eksploitasi Dirty COW merupakan kasus eksploitasi kerentanan kondisi balapan. Dalam kasus ini, penyerang memanfaatkan izin root yang didapatkan kernel saat berjalan dan menciptakan kondisi balapan yang memungkinkan peningkatan hak istimewa dari pengguna tingkat rendah ke pengguna dengan hak istimewa root penuh.

Ketika sebuah program perangkat lunak yang sedang berjalan menemui beberapa jalur kode yang dijalankan secara bersamaan, hal tersebut dapat "membingungkan" perangkat lunak dan menciptakan "perlombaan" antara jalur kode, yang menyebabkan jalur kode tersebut berakhir dalam urutan yang berbeda dari yang diantisipasi, sehingga mengakibatkan bug dan perilaku aplikasi yang tidak diharapkan.



Gambar 1. Kondisi balapan dirty COW

Dalam kasus eksploitasi Dirty COW, persaingan terjadi antara dua operasi: satu operasi menulis ke pemetaan memori COW dan operasi lainnya membuangnya secara terus-menerus. Ketika operasi ini berulang tanpa henti, kernel dapat dibingungkan dengan menulis data ke pemetaan memori hanya-baca alih-alih terlebih dahulu membuat salinan data pribadi. Atau, sistem target akan mogok.

Data analisis serangan eksploitasi drity COW

Meskipun bug kernel Linux ini telah ditambal, penyerang terus menggunakannya dalam aktivitas mereka. Ancaman yang terus berlanjut ini membuat Dirty COW perlu diwaspadai, bahkan bertahun-tahun setelah penemuan dan perbaikan awal.

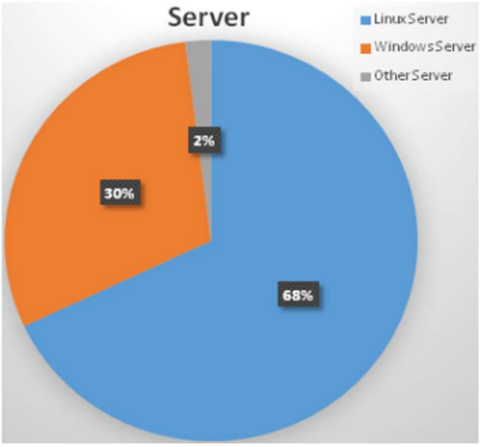
Pada pertengahan tahun 2023, tim keamanan Akamai menemukan bahwa penyerang menargetkan situs eCommerce menggunakan Magento 2. Serangan tersebut, yang diberi nama Xurum dan ditelusuri ke Rusia, menggunakan eksploitasi Dirty COW untuk eskalasi hak istimewa pada server Linux. Pendekatan ini efektif pada server yang menjalankan versi kernel lama yang belum diperbaiki.

```
#!/bin/bash
# Version: 1.5

RED="\033[1;31m"
YELLOW="\033[1;33m"
GREEN="\033[1;32m"
BOLD="\033[1m"
RESET="\033[0m"

SAFE_KERNEL="SAFE_KERNEL"
SAFE_KPATCH="SAFE_KPATCH"
MITIGATED="MITIGATED"
VULNERABLE="VULNERABLE"

MITIGATION_ON='CVE-2016-5195 mitigation loaded'
MITIGATION_OFF='CVE-2016-5195 mitigation unloaded'


VULNERABLE_VERSIONS=(
    # RHEL5
    "2.6.18-8.1.1.el5"
    "2.6.18-8.1.3.el5"
    "2.6.18-8.1.4.el5"
    "2.6.18-8.1.6.el5"
    "2.6.18-8.1.8.el5"
    "2.6.18-8.1.10.el5"
```

Gambar 2. eksploitasi bernama Xurum menggunakan dirty COW

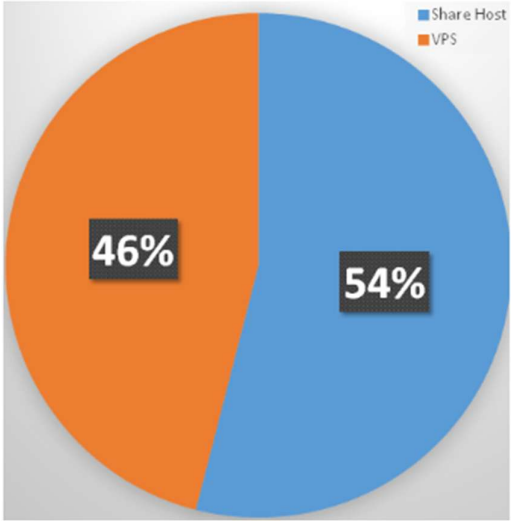
Penyerang *cyber* sering kali menggunakan metode coba-coba untuk mengeksploitasi kerentanan yang diketahui, meskipun kerentanan tersebut telah diperbaiki. Mereka berfokus pada kelemahan ini karena kerentanan yang belum ditambal tetap menjadi masalah utama dalam perusahaan IT dan *DevOps.* Masalah yang belum diperbaiki ini berkontribusi terhadap sekitar 60% dari semua pelanggaran data, yang menyoroti signifikansinya dalam keamanan *cyber.*

Di bagian negara Bangladesh juga mengalami serangan eskploitasi dirty COW, sebanyak 200 server web diperiksa untuk tujuan analisis awal. Di antara 200 server tersebut, 68% ditemukan menggunakan berbagai distribusi Linux, 30% menggunakan sistem operasi Windows, dan 2% menggunakan berbagai sistem operasi lainnya seperti yang ditunjukkan pada gambar 3.

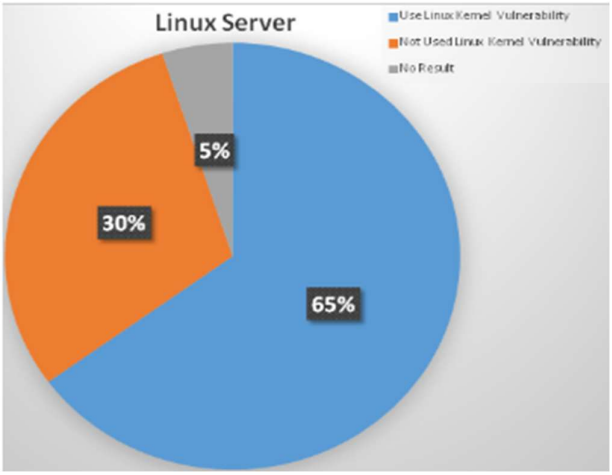


Gambar 3. Penggunaan berbagai sistem operasi

Efek keparahan serangan Dirty COW sebagian bergantung pada kategori server Linux host. Server yang diperiksa dalam studi ini ditemukan menggunakan dua kategori: VPS dan Host bersama. Kedua kategori tersebut rentan berdasarkan kernel Linux. Efek pada VPS lebih kecil karena server ini hanya menghosting satu pengguna. Efek serangan Dirty COW paling besar pada server host bersama karena satu server dapat menghosting banyak aplikasi web. Di antara server rentan yang diperiksa, 54% adalah host bersama sementara 46% adalah VPS seperti yang ditunjukkan pada Gambar 4. Karena semua kernel Linux tidak rentan terhadap serangan Dirty COW, di antara server Linux yang diperiksa sekitar 65% ditemukan rentan terhadap serangan seperti yang ditunjukkan pada Gambar 10. 30% server Linux tidak rentan terhadap serangan Dirty COW karena menggunakan Kernel Linux yang aman. Kerentanan 5% server tidak dapat diprediksi.



Gambar 4. Server linux menggunakan VPS dan Host bersama



Gambar 5. Statistik kerentanan kernel Linux yang diperiksa

5. KESIMPULAN DAN SARAN

Dirty COW merupakan kerentanan signifikan dalam kernel Linux yang memungkinkan penyerang meningkatkan hak akses ke tingkat root, dengan dampak luas pada server dan perangkat berbasis Linux yang tidak diperbarui. Untuk mengurangi risiko eksploitasi, langkah mitigasi seperti penerapan patch kernel, aktivasi modul keamanan seperti SELinux dan AppArmor, serta teknik Address Space Layout Randomization (ASLR) telah terbukti efektif. Oleh karena itu, administrator sistem disarankan untuk secara rutin memperbarui kernel Linux ke versi terbaru dan mengaktifkan modul keamanan yang sesuai dengan kebutuhan sistem. Selain itu, peningkatan kesadaran akan

pentingnya pengelolaan keamanan sistem operasi melalui pelatihan keamanan dan penerapan kebijakan pengawasan berkala sangat penting untuk menjaga integritas sistem. Penelitian lebih lanjut juga diperlukan untuk mengembangkan teknik mitigasi baru yang lebih adaptif terhadap ancaman eksploitasi yang terus berkembang.

DAFTAR REFERENSI

Akamai. (2023). *Xurum Exploits and Dirty COW: An Analysis*. Retrieved from https://www.akamai.com

Anderson, R. J. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems* (3rd ed.). Wiley.

Bovet, D. P., & Cesati, M. (2005). *Understanding the Linux Kernel* (3rd ed.). O'Reilly Media.

Canonical. (2023). *Mitigation of Dirty COW on Ubuntu Systems*. Retrieved from https://ubuntu.com

Chou, A., et al. (2001). An empirical study of operating systems errors. *ACM Symposium on Operating Systems Principles*, 73-88.

Corbet, J., Kroah-Hartman, G., & McPherson, A. (2009). *Linux Kernel Development*. Addison-Wesley.

Cowan, C., et al. (1998). StackGuard: Automatic adaptive detection and prevention of buffer-overflow attacks. *USENIX Security Symposium Proceedings*, 63-78.

Google Project Zero. (2016). *Dirty COW Exploit Analysis*. Retrieved from https://googleprojectzero.blogspot.com

Grsecurity. (2016). *Kernel Hardening: The Role of Grsecurity and PaX*. Retrieved from https://grsecurity.net

Intel. (2016). *Security Best Practices for Linux Systems*. Retrieved from https://www.intel.com

Love, R. (2010). *Linux Kernel Development* (3rd ed.). Addison-Wesley.

Mitre. (2016). CVE-2016-5195. Retrieved from https://cve.mitre.org

NIST. (2016). National Vulnerability Database: CVE-2016-5195. Retrieved from https://nvd.nist.gov

Openwall Project. (2017). *Linux Kernel Security Hardening: Openwall Patches*. Retrieved from https://openwall.com

PaX Team. (2016). *Address Space Layout Randomization (ASLR): Enhancing Linux Kernel Security*. Retrieved from https://pax.grsecurity.net

Red Hat. (2016). *Dirty COW Vulnerability and Its Impacts*. Retrieved from https://www.redhat.com

Stallings, W. (2017). *Operating Systems: Internals and Design Principles* (9th ed.). Pearson.

Tanenbaum, A. S., & Bos, H. (2014). *Modern Operating Systems* (4th ed.). Pearson.

Torvalds, L. (2007). *Linux Kernel Source Code*. Available at https://kernel.org

Wagner, D., & Dean, D. (2001). Intrusion detection via static analysis. *Proceedings of the IEEE Symposium on Security and Privacy*, 156-168.