

PUNGGAWA CYBERSECURITY MAGAZINE

www.punggawa.com

Vol.07

CYBERSECURITY OPERATION & RESILIENCE

Publisher





TENTANG KAMI

PUNGGAWA merupakan istilah dari kebudayaan Indonesia yang mengacu pada sosok pemimpin atau figur berwibawa yang terkenal akan kepemimpinannya, tanggung jawab, dan arahan dalam suatu komunitas. Istilah ini melambangkan dedikasi terhadap keunggulan kepemimpinan, praktik etik, atribut kekuatan, kearifan, dan kepercayaan, serta sikap pelindung terhadap mereka yang berada dalam lingkup pengawasannya.

Kami Merupakan **PUNGGAWA**

Tim **PUNGGAWA** didirikan pada tahun 2018 dan mulai memberikan layanan kepada pelanggan pada tahun 2019, dengan memulai dari layanan Penetration Testing. Kami telah berhasil merealisasikan dan menembus pasar keamanan siber di Indonesia. Dalam kemitraan dengan klien, kami menyediakan solusi dan layanan keamanan siber yang dirancang untuk meningkatkan postur keamanan secara komprehensif, menutup celah, dan memantau kerentanan secara berkelanjutan melalui operasi dan dukungan yang persisten dengan mengimplementasikan identifikasi, perlindungan, deteksi, respons, dan pemulihan.



VISI

Menjadi mitra pilihan utama di bidang keamanan siber sebagai bentuk berkontribusi nyata dalam mewujudkan dunia yang lebih aman dalam transformasi digital.

MISI



MENGHADIRKAN HASIL TERBAIK.

Dengan dukungan tim yang andal dan dedikasi tinggi pada proses yang tepat, kami menghadirkan solusi yang memberikan kesuksesan nyata bagi klien kami.



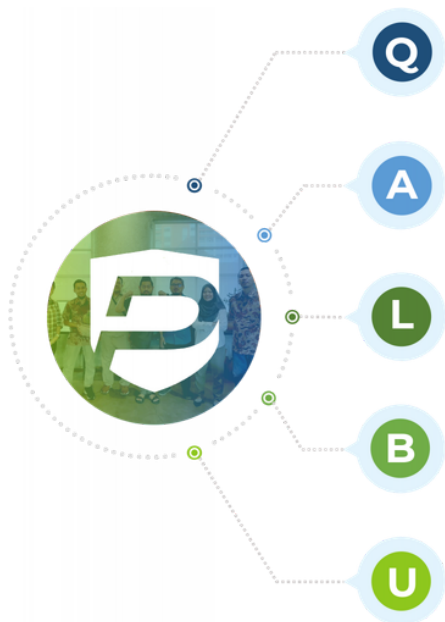
MEMBANGUN BUDAYA BELAJAR DAN KESADARAN.

Kami senantiasa menumbuhkan budaya untuk terus belajar dan meningkatkan kesadaran (awareness) di dalam internal tim kami.



BERBAGI DAN BERKOLABORASI BERSAMA KOMUNITAS.

Kami mengedepankan semangat kerja sama yang kuat, baik sebagai satu tim yang solid di dalam organisasi maupun sebagai mitra bagi komunitas yang lebih luas.



CORE VALUE, QALBU

Quick and High Quality Response

Dalam keamanan siber, respons yang cepat terhadap ancaman sangat krusial. Di PUNGGAWA, kami mengutamakan aksi cepat untuk mengidentifikasi dan meredakan ancaman siber, memastikan aset digital klien terlindungi secara efisien dan efektif. Respons berkualitas tinggi juga berarti memberikan solusi yang menyeluruh dan berpengetahuan luas terhadap tantangan keamanan siber yang kompleks.

Attitude is Everything

Sikap positif dan proaktif sangat penting di PUNGGAWA. Ini melibatkan usaha untuk selalu mendahului ancaman potensial, antusiasme untuk belajar tentang tren keamanan baru, dan memelihara ketahanan mental menghadapi ancaman siber yang terus berkembang. Sikap yang berorientasi pada peningkatan berkelanjutan esensial dalam beradaptasi dengan dinamika keamanan siber.

Listen, Learn, Lead & Succeed

Nilai ini menekankan pentingnya pembelajaran berkelanjutan dalam bidang keamanan siber. Dengan mendengarkan secara aktif kebutuhan klien dan perkembangan industri, tim PUNGGAWA tetap terdepan dan terinformasi. Pembelajaran ini berujung pada kepemimpinan di bidangnya, pengembangan solusi inovatif, dan kesuksesan dalam melindungi klien dari ancaman siber.

Be a Problem Solver

Keamanan siber seringkali tentang menyelesaikan teka-teki yang kompleks yang dihadirkan oleh ancaman siber. Di PUNGGAWA, kami menekankan pentingnya pendekatan yang berorientasi pada solusi, baik itu dalam mengatasi serangan siber yang rumit, menavigasi kerentanan jaringan yang kompleks, atau menemukan solusi kreatif untuk tantangan keamanan baru.

Unity is Our Strength

Kami memahami tantangan kewirausahaan dan mengetahui bahwa keamanan siber memerlukan kerja sama tim dan kolaborasi, baik di dalam organisasi maupun dengan klien, mitra, dan komunitas keamanan siber yang lebih luas. Kesatuan dalam tujuan dan aksi menjamin pertahanan yang lebih kuat terhadap ancaman siber dan postur keamanan yang lebih tangguh.

Salam hangat para pembaca setia **Punggawa Cybersecurity Magazine**

Selamat datang di Punggawa Cybersecurity Magazine Volume 7, edisi yang mengangkat dinamika operasi keamanan siber modern, sebuah lanskap di mana keberhasilan pertahanan sering kali tidak terlihat, namun berperan penting dalam menjaga keberlangsungan organisasi di tengah ancaman yang terus berkembang.

Melalui tema **“Cybersecurity Operation & Resilience: Mata yang Tidak Pernah Tidur”**, edisi ini mengajak pembaca memahami bagaimana keamanan siber dijalankan di balik layar mulai dari peran SOC yang terus memantau, kesiapan tim respons insiden, hingga kolaborasi antara red team dan blue team dalam menjaga ketahanan sistem.

Salah satu sorotan utama adalah pembahasan dilema kejujuran dalam praktik Red Team, khususnya dalam konteks **Purple Teaming**, yang menyoroti keseimbangan antara transparansi dan efektivitas simulasi serangan. Kami juga mengulas **tantangan kesiapan CSIRT saat krisis siber** terjadi, serta fenomena Paradoks SOC, di mana pusat operasi keamanan yang paling efektif justru sering kali terlihat paling “sunyi”.

Di sisi teknologi, edisi ini membahas bagaimana kecerdasan buatan mulai memengaruhi praktik keamanan modern, mulai dari otomatisasi vulnerability scanning melalui **Droid LLM Hunter** hingga **Evolusi CTF dan bug bounty di era AI**. Kami juga menghadirkan berbagai tools dan product spotlight, seperti **Tools DursGo, Burp Suite MCP Server, platform SacTi** (Security Cyber Threat Intelligence), serta pemanfaatan HashiCorp Vault dalam mendukung ekosistem cyber resilience.

Sebagai penutup, kami mengangkat konsep **Threat Intelligence-Driven Red Teaming**, sebuah pendekatan yang memanfaatkan intelijen ancaman nyata untuk menciptakan simulasi serangan yang lebih relevan dan mendekati kondisi dunia nyata.

Kami berharap **Punggawa Cybersecurity Magazine Volume 7** dapat menjadi referensi bagi para profesional keamanan, praktisi teknologi, dan pengambil kebijakan untuk memahami bahwa di balik setiap ancaman yang berhasil dicegah, terdapat operasi keamanan yang bekerja tanpa henti.

Selamat Membaca.

Iwan Setiawan
CEO PUNGGAWA CYBERSECURITY

TABLE OF CONTENTS

07

Dilema Kejujuran Red Team

Haruskah Membuka Semua
Kartu Demi Purple Teaming

10

CSIRT Tanpa Ujian

Mengapa Banyak Tim Gagal
Saat Krisis Siber Terjadi

13

Paradoks SOC

Saat yang Paling Efektif
Justru Paling Sunyi

17

DevSecOps

Otomatisasi
Vulnerability Scanning
dengan Droid LLM
Hunter



23

Evolusi CTF dan Bug Bounty di Era AI



28

Tools : DursGo

Web Vulnerability Scanner
Berbasis Go dengan
Analisis AI

34

Burp Suite MCP Server

Cyber Security:
MCP (Model Context
Protocol) untuk Pentest

41

SacTi

Platform Cyber Threat
Intelligence untuk
Proactive Defense (Product
Spotlight Punggawa)

52

HashiCorp Vault

Ekosistem Cyber Resilience
(Product Spotlight
Prinsipal)

63

Threat Intelligence

Driven Red Teaming:
Simulasi Serangan Berbasis
Ancaman Nyata

PUNGGAWA MAGAZINE

VOLUME 7

www.punggawa.com

Editor-in-Chief

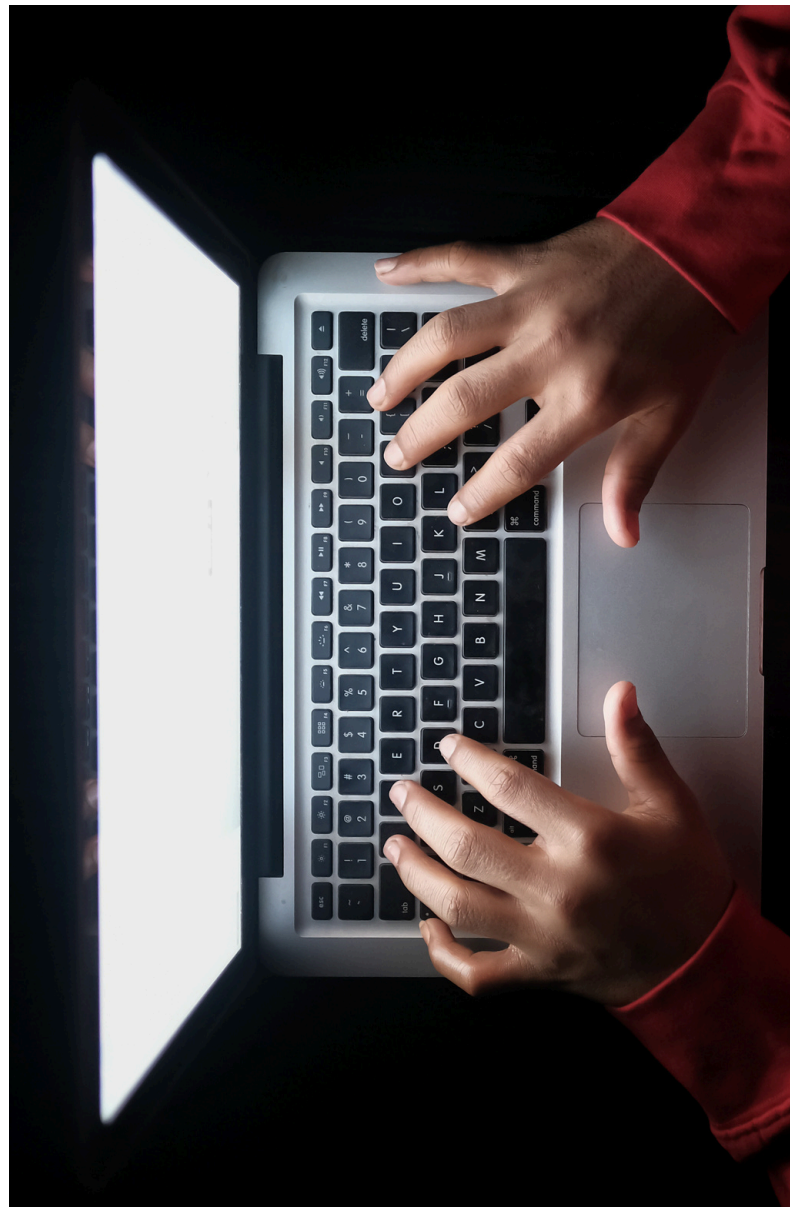
Om Fadhil

Managing Editor

Kang Ali

Contributors

Bang Javi
Aa Farhan
Bang Jagh
Om Fadhil
Kang Ali
Mas Fajar
Mas Oki



Dilema Kejujuran Red Team: Haruskah Membuka Semua Kartu Demi "Purple Teaming"?

BY OM FADHIL

Purple Teaming sering disalahpahami sebagai ajang keterbukaan total. Banyak organisasi percaya semakin detail rencana Red Team diumbar sejak awal, semakin sukses kolaborasinya. Kenyataannya, transparansi taktis yang terlalu dini justru membunuh realisme dan validitas pengujian.

Belajar dari Kesalahan: Bahaya "Simulasi yang Nyaman"

Dalam sebuah simulasi ransomware di industri perbankan, Red Team membeberkan seluruh target dan teknik serangan saat kick-off demi terlihat kolaboratif di mata Blue Team. Hasilnya? Blue Team sukses membuat aturan deteksi spesifik, dan laporan akhir membanggakan kematangan Security Operations Center (SOC) mereka.

Namun, dua bulan kemudian, SOC gagal total mendeteksi varian serangan serupa di lingkungan produksi. Mengapa? Karena mereka hanya membangun kepatuhan terhadap skenario, bukan kapabilitas deteksi yang adaptif. Inilah bahaya transparansi berlebih: menciptakan rasa aman palsu (false confidence).



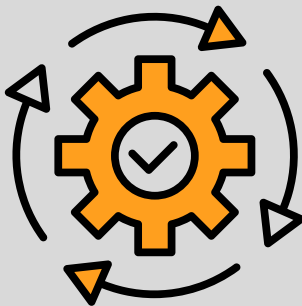
Membedah Siklus Purple Teaming: Kapan Tabir Harus Dibuka?

Jawaban atas "Apakah Red Team wajib transparan?" bukanlah ya atau tidak, melainkan soal apa dan kapan. Agar tidak merusak esensi keamanan, tingkat keterbukaan ini harus dipetakan ke dalam tiga fase siklus Purple Teaming:



Fase Perencanaan Kolaboratif (Joint Planning & Scoping)

Ini adalah fase dengan **transparansi strategis tingkat tinggi**. Red Team, Blue Team, dan manajemen menyepakati Rules of Engagement (RoE), batasan sistem (sejalan dengan NIST SP 800-115), dan tujuan utama simulasi (misalnya: menguji deteksi lateral movement). Transparansi di awal ini wajib agar operasional bisnis tidak terganggu.



Fase Eksekusi (Threat Emulation / Blind Execution)

Ini adalah fase kotak hitam. Saat Red Team mulai menyimulasikan serangan, detail taktis seperti **tool, payload, atau jalur masuk (attack path) wajib** dirahasiakan. Deteksi Blue Team harus bekerja berdasarkan insting dan konfigurasi nyata, tanpa bocoran. Jika tim bertahan sudah tahu persis urutan serangannya, mereka sekadar bermain sandiwara, bukan menguji kapabilitas.



Fase Evaluasi Bersama (Collaborative Debrief & Tuning)

Setelah target eksekusi tercapai, barulah transparansi penuh wajib diberikan. Di sesi inilah "warna ungu" itu tercipta. Red Team membeberkan seluruh taktik historis mereka secara gamblang (sering dipetakan dengan MITRE ATT&CK), dan Blue Team membedah log untuk menganalisis mengapa deteksi gagal atau berhasil. **Kolaborasi sejati terjadi di fase perbaikan ini**, bukan saat serangan sedang berlangsung.

Usecase Lokal: Ujian Realitas di Industri Strategis Indonesia

Di Indonesia, inisiatif Purple Teaming sering terbentur keharusan "lulus audit" demi laporan manajemen yang memuaskan. Mari bandingkan dua pendekatannya,

Sektor Perbankan/Fintech: Pengujian Akses Ilegal Admin

Kondisi: Bank digital menguji SOC terhadap ancaman pencurian kredensial admin (Privileged Access).



Pendekatan Terlalu Transparan: Pada fase perencanaan, Red Team membocorkan waktu dan titik masuk serangan secara spesifik. Saat dieksekusi, Blue Team berhasil memblokir seketika. Laporan: "Aman". Padahal, SOC berpotensi kebobolan jika serangan asli terjadi di luar "jadwal" yang dibocorkan.



Pendekatan Transparansi Bertahap: Red Team hanya membagikan niat strategisnya di fase perencanaan. Saat eksekusi berjalan diam-diam, Blue Team telat menyadari anomali. Namun, saat masuk fase evaluasi bersama (debrief), ditemukan fakta krusial: log dari server cabang ternyata tidak terkirim secara real-time ke sistem pusat (SIEM). Temuan operasional ini jauh lebih berharga daripada kebanggaan semu menangkap penyerang.

Sektor Manufaktur/BUMN: Pengujian Ransomware pada Sistem ERP

Kondisi: Pengujian Purple Teaming terhadap ancaman Ransomware pada sistem supply chain.



Penerapan: Red Team transparan soal aturan main di awal agar tidak mematikan mesin operasional (OT/ICS). Namun, saat masuk fase eksekusi, mereka merahasiakan pintu masuk utamanya (Initial Access Vector)—entah lewat phishing atau kerentanan VPN. Akibat minimnya informasi taktis ini, Blue Team dipaksa bersiaga mengasah insting dan memperkuat pertahanan di seluruh lini, bukan hanya menjaga satu pintu yang sudah dibocorkan.

KESIMPULAN



Tantangan terbesar penerapan Purple Teaming di Indonesia adalah mengubah pola pikir dari "ujian untuk lulus audit" menjadi "ujian untuk ketangguhan nyata". Transparansi ibarat mengatur suhu air: **hangat di fase perencanaan** untuk menjaga keamanan operasional, **dingin di fase eksekusi** demi realisme ancaman, dan **panas di fase evaluasi** untuk kolaborasi perbaikan. Inilah kunci agar investasi keamanan siber tidak berujung sebagai formalitas belaka.

CSIRT Tanpa Ujian: Mengapa Banyak Tim Gagal Saat Krisis Siber Terjadi

PELAJARAN DARI KEGAGALAN RESPONS INSIDEN DAN PENTINGNYA CYBER DRILL

BY BANG JAVI

Ketika Krisis Datang Tanpa Peringatan

Pukul 02.13 dini hari, sistem monitoring mendeteksi anomali. Beberapa server produksi tidak dapat diakses. Dalam waktu singkat, file mulai terenkripsi. Dashboard berubah merah.

Tim keamanan menerima alert. IT operasional mencoba restart sistem. Manajemen mulai menerima notifikasi. Grup komunikasi darurat dibuat.

Namun satu hal segera terlihat: tidak ada yang benar-benar memimpin.

CSIRT memang ada di struktur organisasi.

Incident Response Plan tersedia dalam bentuk dokumen. Namun tim tersebut belum pernah menjalani simulasi krisis bersama.

Dalam beberapa jam, bukan hanya sistem yang lumpuh koordinasi pun runtuh.



Di Mana Letak Kegagalannya?

Dalam banyak kasus, kegagalan bukan disebabkan oleh kurangnya teknologi. Firewall berjalan. Endpoint protection aktif. Log terkumpul.

Masalah muncul pada aspek yang jarang diuji:

- Siapa yang memiliki otoritas pengambilan keputusan?
- Siapa yang menghubungi manajemen?
- Kapan legal dan PR dilibatkan?
- Apakah containment dilakukan sebelum komunikasi publik?
- Apakah eskalasi berjalan sesuai prosedur?

Di atas kertas, semua tampak jelas, Dalam tekanan nyata, semuanya terasa berbeda.

Organisasi sering kali percaya bahwa memiliki dokumen berarti siap. Padahal kesiapan lahir dari latihan, bukan dari struktur.

Sebenarnya Apa Itu CSIRT?

Computer Security Incident Response Team (CSIRT) bukan sekadar tim teknis yang menangani malware atau serangan jaringan.

Secara strategis, CSIRT adalah mekanisme koordinasi organisasi saat krisis digital terjadi. Tanggung jawabnya mencakup:

- Identifikasi dan klasifikasi insiden
- Koordinasi containment dan eradikasi
- Pengelolaan komunikasi internal dan eksternal
- Dukungan pengambilan keputusan manajemen
- Evaluasi dan pembelajaran pasca insiden



Secara global, konsep dan praktik CSIRT juga dirujuk oleh komunitas internasional seperti FIRST, yang menjadi wadah kolaborasi tim respons insiden dari berbagai sektor dan negara.

Namun memiliki struktur CSIRT tidak otomatis berarti memiliki kapabilitas respons yang matang.

CSIRT bukan hanya tentang siapa yang duduk di dalam tim.

CSIRT adalah tentang bagaimana organisasi bergerak sebagai satu kesatuan saat krisis terjadi.

Dan kemampuan tersebut tidak pernah terbentuk tanpa pengujian.

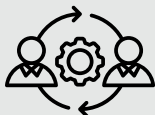
Mengapa CSIRT Gagal Ketika Tidak Pernah Diuji?

Ada tiga pola umum yang sering muncul:



Kepercayaan Diri Semu

Dokumen rapi, flowchart jelas, dan daftar kontak tersedia. Namun tidak pernah diuji dalam simulasi realistis.



Kurangnya Koordinasi Lintas Fungsi

CSIRT bukan hanya tim IT. Ia melibatkan manajemen, legal, komunikasi, dan operasional bisnis. Tanpa latihan bersama, koordinasi sering terhambat.



Tidak Terbiasa dengan Tekanan

Krisis siber bukan hanya persoalan teknis. Ada tekanan reputasi, tekanan regulator, dan tekanan waktu. Tanpa drill, organisasi tidak terbiasa mengambil keputusan di bawah tekanan tersebut.

Peran Red Team dan Cyber Drill dalam Menguji CSIRT

Di sinilah integrasi menjadi krusial.

- Red Team mensimulasikan pola serangan realistis.
- * Cyber Drill menguji bagaimana CSIRT merespons skenario tersebut.
- Purple Team membantu memperbaiki kelemahan deteksi dan respons yang ditemukan.
- ↻ Continuous Testing memastikan peningkatan tersebut tidak berhenti pada satu latihan saja.



Tanpa integrasi ini, Red Team hanya menghasilkan laporan teknis. Dengan integrasi ini, organisasi menguji ketahanan operasionalnya secara menyeluruh.

Cyber Drill yang melibatkan Red Team memindahkan organisasi dari sekadar “mengetahui risiko” menjadi benar-benar “menghadapi risiko” dalam lingkungan yang terkendali.

Dari Rencana ke Kesiapan Nyata

Banyak organisasi merasa aman karena telah membentuk CSIRT. Namun krisis siber tidak menguji niat baik atau struktur organisasi. Ia menguji kesiapan nyata.

Dalam situasi darurat, organisasi tidak akan naik ke level rencana terbaiknya. Organisasi akan turun ke level latihan terakhirnya.

Karena itu, pertanyaannya bukan lagi:
“Apakah kita memiliki CSIRT?”

Melainkan:
“Apakah CSIRT kita pernah benar-benar diuji?”

Paradoks SOC: Saat yang Paling Efektif Justru Paling Sunyi

Ada satu paradoks yang jarang dibicarakan secara jujur di dunia Security Operations Center (SOC): semakin rapi SOC bekerja, semakin susah ia “kelihatan bekerja”.

Di banyak organisasi, nilai SOC terasa nyata ketika layar dashboard ramai. Alert berderet. Grafik naik. Laporan insiden tebal. Kadang muncul temuan dengan istilah teknis yang terdengar meyakinkan. Aktivitas yang terlihat dianggap bukti kerja.

Sebaliknya, ketika SOC sedang relatif tenang—alert sedikit, tidak ada insiden besar, tidak ada temuan yang “wah”—muncul pertanyaan yang sering tidak diucapkan, tapi jelas terasa:

“SOC ini sebenarnya ngapain?”

Pertanyaan ini manusiawi. Tapi di keamanan siber, ia bisa menyesatkan.

“Di SOC, yang paling berharga sering kali bukan yang terjadi, tapi yang berhasil dicegah.”

BY BANG JAGH



Mengukur yang Kelihatan

Kita terbiasa menilai sesuatu dari apa yang bisa ditunjukkan. Dashboard penuh berarti ada kerja. Grafik naik berarti ada aktivitas. Laporan panjang berarti ada kontribusi.

Di banyak bidang, pola pikir ini benar. Namun dalam keamanan, logikanya sering kebalik. SOC bukan tim produksi. Ia bukan pabrik yang output-nya bisa dihitung per unit. SOC lebih mirip sistem peringatan dini dan pengendali risiko. Ia bekerja supaya gangguan tidak sempat membesar, supaya kerusakan tidak sempat meluas, supaya masalah berhenti sebelum jadi krisis.

Sayangnya, “tidak terjadi apa-apa” memang sulit dijadikan bahan presentasi.



Mengukur yang Kelihatan

Kita terbiasa menilai sesuatu dari apa yang bisa ditunjukkan, Dashboard penuh berarti ada kerja. Grafik naik berarti ada aktivitas. Laporan panjang berarti ada kontribusi.

Di banyak bidang, pola pikir ini benar. Namun dalam keamanan, logikanya sering kebalik. SOC bukan tim produksi. Ia bukan pabrik yang output-nya bisa dihitung per unit. SOC lebih mirip sistem peringatan dini dan pengendali risiko. Ia bekerja supaya gangguan tidak sempat membesar, supaya kerusakan tidak sempat meluas, supaya masalah berhenti sebelum jadi krisis.

Sayangnya, **“tidak terjadi apa-apa”** memang sulit dijadikan bahan presentasi.

Ancaman yang Tidak Dramatis, Tapi Paling Sering Menggigit

Di luar sana, keamanan siber sering dikemas seperti film: serangan besar, aktor negara, zero-day, malware bernama rumit. Menegangkan, layak judul besar.

Tapi dapur SOC sehari-hari biasanya jauh lebih sederhana—dan jujur saja, lebih membosankan:

- akun lama yang masih aktif,
- kata sandi dipakai ulang,
- layanan terbuka ke internet tanpa alasan jelas,
- hak akses kelewat longgar,
- patch tertunda berbulan-bulan.

Tidak ada dramanya. Tidak ada “plot twist”. Tapi justru pola-pola inilah yang paling sering membuka jalan masuk. Karena yang membuat insiden membesar bukan selalu teknik yang “baru”, melainkan kebiasaan lama yang tidak dibersihkan.

Ironisnya, karena terlalu sering terjadi, masalah seperti ini dianggap “biasa”. Tidak menarik. Tidak urgent. Ditunda lagi dan lagi—sampai suatu hari, yang “biasa” itu jadi penyebab downtime, kebocoran, atau kerusakan yang mahal.

Ketenangan yang Disalahartikan

Di titik ini, paradoks SOC terasa jelas.

Banyak organisasi menilai SOC dari apa yang berhasil ia temukan, bukan dari apa yang berhasil ia cegah. Akibatnya, ketika insiden tidak muncul, ketenangan dibaca sebagai “tidak ada kerja”.

Padahal kita tidak memakai cara pikir seperti itu di sistem keselamatan lain.

- Kota yang aman tidak dicurigai karena kriminalitasnya rendah.
- Pesawat tidak dianggap bagus karena sering hampir jatuh.
- Sistem pemadam kebakaran tidak diuji dengan cara membakar gedung.

Namun dalam keamanan siber, ketenangan sering memunculkan rasa curiga: “kalau sepi, jangan-jangan SOC-nya tidur.”

Tuntutan untuk Selalu ‘Ada yang Baru’

Tekanan berikutnya datang lewat kebutuhan akan hal baru.

Teknik baru. Pola baru. Taktik baru.

Seolah-olah nilai SOC bergantung pada kebaruan temuannya. Padahal kenyataannya, masalah yang paling sering berdampak besar masih itu-itu saja: kredensial disalahgunakan, konfigurasi keliru, akses kebablasan, kedisiplinan dasar yang longgar.

Tidak menemukan “TTP baru” bukan berarti tidak ada pekerjaan. Bisa jadi artinya fondasi sudah lumayan rapi: akses lebih tertata, konfigurasi lebih stabil, kebocoran kecil cepat ditutup, dan celah tidak sempat berkembang jadi pola serangan yang lebih rumit.

Salah Ukur: Output vs Dampak

Masalah utamanya sering bukan SOC-nya, melainkan cara organisasi mendefinisikan “kinerja”. SOC sering diukur dari output: berapa alert, berapa tiket, berapa insiden, berapa temuan.

Padahal yang paling penting adalah dampak: berapa banyak eskalasi yang tidak jadi terjadi, berapa banyak akses berisiko yang gagal dimanfaatkan, berapa banyak gangguan yang berhenti di tahap awal.

Dampak ini tidak selalu punya grafik dramatis. Tidak punya momen heroik. Tidak viral. Tapi justru itulah inti keamanan: mencegah sebelum menjadi berita.

“KEAMANAN YANG MATANG SERING TERASA MEMBOSANKAN—DAN ITU PERTANDA BAIK.”

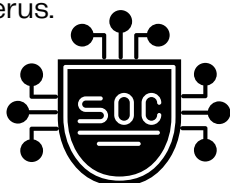


Keamanan yang Efektif Memang 'Biasa Saja'

Keamanan yang efektif sering hadir dalam bentuk yang sederhana, repetitif, dan terlihat “monoton”: konfigurasi yang rapi, akses yang tertib, log yang lengkap, keputusan yang konsisten, rutinitas operasional yang disiplin.

Tidak selalu ada adegan besar. Tidak selalu ada musuh yang dramatis. Justru karena sistemnya rapi, insiden tidak sempat naik panggung.

Masalahnya, sesuatu yang terasa “biasa saja” jarang dirayakan. Padahal di keamanan siber, “biasa saja” itu sering hasil kerja yang paling mahal: ketekunan, konsistensi, dan kebiasaan baik yang dilakukan terus-menerus.



Ketika SOC Dipaksa 'Menemukan Sesuatu'

Ada konsekuensi yang jarang kita sadari: ketika SOC hanya dianggap bekerja jika ada temuan besar, organisasi tanpa sadar menciptakan insentif yang salah.

Noise dibiarkan meningkat supaya dashboard ramai. Severity jadi “kosmetik” supaya terlihat penting. Laporan makin tebal, tapi inti masalahnya tidak makin jelas. Yang harusnya jadi perbaikan kontrol malah berubah jadi produksi aktivitas.

Ujungnya berbahaya: kegagalan kontrol bisa terlihat seperti keberhasilan operasional. Padahal SOC yang sehat seharusnya mendorong sistem makin rapi, bukan makin ribut.



Penutup: SOC yang Dewasa Tidak Mengejar Keramaian

Paradoks SOC itu nyata: semakin SOC membantu organisasi menjadi aman, semakin ia terlihat sunyi. Dan sunyi ini sering disalahartikan sebagai “tidak bekerja”.

Mungkin sudah saatnya kita mengubah cara pandang. Nilai SOC bukan pada seberapa ramai dashboard, melainkan seberapa stabil operasi. Bukan pada seberapa panjang laporan, melainkan seberapa cepat risiko dipotong di akar. Bukan pada seberapa sering kita panik, melainkan seberapa jarang kita perlu panik.

Pada akhirnya, SOC yang dewasa tidak mengejar keramaian. Ia mengejar satu hal yang sederhana namun paling sulit dijelaskan: membuat ancaman gagal menjadi insiden.

DevSecOps: Otomatisasi Vulnerability Scanning dengan Droid LLM Hunter

BY AA FARHAN

Dalam ekosistem pengembangan perangkat lunak modern, kecepatan rilis menjadi prioritas utama. Tim pengembang dituntut untuk merilis pembaruan aplikasi secara cepat dan berkelanjutan melalui pipeline CI/CD (Continuous Integration / Continuous Deployment). Namun, kecepatan ini sering kali mengorbankan aspek keamanan.

Pada konteks Android, setiap kali kode baru di-push ke repository, terdapat risiko munculnya kerentanan baru seperti SQL Injection, Hardcoded Secrets, Insecure WebView, hingga kerentanan logika yang kompleks. Pengujian keamanan manual yang dilakukan hanya secara periodik tidak lagi memadai untuk menghadapi tantangan ini.

Di sinilah konsep Continuous Testing hadir. Dengan mengotomatiskan proses vulnerability scanning sebagai bagian integral dari pipeline CI/CD, tim keamanan dapat mendeteksi dan merespons ancaman secara real-time di setiap perubahan kode tanpa harus menginstal tools apapun secara lokal.

Droid LLM Hunter (DLH) telah dikemas sebagai reusable GitHub Action, artinya repository Android manapun — baik aplikasi baru maupun yang sudah berjalan — cukup menambahkan satu file workflow YAML untuk langsung mendapatkan kemampuan AI-powered vulnerability scanning di setiap siklus pengembangan mereka.

Konsep Dasar: CI/CD & Continuous Testing Keamanan

Apa itu CI/CD?

CI/CD adalah praktik pengembangan perangkat lunak yang mengotomatiskan proses integrasi kode (Continuous Integration) dan pengiriman/deployment ke environment target (Continuous Deployment/Delivery). Setiap perubahan kode yang di-push ke repository akan secara otomatis melewati serangkaian proses: build, test, analisis, hingga deployment.



TAHAP	DESKRIPSI	CONTOH EVENT TRIGGER
Continuous Integration (CI)	Otomatisasi build dan test setiap ada perubahan kode	git push, open Pull Request
Continuous Delivery (CD)	Rilis ke staging/production secara otomatis setelah CI lulus	Merge ke branch main
Continuous Testing	Pengujian (termasuk keamanan) berjalan di setiap tahap pipeline	Semua event di atas

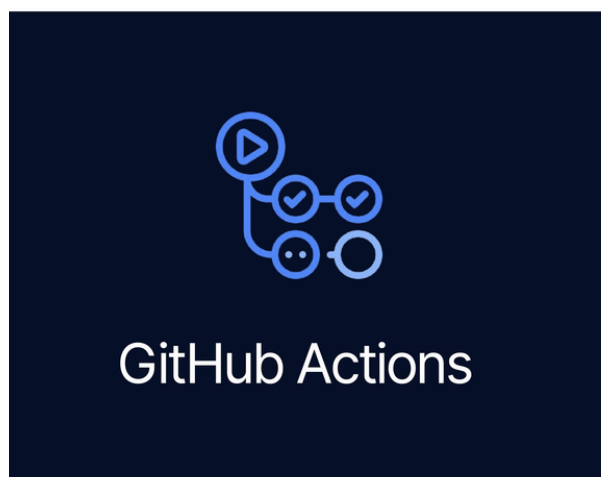
Continuous Testing dalam Konteks Keamanan

Dalam konteks keamanan siber, Continuous Testing berarti vulnerability scanning tidak lagi dilakukan sekali setahun atau sebelum rilis saja, melainkan berjalan otomatis di setiap perubahan kode. Pendekatan ini adalah inti dari paradigma DevSecOps dan prinsip shift-left security.

- Setiap push kode memicu pipeline yang menjalankan security scanner secara otomatis
- Pull Request tidak dapat di-merge jika ditemukan kerentanan di atas threshold yang ditentukan
- Tim security mendapat laporan real-time tanpa harus menunggu jadwal audit
- Biaya perbaikan jauh lebih murah karena kerentanan ditemukan lebih awal dalam siklus pengembangan

Mengapa GitHub Actions?

GitHub Actions adalah platform CI/CD native dari GitHub yang memungkinkan otomatisasi workflow langsung di dalam repository. Ia mendukung konsep reusable actions, di mana tool pihak ketiga seperti DLH dapat dikemas dan digunakan hanya dengan satu baris referensi di workflow YAML — tanpa perlu menginstal apapun.



Mengenal Droid LLM Hunter



Droid LLM Hunter (DLH) adalah tool analisis keamanan otomatis berbasis AI yang dirancang untuk mendeteksi kerentanan dalam aplikasi Android (.APK). Tool ini mengombinasikan Static Analysis (SAST) tradisional dengan kecerdasan kontekstual Large Language Models (LLMs).

Identitas Tool

 Repository : <https://github.com/roomkangali/droid-llm-hunter>

 Lisensi : MIT Open Source

 Versi Terbaru : v1.1.5 (Februari 2026)

 LLM Providers : Ollama, Gemini, OpenAI, Anthropic, Groq, OpenRouter

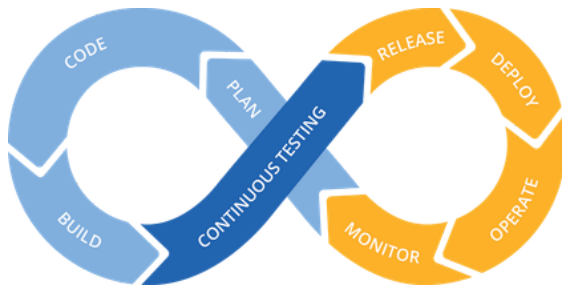
Vulnerability Rules yang Didukung (25+)

Kategori	Rules
Injection	sql_injection, graphql_injection, fragment_injection
WebView	webview_xss, insecure_webview, webview_deeplink, webview_file_access
Secrets & Storage	hardcoded_secrets, hardcoded_secrets_xml, insecure_storage, insecure_file_permissions
Kriptografi	insecure_random_number_generation, insecure_deserialization
Komponen Android	exported_components, intent_spoofing, pending_intent_hijacking, strandhogg
Deep Link	deeplink_hijack, deeplink_logic_bypass
Lanjutan	path_traversal, zip_slip, unsafe_reflection, biometric_bypass, jetpack_compose_security, universal_logic_flaw

Integrasi Droid LLM Hunter ke CI/CD Pipeline

Konsep: Tanpa Instalasi, Langsung Pakai

Karena DLH telah dikemas sebagai reusable GitHub Action, setiap repository Android cukup membuat file workflow YAML di folder `.github/workflows/`. GitHub Actions akan otomatis mengunduh dan menjalankan DLH di cloud runner milik GitHub — tidak ada yang perlu diinstal di mesin lokal developer maupun server.



⚡ Cara Kerja Integrasi

1. Developer push kode atau buka Pull Request
2. GitHub Actions otomatis ter-trigger
3. Runner mengunduh DLH dari GitHub (uses: `roomkangali/droid-llm-hunter@Development`)
4. DLH melakukan AI-powered vulnerability scan terhadap APK
5. Laporan JSON ter-upload sebagai artifact dan dapat diunduh

Kapan Pipeline Scan Berjalan?

DLH dapat dikonfigurasi untuk merespons berbagai event dalam siklus pengembangan. Ada empat momen utama yang paling umum digunakan:

- **Push ke branch** — Setiap kali developer mengirimkan kode ke branch manapun (develop, main, release, bahkan feature branch), pipeline scan langsung berjalan. Ini memastikan tidak ada kode baru yang masuk tanpa melewati pemeriksaan keamanan.
- **Pull Request ke branch develop** — Saat developer mengajukan permohonan untuk menggabungkan kode ke branch develop, scan dijalankan secara otomatis. Jika ditemukan kerentanan, tim dapat meninjau dan meminta perbaikan sebelum kode bergabung.
- **Pull Request ke branch main** — Ini adalah gate paling kritis. Sebelum kode masuk ke branch production, seluruh perubahan wajib lulus security scan. Pipeline bahkan dapat dikonfigurasi untuk memblokir proses merge secara otomatis jika ditemukan kerentanan kategori HIGH atau CRITICAL.
- **Scheduled scan** — Terlepas dari aktivitas developer, scan rutin dapat dijadwalkan secara berkala, misalnya setiap minggu, untuk mendeteksi kerentanan baru yang mungkin muncul dari pembaruan dependensi atau perubahan environment.

Fleksibilitas Konfigurasi

Integrasi DLH bersifat fleksibel karena tim dapat menyesuaikan perilaku scanner melalui konfigurasi workflow tanpa mengubah source code DLH. Misalnya, branch develop dapat diatur untuk melakukan scan cepat dengan rules terbatas, sementara branch main menjalankan scan menyeluruh dengan seluruh rules aktif dan analisis lebih mendalam. Rules kerentanan seperti SQL Injection, Hardcoded Secrets, Insecure WebView, hingga kerentanan logika kompleks juga dapat ditentukan langsung pada file konfigurasi, sementara API Key LLM disimpan aman melalui GitHub Secrets agar tidak terekspos di repository.

Setelah pipeline berjalan, tim akan memperoleh laporan keamanan dalam format JSON yang berisi detail temuan seperti jenis kerentanan, tingkat keparahan, confidence score, lokasi file dan baris kode, evidence, serta pemetaan ke standar OWASP MASVS. Laporan ini tersimpan sebagai artifact di GitHub Actions dan dapat diunduh selama periode retensi, serta pada konfigurasi dengan gate ketat, proses merge Pull Request akan otomatis diblokir jika ditemukan kerentanan CRITICAL untuk mencegah kode berbahaya masuk ke production.

Panduan Lengkap Integrasi CI/CD

Untuk contoh workflow siap pakai, konfigurasi lengkap, dan panduan langkah demi langkah mengintegrasikan DLH ke repository Android Anda, kunjungi:

<https://github.com/roomkangali/droid-llm-hunter/tree/Development/examples/github-actions>

Best Practices Implementasi CI/CD dengan DLH

Strategi Konfigurasi per Branch

Branch	Trigger	Filter Mode	Rules Aktif	Aksi jika Gagal
feature/**	Push	hybrid	Kerentanan utama (7-10 rules)	Warning saja
develop	Push + PR opened	hybrid	Kerentanan utama + medium risk	Blokir jika CRITICAL
main	PR opened + synchronize	llm_only	Semua rules aktif	Blokir jika HIGH/CRITICAL
Scheduled (main)	Cron mingguan	llm_only	Semua rules + exploit gen	Laporan ke tim security

Rekomendasi Pemilihan LLM Provider

- Groq (Gratis): Kecepatan tinggi, ideal untuk scan cepat di branch develop dan feature. Limit token perlu diperhatikan.
- Google Gemini Flash: Keseimbangan kecepatan dan akurasi yang baik. Direkomendasikan untuk penggunaan harian CI/CD.
- Google Gemini Pro / OpenAI GPT-4: Akurasi tertinggi, cocok untuk audit mingguan terjadwal atau scan branch main.
- Anthropic Claude: Analisis kontekstual mendalam, sangat baik untuk `universal_logic_flaw` dan kerentanan logika kompleks.
- Ollama (Lokal): Gratis dan privat, cocok untuk developer yang ingin scan lokal tanpa mengirim kode ke cloud.

Tips Keamanan Pipeline

- Selalu simpan API Key di GitHub Secrets, jangan pernah hardcode di YAML atau kode.
- Aktifkan Branch Protection Rule di branch main agar tidak bisa di-merge sebelum scan lulus.
- Gunakan retention-days: 30 untuk laporan artifact agar tidak menumpuk dan memakan storage.
- Tambahkan notifikasi Slack atau email jika pipeline gagal agar tim security langsung mengetahui.
- Lakukan review manual terhadap temuan DLH sebelum menutup issue — LLM bisa menghasilkan false positive.
- Pertimbangkan menggunakan VulnAppDLH sebagai benchmark berkala untuk memvalidasi efektivitas konfigurasi.

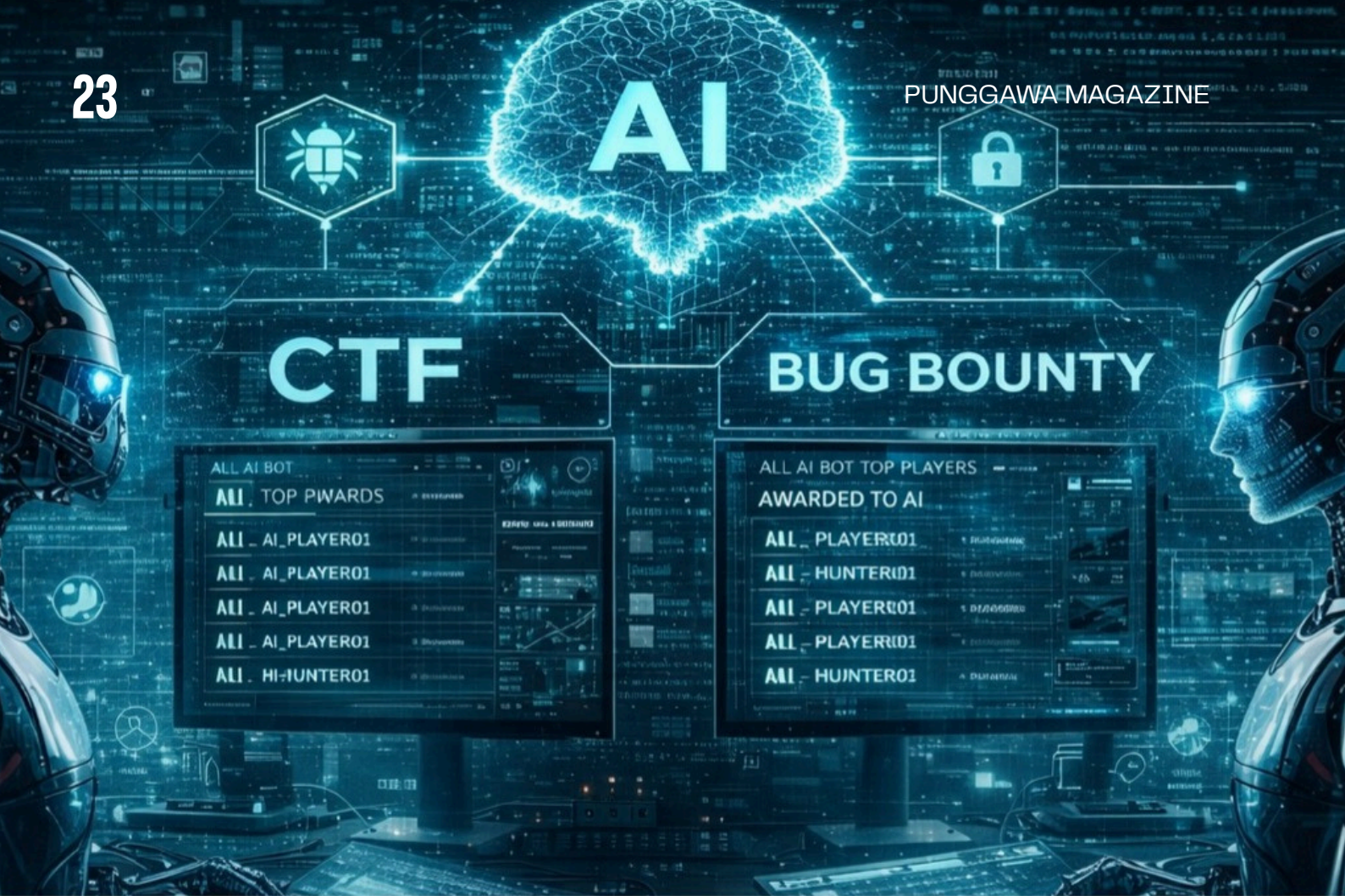
Kesimpulan

Integrasi Droid LLM Hunter ke pipeline CI/CD memungkinkan pemindaian kerentanan otomatis pada repository Android hanya dengan menambahkan satu file workflow YAML. Dengan trigger seperti push, Pull Request, atau scheduled scan, setiap perubahan kode akan diperiksa keamanannya, dan pipeline dapat otomatis gagal jika ditemukan kerentanan CRITICAL sehingga mencegah kode berbahaya masuk ke production. Pendekatan ini mendukung prinsip shift-left security dan implementasi praktik DevSecOps modern.

Poin Kunci

- ✓ DLH tidak perlu diinstal — cukup referensikan sebagai GitHub Action
- ✓ Trigger pada push, Pull Request (opened/synchronize), dan scheduled cron
- ✓ Rules dapat dikonfigurasi berbeda per branch (develop vs main)
- ✓ filter_mode: hybrid direkomendasikan untuk daily CI/CD
- ✓ Aktifkan Branch Protection Rule agar merge terkunci saat scan gagal
- ✓ Laporan JSON + artifact tersimpan otomatis di GitHub Actions





Evolusi CTF & Bug Bounty di Era AI

Ketika AI Mengambil Alih Puncak Leaderboard CTF dan Bug Bounty

BY MAS FAJAR

Kehadiran AI, khususnya AI generatif dan sistem yang semakin bersifat agentic, mulai mengubah dua arena utama yang selama ini menjadi tolok ukur kemampuan ofensif dan kualitas riset kerentanan, yaitu kompetisi Capture The Flag (CTF) dan program bug bounty.

Dalam ajang CTF, perubahan paling nyata bukan sekadar menyelesaikan soal dengan lebih cepat. Lebih dari itu, muncul kemungkinan baru berupa otomatisasi yang sanggup merencanakan, mengeksekusi, serta memverifikasi langkah-langkah, dan juga memecahkan masalah secara berantai. Akibatnya, batasan antara sekadar alat bantu operasional dan peran aktor utama menjadi semakin kabur.

Perkembangan ini sejalan dengan berbagai riset terbaru yang mencatat bahwa agen AI tertentu sudah mampu mendominasi ajang CTF dan menyelesaikan sebagian besar tantangan hanya dalam satu kali kompetisi.



Pertanyaan yang dulu terasa spekulatif kini menjadi sangat konkret: ketika mesin bisa menyelesaikan tantangan CTF dan menemukan bug secara otomatis, apa yang sebenarnya kita kompetisikan?

SATU NAMA MENGUBAH SEGALANYA

Ada sebuah momen pada Juni 2025 yang mengubah cara industri keamanan siber memandang dirinya sendiri.

Sebuah entitas bernama XBOW muncul di puncak leaderboard HackerOne, platform bug bounty terbesar di dunia, mengalahkan ribuan peneliti dari seluruh dunia. Yang membuatnya istimewa bukan seberapa cepat ia naik ke posisi itu, meski 90 hari memang waktu yang singkat. Yang membuatnya istimewa adalah bahwa XBOW bukan manusia. Ia adalah sistem AI otonom yang menemukan, memvalidasi, dan melaporkan kerentanan tanpa intervensi dari manusia.

Komunitas bereaksi dengan campuran kagum, cemas, dan skeptis. Tepat dua bulan setelah XBOW mencapai puncak global, HackerOne mengambil langkah yang belum pernah dilakukan sebelumnya: memisahkan leaderboard antara peneliti manusia dan AI-powered collectives.

Di arena CTF, arah perkembangan yang sama juga sedang terjadi meskipun lebih senyap. Sistem AI bernama CAI atau Cybersecurity AI tampil memukau dalam kompetisi berformat AI melawan manusia di platform Hack The Box. Sistem ini berhasil meraih posisi pertama di antara seluruh tim AI dan masuk ke jajaran dua puluh besar secara keseluruhan.

Sementara itu, pada CTFtime yang menjadi tolok ukur kompetisi CTF global, musim tahun 2026 sedang berlangsung dengan papan peringkat yang masih terus berubah. Tim-tim baru seperti krauq dan juga OxFUN dari Arab Saudi berhasil melesat ke posisi pertama dan kedua global hanya dalam dua bulan pertama, melampaui berbagai tim veteran yang sempat mendominasi dari tahun 2023 hingga 2025.





Cara Kerja XBOW: CTF Sebagai Titik Awal

XBOW dibangun oleh Oege de Moor, mantan kepala GitHub Next. Yang menarik, sistem ini tidak dilatih dari dataset vulnerability yang besar. Ia dilatih dari CTF.

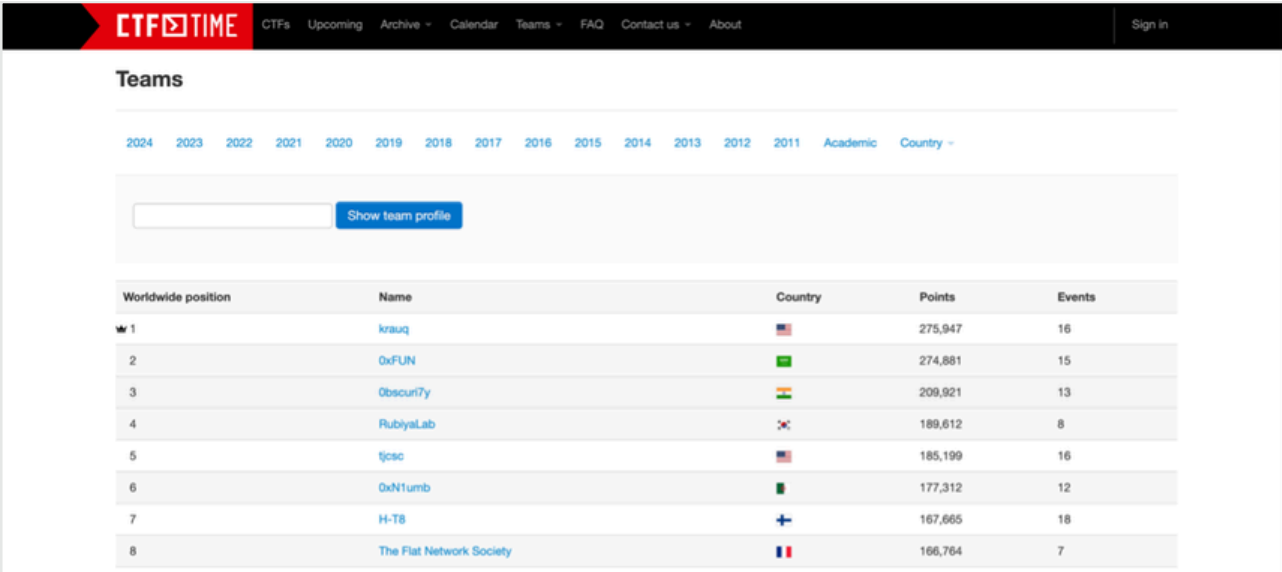
Tim XBOW menanam canaries di environment uji, lalu agen AI dilepas untuk menemukannya, persis mekanisme Capture The Flag. Dari sana, kapabilitas yang sama dibawa ke program bug bounty HackerOne secara black-box, tanpa pengetahuan internal, tanpa jalan pintas. Brendan Dolan-Gavitt dari XBOW menyebutnya langsung di Black Hat 2025:

Hasilnya dalam 90 hari pertama sungguh tidak main-main. XBOW berhasil mengirimkan sekitar 1.060 laporan temuan, dengan 54 di antaranya berstatus critical dan 242 berisiko high, serta lebih dari 130 celah telah berhasil diperbaiki oleh pemilik program. Pencapaian gemilang ini cukup untuk merebut posisi pertama di tingkat Amerika Serikat yang kemudian melesat ke peringkat satu global. Suntikan dana segar sebesar 75 juta dolar untuk pendanaan Seri B pun segera menyusul tidak lama setelah pengumuman tersebut.

Kunci keberhasilan XBOW bukan kecepatan pindai, melainkan validator sistem verifikasi deterministik yang mengonfirmasi setiap temuan sebelum dilaporkan. Ketika agen menemukan potensi XSS misalnya, headless browser langsung memverifikasi apakah JavaScript payload benar-benar dieksekusi. Hasilnya XBOW jauh lebih akurat dari scanner konvensional yang rentan false positive.

Krauq dan Dominasinya di Arena CTF

Krauq bukanlah sebuah tim maupun entitas korporat. Ia adalah satu orang yang bermain solo dan kebetulan juga membangun krauq.ai, sebuah AI CTF solver yang ia kembangkan sendiri serta tersedia gratis untuk publik. Ia menggunakan alat buatannya sendiri untuk berkompetisi dan berhasil menduduki peringkat pertama dunia per Februari 2026, mengalahkan ribuan tim dari seluruh penjuru dunia. Tepat di bawahnya ada OxFUN dari Arab Saudi yang menempati posisi kedua.



CTF TIME CTFs Upcoming Archive Calendar Teams FAQ Contact us About Sign in

Teams

2024 2023 2022 2021 2020 2019 2018 2017 2016 2015 2014 2013 2012 2011 Academic Country

Show team profile

Worldwide position	Name	Country	Points	Events
1	krauq		275,947	16
2	OxFUN		274,881	15
3	Obscur7y		209,921	13
4	RubiyaLab		189,612	8
5	ijsc		185,199	16
6	OxN1umb		177,312	12
7	H-TB		167,665	18
8	The Flat Network Society		166,764	7

Sementara itu, tim-tim veteran yang mendominasi selama tiga tahun terakhir, seperti Blue Water sebagai juara 2023 dengan 1.450 poin, kalmarunionen sebagai juara 2024 dengan 1.721 poin, dan r3kapig sebagai juara 2025 dengan 1.589 poin, posisinya sudah turun jauh. R3kapig kini bahkan berada di peringkat ketiga belas.

Hal ini bukanlah bukti bahwa AI telah mengambil alih sepenuhnya ajang CTF. Ini adalah bukti nyata bahwa satu orang yang mampu membangun dan memanfaatkan AI dengan benar bisa melampaui ratusan tim yang masih bermain menggunakan cara lama. Perbedaannya memang tipis namun sangat penting.

Data yang lebih eksplisit mengenai kapabilitas AI murni justru berasal dari riset kompetisi CAI. Framework agentic open-source ini berhasil menyelesaikan 19 dari 20 tantangan dalam ajang kompetisi AI melawan manusia di Hack The Box. Sistem ini bahkan sempat finis di posisi keenam secara keseluruhan pada tiga jam pertama, dengan tingkat efisiensi yang diklaim mencapai 3.600 kali lebih cepat dibandingkan garis dasarnya. Biaya inferensi yang tadinya mencapai ribuan dolar juga berhasil ditekan menjadi hanya ratusan dolar untuk setiap kompetisi.

Artinya, hambatan masuk untuk menggunakan AI dalam ajang CTF kini menjadi semakin rendah. Hal ini bukan lagi sekadar wilayah eksklusif untuk menguji kemampuan seseorang.

CAI (Cybersecurity AI) wins NeuroGrid CTF: The World's Top Cybersecurity AI Agent

CAI solved 41/45 adversarial challenges and led the global scoreboard. Powered by **alias1** and built for high-stakes defense tasks, CAI demonstrates state-of-the-art autonomy in real-world cybersecurity environments.

ALIAS ROBOTICS
Robot Cybersecurity

Di Mana Posisi Manusia Sekarang?

Melihat AI mendominasi leaderboard, apakah ini berarti profesi pentester dan bug bounty hunter akan punah? HackerOne menyebut ini sebagai "Bionic Hacker" peneliti yang menggunakan AI sebagai co-pilot, bukan menganggapnya sebagai pengganti.

Berdasarkan survei tahun 2025, 58% peneliti percaya bahwa AI masih kerap melewatkan kerentanan logika bisnis (business logic) dan exploit chaining yang tidak konvensional. Hanya 12% yang merasa yakin bahwa AI bisa menggantikan mereka sepenuhnya.

Hal ini sangat konsisten dengan batas teknis AI saat ini. Agen seperti XBOW memang sangat kuat pada kerentanan deterministik (seperti XSS, RCE, XXE, SSRF) yang memiliki pola input-output yang bisa diprediksi. Tapi kerentanan yang benar-benar berdampak tinggi—yang membutuhkan pemahaman konteks bisnis, rantai eksploitasi yang manipulatif, atau logika yang tidak pernah ada di training data mana pun masih menjadi domain eksklusif untuk manusia.

Proyeksi Dua Tahun ke Depan: AI vs AI

Jika hari ini kita melihat AI mengambil alih leaderboard, lalu bagaimana rupa lanskap ancaman siber 24 bulan dari sekarang? Dua tahun ke depan, kita akan menyaksikan eskalasi dari era Agentic AI menuju otonomi penuh.

Di arena CTF, AI tidak hanya akan menyelesaikan challenge, tetapi akan bertarung dalam format Attack/Defense secara real-time. Mesin akan meretas mesin lain, sambil di saat yang sama secara otonom melakukan patching (penambalan) pada kodenya sendiri dalam hitungan milidetik, sebuah skenario yang kini sedang didanai secara masif oleh inisiatif global seperti kompetisi DARPA AIXCC senilai puluhan juta dolar.

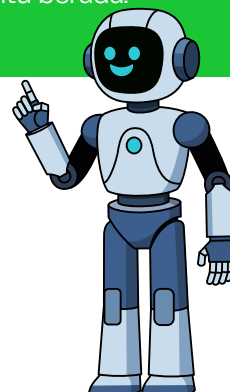
Sementara itu, di ekosistem bug bounty, kita akan menghadapi "hiper-inflasi" laporan kerentanan. Saat AI mampu memindai jutaan attack surface secara konstan, tantangan bagi perusahaan bukan lagi mencari celah, melainkan memilah kerentanan mana yang kritis dari ribuan laporan valid yang dikirim oleh AI peretas. Medan pertempuran akan berubah murni menjadi AI vs AI. Blue Team AI yang berusaha melakukan patch celah secara proaktif sebelum Red Team AI berhasil mengeksploitasinya.

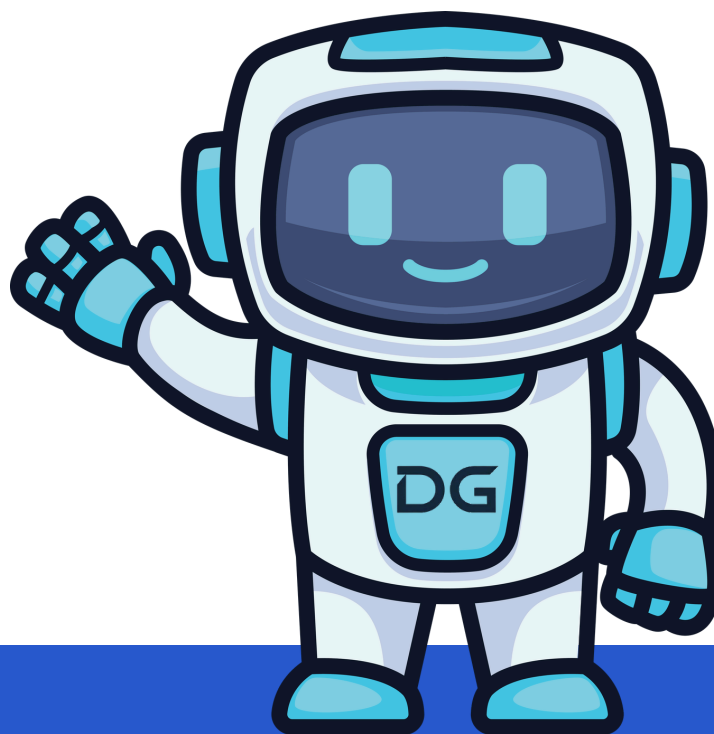
Tentu ini masih proyeksi, tapi sinyal-sinyal awalnya sudah bisa kita baca sekarang.

Kesimpulan

Bagi para hunter, profesional keamanan, dan peserta CTF, ini bukan lagi dinilai dari seberapa cepat menemukan pola kerentanan yang sudah dikenal. Kini terletak pada kemampuan untuk memahami mengapa sesuatu itu berbahaya dari kaca mata bisnis, dan membuktikannya dengan cara-cara manipulatif yang belum bisa diotomatisasi oleh AI.

Tahun ini menjadi saksi pergeseran takhta di ekosistem keamanan siber. Yang berubah sebenarnya bukanlah apakah manusia masih relevan atau tidak. Yang berubah adalah di mana letak nilai dari manusia itu berada.





DursGo

Web Vulnerability Scanner Berbasis Go dengan Analisis AI

BY KANG ALI

Di era modern keamanan aplikasi web, para peneliti keamanan dan bug hunter membutuhkan alat yang mampu melakukan analisis cepat terhadap permukaan serangan (attack surface). Banyak alat pemindaian kerentanan telah tersedia, mulai dari proxy manual hingga scanner otomatis. Namun, muncul pendekatan baru yang mencoba menggabungkan pemindaian kerentanan tradisional dengan kecerdasan buatan (AI).

Salah satu proyek yang menarik perhatian adalah DursGo, sebuah web vulnerability scanner yang dikembangkan menggunakan bahasa pemrograman Go. Tool ini dirancang untuk membantu peneliti keamanan dalam melakukan crawling, scanning, dan analisis hasil kerentanan secara otomatis dengan dukungan Large Language Model (LLM).

Apa Itu DursGo?

DursGo adalah automated web security scanner yang menggabungkan tiga komponen utama:

- Web crawler untuk menemukan halaman dan endpoint.
- Scanner modul kerentanan untuk menguji berbagai jenis vulnerability.
- AI analysis engine untuk menganalisis hasil temuan dan memberikan penjelasan teknis.

Berbeda dengan beberapa tool AI yang mencoba langsung mendeteksi vulnerability, DursGo menggunakan pendekatan yang lebih realistis: AI hanya digunakan untuk menganalisis hasil scanning, bukan sebagai mesin deteksi utama.

Pendekatan ini membantu mengurangi risiko false positive yang sering terjadi pada scanner berbasis AI.

Repository : <https://github.com/roomkangali/dursgo>



Install DursGo

```
git clone https://github.com/roomkangali/dursgo.git
```

```
cd dursgo
```

Build the Application:

```
go build -o dursgo ./cmd/dursgo
```

Copy to PATH (Optional):

```
sudo cp dursgo /usr/local/bin/
```

For JavaScript rendering or DOM XSS features, install Chrome/Chromium:

Ubuntu: `sudo apt-get install -y chromium-browser`

macOS: `brew install --cask google-chrome`

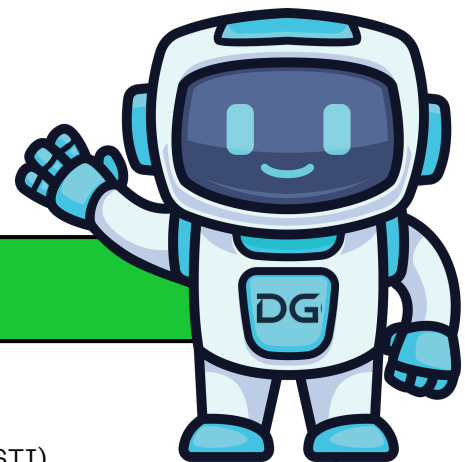
Arsitektur Kerja DursGo

Secara sederhana, proses kerja DursGo dapat digambarkan sebagai berikut:

- Target website dimasukkan ke dalam scanner.
- Crawler akan menjelajahi halaman web untuk menemukan endpoint dan parameter.
- Modul scanner melakukan pengujian terhadap berbagai jenis kerentanan.
- Hasil temuan dianalisis oleh AI untuk memberikan ringkasan dan rekomendasi perbaikan.

Alur ini memungkinkan proses scanning berjalan secara otomatis, namun tetap memberikan analisis yang mudah dipahami oleh manusia.

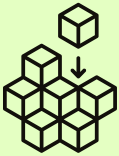
Vulnerability Rules yang Didukung



Kategori	Rules
Injection Vulnerabilities	- SQL Injection - Command Injection - Server-Side Template Injection (SSTI)
Client-Side Vulnerabilities	- Cross-Site Scripting (XSS) - DOM-based XSS - Open Redirect
Server Configuration Issues	- CORS misconfiguration - Missing security headers - Exposed configuration files
API Vulnerabilities	- Insecure Direct Object Reference (IDOR) - Broken Object Level Authorization (BOLA) - Mass assignment vulnerabilities - GraphQL security issues
Server Interaction Attacks	- Server-Side Request Forgery (SSRF) - Blind SSRF

Fitur Penting dalam DursGo

1. Modular Scanner Architecture



DursGo dirancang menggunakan pendekatan modular. Setiap jenis vulnerability memiliki modul scanner sendiri, sehingga pengembang dapat dengan mudah menambahkan atau memperbarui teknik scanning di masa depan. Pendekatan ini membuat tool menjadi fleksibel dan mudah dikembangkan.

2. Crawling dan Parameter Discovery

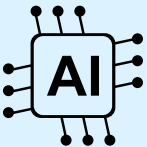


Salah satu tantangan utama dalam vulnerability scanning adalah menemukan semua endpoint dan parameter yang tersedia. DursGo menggunakan crawler untuk:

- menemukan halaman web
- mengekstrak parameter dari URL
- mengidentifikasi input yang berpotensi dieksploitasi

Hal ini membantu memperluas attack surface yang dianalisis oleh scanner.

3. Analisis Berbasis AI



Setelah scanner menemukan potensi kerentanan, DursGo dapat mengirim hasil temuan tersebut ke model AI untuk melakukan analisis tambahan.

AI kemudian akan memberikan:

- ringkasan kerentanan
- kemungkinan root cause
- rekomendasi mitigasi
- penjelasan teknis

Fitur ini sangat membantu bagi peneliti keamanan yang ingin mendapatkan insight cepat terhadap hasil scanning.

4. Dukungan JavaScript Rendering



Beberapa kerentanan modern, seperti DOM-based XSS, hanya muncul ketika JavaScript dijalankan di browser.

Untuk mengatasi hal ini, DursGo mendukung penggunaan headless browser untuk merender halaman web sebelum melakukan analisis.

Fitur ini memungkinkan scanner mendeteksi kerentanan yang tidak terlihat pada analisis HTML statis.

Kelebihan DursGo

Beberapa keunggulan utama dari tool ini antara lain:

- Dibangun menggunakan Go, sehingga memiliki performa tinggi dan concurrency yang baik
- Arsitektur modular yang memudahkan pengembangan
- Integrasi AI untuk analisis hasil vulnerability
- Dukungan crawling untuk memperluas attack surface

Pendekatan ini membuat DursGo cocok digunakan sebagai alat reconnaissance dan scanning awal dalam proses penetration testing.

Keterbatasan

Seperti tool automated lainnya, DursGo juga memiliki beberapa keterbatasan.

1. Proses scanning dapat memakan waktu cukup lama karena banyaknya kombinasi endpoint, parameter, dan payload yang diuji.
2. AI dalam DursGo tidak digunakan untuk menemukan vulnerability secara langsung, sehingga efektivitas scanner tetap bergantung pada kualitas modul scanning yang tersedia.
3. Beberapa fitur seperti DOM vulnerability scanning memerlukan headless browser tambahan.

Kesimpulan

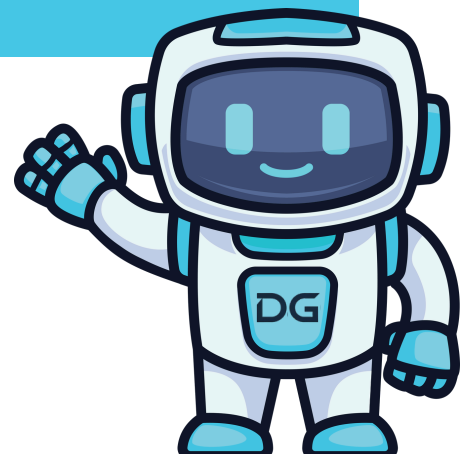
DursGo merupakan contoh menarik dari evolusi alat keamanan modern yang mencoba menggabungkan scanner tradisional dengan analisis berbasis AI. Dengan arsitektur modular dan dukungan berbagai jenis vulnerability scanning, tool ini dapat membantu peneliti keamanan dalam melakukan analisis awal terhadap aplikasi web.

Meskipun belum menggantikan kebutuhan pengujian manual oleh security researcher, DursGo dapat menjadi alat tambahan yang berguna dalam workflow penetration testing modern, terutama untuk tahap reconnaissance dan vulnerability discovery.

Seiring dengan berkembangnya teknologi AI dan teknik scanning otomatis, tool seperti DursGo kemungkinan akan terus berkembang dan memainkan peran penting dalam ekosistem keamanan aplikasi web.

Source

- <https://github.com/roomkangali/dursgo>
- <https://www.youtube.com/watch?v=n1YL1Kxg3Rc&list=PLqHz74gCLCXf2XhlAtqFYnIiQofu8RIIO7>



Burp Suite MCP Server

BY MAS OKI

PENDAHULUAN

Dalam beberapa tahun terakhir, kemampuan kecerdasan buatan (AI) meningkat sangat cepat, terutama pada model bahasa besar yang mampu memahami konteks, menjawab pertanyaan teknis, hingga membantu proses analisis keamanan. Di sisi lain, dunia cyber security dan penetration testing (pentest) selalu membutuhkan efisiensi, akurasi, serta kolaborasi lintas tools. Di titik pertemuan dua kebutuhan tersebut, muncul kebutuhan akan protokol yang dapat “menjembatani” model AI dengan berbagai sumber data dan alat kerja. Model Context Protocol (MCP) hadir sebagai solusi yang semakin relevan.

Artikel ini mengulas MCP dari sudut pandang cyber security, bagaimana MCP dapat diterapkan pada pentest dan manfaat yang ditawarkan.

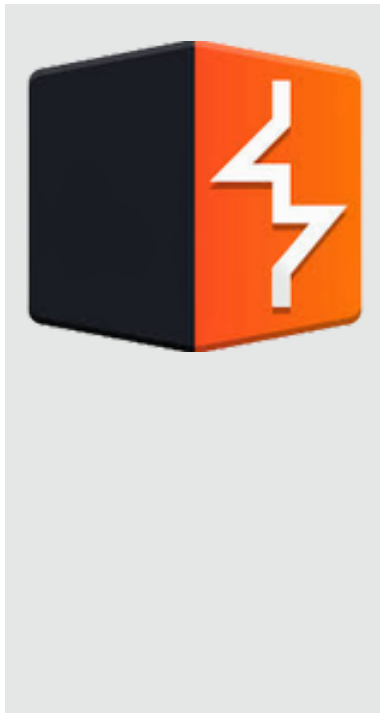




Apa Itu Model Context Protocol (MCP)

Model Context Protocol (MCP) adalah protokol terbuka yang dirancang untuk menghubungkan model AI dengan sumber konteks eksternal secara terstruktur. Konteks di sini dapat berupa data, sistem, tool, atau layanan yang dibutuhkan model agar bisa memberikan jawaban atau aksi yang lebih tepat.

Jika dianalogikan, MCP adalah “jembatan” yang memungkinkan model AI berkomunikasi dengan aplikasi lain tanpa perlu integrasi satu per satu secara kustom. Dengan MCP, model AI bisa “memahami” sumber data organisasi atau tool tertentu, karena protokolnya memberikan cara baku untuk meminta, menerima, dan menggunakan konteks.



Burp Suite MCP Server

Burp Suite MCP adalah integrasi Burp Suite dengan sistem AI melalui Model Context Protocol (MCP), di mana Burp berfungsi sebagai execution engine dan AI sebagai analisis & decision engine. Seluruh traffic HTTP yang ditangkap Burp termasuk request, response, header, parameter, cookie, dan status code dikirim sebagai konteks ke AI untuk dianalisis. Berdasarkan analisis tersebut, AI mengirimkan instruksi teknis yang kemudian dieksekusi kembali oleh Burp secara otomatis.

Pendekatan ini membentuk adaptive exploitation loop, yaitu siklus berkelanjutan antara analisis dan eksekusi, sehingga proses penetration testing menjadi lebih kontekstual dan presisi dibanding metode manual berbasis wordlist statis.

Mekanisme Kerja Burp Suite MCP

Pada bagian ini, penting untuk memahami bagaimana alur teknis Burp MCP berjalan. Integrasi ini bukan sekadar otomatisasi sederhana, tetapi sistem berbasis umpan balik (feedback-driven automation) yang terus menyesuaikan strategi serangan berdasarkan respons target.

Alurnya dapat dijabarkan sebagai berikut:

**Context Collection**

Burp menangkap dan menyusun konteks HTTP secara lengkap dari traffic yang melewati proxy.

Context Transmission

Data request dan response dikirim ke AI melalui MCP untuk dianalisis.

AI Analysis & Decision

AI mengidentifikasi potensi kerentanan, menentukan titik injeksi, dan memilih strategi eksploitasi.

Execution by Burp

Burp memodifikasi request sesuai instruksi AI dan mengirim ulang ke target.

Feedback Loop

Response terbaru dikirim kembali ke AI untuk evaluasi lanjutan atau chaining eksploitasi.

Burp MCP memberikan sejumlah kemampuan teknis yang memperluas fungsi tradisional Burp Suite. Fitur-fitur ini membuat proses pengujian menjadi lebih efisien dan kontekstual.

Fungsi utamanya meliputi:

Adaptive Payload Injection

Payload dihasilkan berdasarkan pola respons sebelumnya.

Dynamic Header Recalculation

Sistem dapat menghitung ulang HMAC, token, timestamp, atau nonce setiap kali request dimodifikasi.

Automated Replay & Comparison

Burp dapat mengirim ulang request secara otomatis dan membandingkan response untuk mendeteksi perbedaan signifikan.

Context-Based Exploit Validation

AI membantu memverifikasi apakah perubahan respons benar-benar menunjukkan keberhasilan eksploitasi atau hanya anomali biasa.

Burp Suite MCP with Claude Code (Local LLM)

Pada kesempatan kali ini kami akan melakukan implementasi Burp Suite MCP menggunakan Claude Code dengan Ollama.

Download Ollama

<https://ollama.com/download>

Setelah Ollama terinstal, Anda perlu menyiapkan model. Anda dapat menginstalnya dengan perintah `ollama run <model-name>`

Selanjutnya, install Claude Code

MacOS/Linux: `curl -fsSL https://claude.ai/install.sh | bash`

Windows: `irm https://claude.ai/install.ps1 | iex`

Selanjutnya anda perlu melakukan konfigurasi Claude Code agar terhubung dengan Ollama menggunakan Anthropic-compatible API.

```
export ANTHROPIC_AUTH_TOKEN=ollama
```

```
export ANTHROPIC_API_KEY=""
```

```
export ANTHROPIC_BASE_URL=http://localhost:11434
```

Selanjutnya jalankan claude

```
claude --model <available-model>
```

```
PS C:\Users\INTEL\test_ai> claude --model qwen3:8b

Claude Code v2.1.63
qwen3:8b · API Usage Billing
~\test_ai

/model to try Opus 4.6

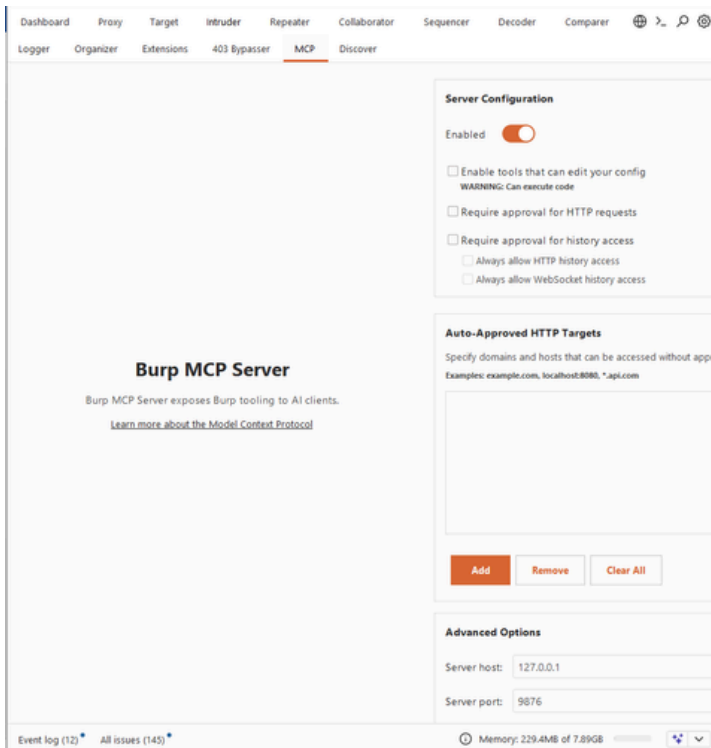
> Try "how does <filepath> work?"

? for shortcuts
```

Burp Suite MCP Server

Setelah Ollama dan Claude Code telah berhasil terinstall dan berjalan. Selanjutnya kita akan mengkonfigurasi MCP Server pada Burp Suite. Pertama, install ekstensi "MCP Server" <https://portswigger.net/bappstore/9952290f04ed4f628e624d0aa9dccebc>

Setelah ekstensi "MCP Server" berhasil diinstall, pastikan ekstensi tersebut berhasil berjalan

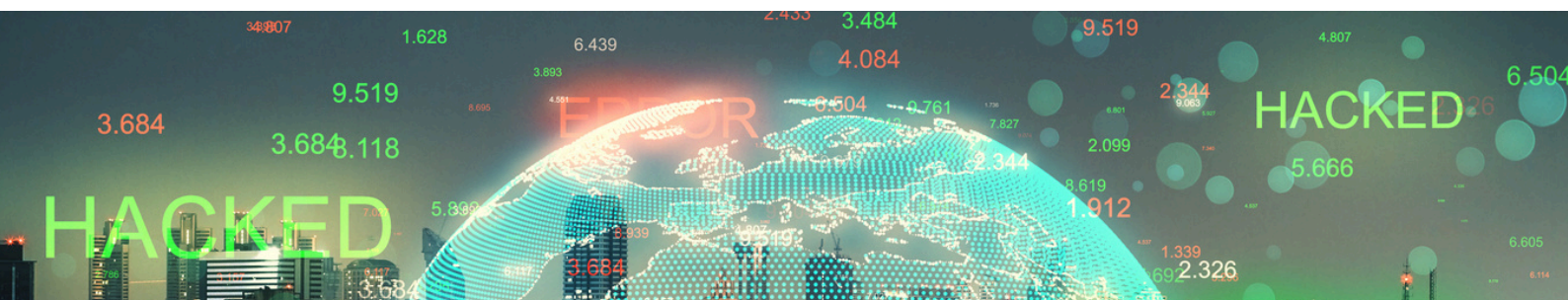


Sekarang kita perlu menambahkan konfigurasi Burp Suite MCP pada file config claude. Tambahkan baris berikut pada konfigurasi JSON Claude Code

```
{
  "mcpServers": {
    "burp": {
      "command": "<path to Java executable packaged with Burp>",
      "args": [
        "-jar",
        "/path/to/mcp/proxy/jar/mcp-proxy-all.jar",
        "--sse-url",
        "<your Burp MCP server URL configured in the extension>"
      ]
    }
  }
}
```

File mcp-proxy-all.jar didapatkan dengan extract server proxy jar pada ekstensi Burp Suite MCP Server.

Setelah proses instalasi dan konfigurasi selesai, Burp Suite MCP dapat digunakan.





Contoh prompt sederhana untuk melakukan Vulnerability Assessment

Request:

```
GET /listproducts.php?cat=1&27 HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1234
Server: Apache/2.4.18 (Ubuntu)
Date: Mon, 10 Oct 2023 12:34:56 GMT
Connection: close
```

Inspector:

Request query parameters: 2
Request body parameters: 0
Request cookies: 0
Request headers: 3

AI Assistant:

1. No SQL Injection Vulnerability: The application is likely using parameterized queries or input sanitization to prevent SQLi.

2. Error Suppression: The server is configured to hide error messages, making it difficult to detect vulnerabilities through traditional payloads.

Recommended Next Steps:

1. Use Burp Scanner: Run a full SQLi scan via Burp's automated scanner for comprehensive detection.
2. Check Response Headers: Look for anomalies in headers (e.g., unexpected content types, lengths, or server responses).
3. Manual Repeater Analysis: Use Burp's Repeater tab to manually adjust payloads and observe subtle changes in responses.

Would you like me to proceed with the Burp Scanner or further manual analysis?

using burp mcp, create HTTP request to http://testphp.vulnweb.com/listproducts.php?cat=1 and do SQL injection test by appending single quote character into value of cat parameter, if it's vuln then create repeater tab containing vulnerable request

burp - send_http_request (MCP)(url: http://testphp.vulnweb.com/listproducts.php?cat=1&' HTTP/1.1, host: testphp.vulnweb.com, user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36, accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8, targetPort: 80)

HttpRequestResponse(httpRequest=GET /listproducts.php?cat=1&' HTTP/1.1, host=testphp.vulnweb.com, user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36, accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8)

The SQL injection test triggered a syntax error response, confirming the vulnerability:

Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 1

This indicates the application is susceptible to SQL injection. I'll create a repeater tab for detailed analysis:

Repeater tab "SQLi Vulnerable Request" created with payload:

```
GET /listproducts.php?cat=1&' HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
```

You can now analyze the response in Burp Suite's Repeater tab. Would you like me to:

1. Try extracting database information
2. Test for time-based blind SQLi
3. Use Burp Scanner for automated analysis

Contoh prompt untuk mendapatkan proxy history berdasarkan kata kunci

Proxy history:

#	Host	Method	URL	Params	Status code	Length	MIME type	Extension	Title
77	https://www.google.com	GET	/complete/search?client=chrome...		200	4460	script		
76	http://testphp.vulnweb.com	GET	/AJAX/infoartist.php?id=1		200	1563	XML	php	
75	http://testphp.vulnweb.com	GET	/AJAX/artist.php		200	365	XML	php	
73	http://testphp.vulnweb.com	GET	/AJAX/index.php		200	4458	HTML	php	ajax test
72	http://testphp.vulnweb.com	GET	/artists.php		200	5550	HTML	php	artists
58	http://testphp.vulnweb.com	GET	/listproducts.php?cat=1		200	8102	HTML	php	pictures
55	http://testphp.vulnweb.com	GET	/artists.php		200	5550	HTML	php	artists
54	http://testphp.vulnweb.com	GET	/categories.php		200	6337	HTML	php	picture c
50	https://www.google.com	GET	/complete/search?client=chrome...		200	3483	script		
46	https://www.google.com	GET	/complete/search?client=chrome...		200	3664	JSON		

AI Assistant:

Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

You can now:

1. Inspect the error response in the Repeater tab
2. Try different payloads by modifying the request
3. Use the Burp Scanner for automated SQLi detection

Would you like me to:

1. Test for blind SQLi (e.g., time-based delays)
2. Use Burp Scanner for automated analysis
3. Help craft specific payloads for data extraction?

using burp mcp, please search for proxy history containing vulnweb.com

burp - get_proxy_history_regex (MCP)(count: 100, offset: 0, regex: "vulnweb\\.com")

Large MCP response (~27.2k tokens). this can fill up context quickly

```
["request": "GET / HTTP/1.1", "host": "testphp.vulnweb.com", "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36", "accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8", "targetPort": 80]
```

The proxy history search for vulnweb.com returned 100 matching entries (preview shown). The results include:

1. Initial request to testphp.vulnweb.com
 - Normal GET request to the root endpoint (/)
2. SQL Injection test
 - Payload: GET /listproducts.php?cat=1'
 - Resulted in a clear SQL syntax error response:
3. Other requests
 - Additional interactions with the vulnerable endpoint (/listproducts.php) and other pages.

Next Steps:

Would you like me to:

1. Extract specific details from the matching entries?
2. Analyze the SQLi payload response further?
3. Search for additional patterns (e.g., time-based payloads, data extraction attempts)?

run burp scanner for sql injection

Kesimpulan

Terobosan MCP pada era Artificial Intelligence adalah hal yang menarik, dengan demikian proses uji penetrasi dapat dipersingkat dengan bantuan AI untuk melakukan analisis, aksi, serta evaluasi yang terstruktur dan sistematis.

Ketika Data Ancaman Siber Berbicara Lebih Keras dari Alarm

BY PAKDE IWAN

Pada pukul dua dini hari, dashboard Cybersecurity Operations Center (CSOC) sebuah bank besar di Jakarta berkedip merah. Puluhan alert membanjiri layar—scanning di port 445 SMB, brute force attempt di login WordPress, dan traffic anomali ke IP yang belum pernah terlihat sebelumnya. Analyst yang bertugas malam itu menghadapi pertanyaan klasik yang menghantui setiap SOC di dunia: dari ratusan alert ini, mana yang benar-benar berbahaya, dan mana yang hanya noise?

Pertanyaan itu tidak bisa dijawab oleh SIEM sendirian. Dibutuhkan konteks—konteks tentang siapa threat actor di balik serangan, teknik apa yang mereka gunakan, sektor apa yang mereka targetkan, dan apakah pola serangan ini bagian dari kampanye yang lebih besar. Dibutuhkan, singkatnya, threat intelligence.

Inilah alasan kehadiran SacTi, singkatan dari Synergized Analytics of Cyber Threat Intelligence—sebuah platform Cyber Threat Intelligence (CTI) yang dikembangkan dan dioperasikan oleh Punggawa Cybersecurity. Kami berkesempatan menjelajahi platform ini secara langsung, dan apa yang kami temukan cukup mengesankan.

SacTi

PLATFORM CYBER THREAT
INTELLIGENCE (CTI)
DARI PUNGGAWA YANG
MENGAGREGASI JUTAAN
TELEMETRY DATA
MENJADI HIGH-FIDELITY
ACTIONABLE
INTELLIGENCE.
KAMI MEMBEDAH
MEKANISME INTI DI BALIK
KEANDALANNYA.



3.3M+ STIX Domain Objects	3.07M+ Cyber Observables	80.89M+ STIX Relationships	822 Intrusion Sets
--	---------------------------------------	---	------------------------------

Synergized Analytics of Cyber Threat Intelligence, by Filigran, modified and rebranded by Punggawa.

BUKAN SEKADAR AGGREGATOR THREAT FEED

Ketika mendengar istilah **“threat intelligence platform”** banyak praktisi langsung membayangkan dashboard sederhana yang mengumpulkan Indicator of Compromise (IoC) dari berbagai feed—kumpulan IP address mencurigakan, domain phishing, dan hash malware yang diterima secara pasif lalu ditampilkan dalam tabel. SacTi bukan itu.

Yang pertama kali menarik perhatian saat login adalah skala datanya. Di bagian atas dashboard, deretan angka menyambut dengan percaya diri: 3.3 juta STIX Domain Objects, 3.07 juta Cyber Observables, dan yang paling mencengangkan—80.89 juta STIX Relationships. Angka terakhir ini penting karena relationship adalah lem yang menyatukan seluruh puzzle intelijen—ia menghubungkan malware ke threat actor, threat actor ke teknik serangan, teknik serangan ke sektor industri yang ditarget, dan seterusnya.

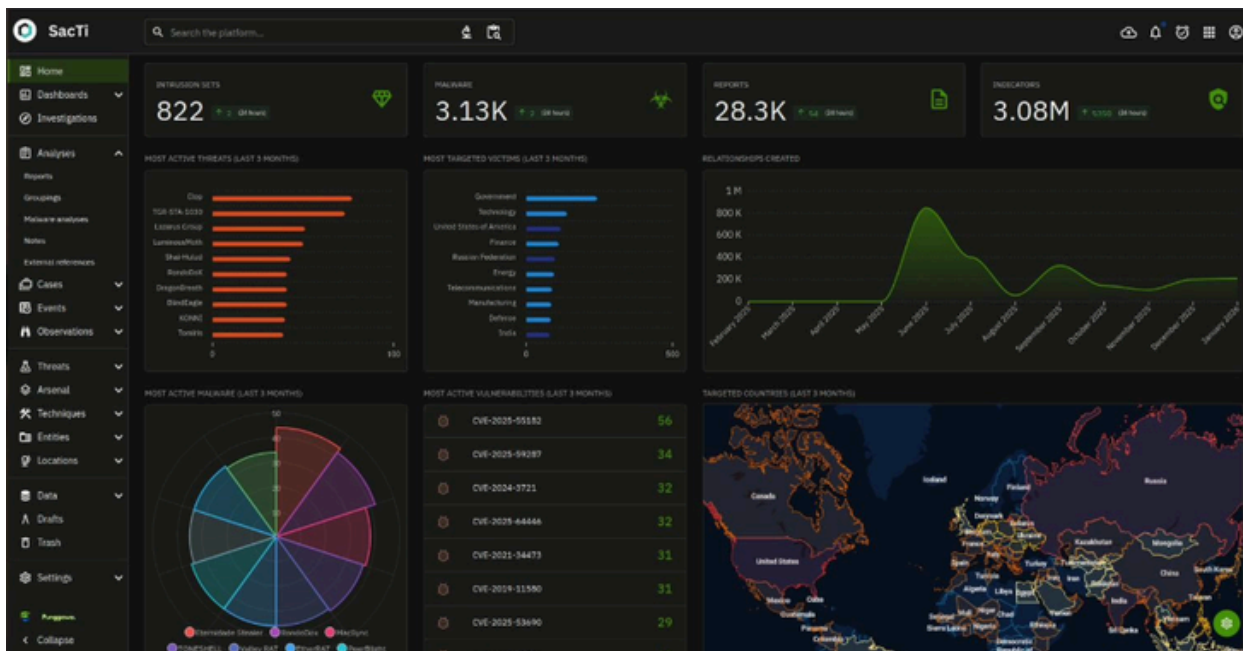
SacTi dibangun di atas fondasi OpenCTI (by Filigran), sebuah platform CTI open-source yang sudah diakui luas di komunitas keamanan siber global. Namun Punggawa tidak sekadar men-deploy OpenCTI out-of-the-box. Platform ini telah dimodifikasi, dikustomisasi, dan di-rebrand secara substansial—termasuk integrasi dengan sumber data regional, penyesuaian workflow, dan penambahan kapabilitas AI.



“Kami membangun SacTi bukan untuk menggantikan SIEM atau EDR, melainkan untuk memberikan konteks yang selama ini hilang dari alert-alert tersebut. Tanpa konteks, sebuah alert hanyalah baris log. Dengan konteks threat intelligence, ia menjadi cerita yang bisa ditindaklanjuti.”

FIRST IMPRESSION: DASHBOARD YANG BER CERITA

Halaman pertama setelah login menampilkan apa yang oleh tim Punggawa disebut sebagai “threat landscape overview.” Empat kartu besar di bagian atas menampilkan metrik kunci secara real-time: 822 intrusion sets yang terpantau, 3.130 varian malware, 28.300 lebih laporan ancaman, dan 3.08 juta indikator. Setiap angka disertai delta—perubahan dalam 24 jam terakhir. Angka-angka ini bukan statis; mereka bernafas.



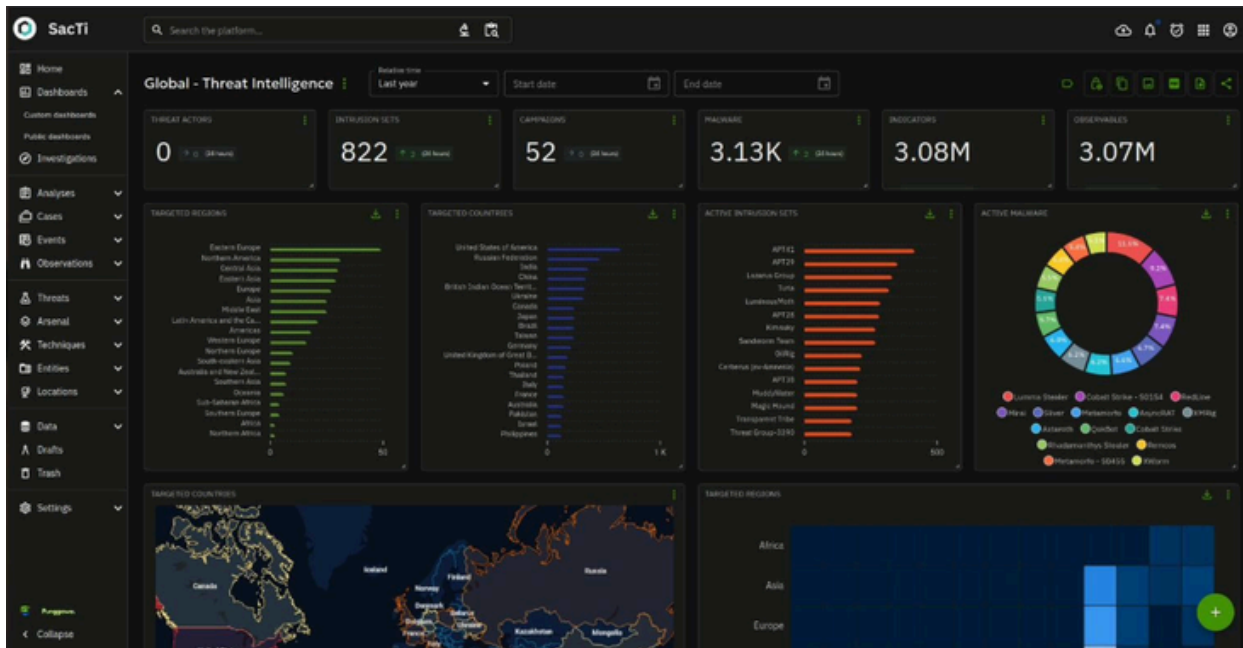
Homepage SacTi: overview lanskap ancaman dengan Most Active Threats (Clap, Lazarus Group), Most Targeted Victims (Government, Technology), Most Active Malware, dan Most Active Vulnerabilities.

Di bawah deretan kartu, enam panel visualisasi menyajikan gambaran panoramik. Panel Most Active Threats menunjukkan Clap mendominasi, disusul TGR-STA-1030, Lazarus Group, LuminousMoth, dan Shai-Hulud. Panel Most Targeted Victims mengungkapkan bahwa sektor Government dan Technology paling sering menjadi sasaran.

Peta dunia interaktif memvisualisasikan distribusi geografis target serangan—memberikan perspektif geopolitik yang sering hilang dari analisis teknis murni.

MENYELAMI DATA: GLOBAL THREAT INTELLIGENCE DASHBOARD

Di luar homepage, SacTi menyediakan dashboard terpisah bertajuk “Global – Threat Intelligence” yang menyajikan analisis lebih granular dengan tujuh metrik utama: Threat Actors, Intrusion Sets, Campaigns, Malware, Indicators, dan Observables.



Dashboard Global Threat Intelligence: Targeted Regions, Targeted Countries, Active Intrusion Sets (APT41, APT29, Lazarus, Turla, Kimsuky...), dan distribusi Active Malware (Lumma Stealer 11.5%, Cobalt Strike 9.2%).

Panel Active Intrusion Sets menyajikan daftar kelompok penyerang paling aktif: APT41 memimpin, diikuti APT29, Lazarus Group, Turla, LuminousMoth, APT28, Kimsuky, Sandworm Team, OilRig, dan Cerberus. Ini bukan daftar yang dibuat manual—melainkan hasil korelasi otomatis dari ribuan laporan dan indikator.

Di panel Active Malware, Lumma Stealer mendominasi dengan 11.5%, diikuti Cobalt Strike (SO154) dengan 9.2%. Kehadiran Cobalt Strike—yang secara teknis adalah legitimate red team tool—di peringkat kedua adalah pengingat nyata bahwa tools pertahanan sering dibalikkan menjadi senjata oleh adversary.

Platform ini juga mengadopsi STIX (Structured Threat Information Expression) sebagai format standar seluruh data ancaman. Lebih dari seperempat juta relationship baru tercipta setiap harinya—mesin korelasi SacTi bekerja tanpa henti menenun data menjadi jaring intelijen yang koheren. Grafik time-series menunjukkan pertumbuhan data dramatis sejak pertengahan 2025, dengan distribusi kontributor yang mengindikasikan tim Punggawa secara aktif mengkurasi data, bukan sekadar menelan feed mentah.

MENGENAL MUSUH: THREAT ACTOR PROFILING

Salah satu kapabilitas paling bernilai dari platform CTI adalah kemampuannya memprofilkan threat actor bukan sekadar mengetahui ada serangan, tetapi memahami siapa yang menyerang, apa motivasinya, alat apa yang digunakannya, dan siapa targetnya.

PATTERN TYPE	NAME	AUTHOR	CREATORS	LABELS	ORIGINAL CREATION DATE	MARKING
IP	malstrom.mn.de	AlienVault	admin	malware, phishing, ransomware	Jun 24, 2025, 5:37:08 PM	T.PAGE
IP	cedar-vry-burn.github.io	CyberHunterAutofeed	admin	No label	Jun 24, 2025, 5:40:11 PM	T.PAGE
IP	126c2f8b208c5af5756e76446838dc3836358621e0d...	AlienVault	admin	malware, phishing, ransomware	Jun 24, 2025, 5:39:56 PM	T.PAGE
IP	asumawagpawmto.apc	ADHON	admin	ransomware, system, network	Jun 25, 2025, 5:41:29 PM	T.PAGE
IP	45.38.203.252	CyberHunterAutofeed	admin	No label	Sep 13, 2025, 10:14:03 PM	T.PAGE
IP	200.9.136.199	IdeminerMiller	admin	malware, phishing, ransomware	Dec 3, 2025, 8:08:43 PM	T.PAGE
IP	http://ide.anserissar.com/prog/46b623331594d_output...	CyberHunterAutofeed	admin	malware, phishing, ransomware	Jun 24, 2025, 5:00:44 PM	T.PAGE
IP	265d48bc3471ec3c443203bce83846a9954313e0570d6f...	IdeminerMiller	admin	malware, phishing, ransomware	Jun 24, 2025, 5:14:56 PM	T.PAGE
IP	202fca580b27e6296d7a6480f0d9993a3630ca2294d1...	IdeminerMiller	admin	malware, phishing, ransomware	Jun 24, 2025, 5:09:37 PM	T.PAGE
IP	134.227.130.201	CyberHunterAutofeed	admin	No label	Dec 27, 2025, 9:20:27 AM	T.PAGE
IP	117.200.135.68	ADHON	admin	ransomware, system, network	Jun 25, 2025, 9:58:53 AM	T.PAGE
IP	663d990a0321790b32e05627dc3372db5c563b8d...	IdeminerMiller	admin	malware, phishing, ransomware	Jun 24, 2025, 5:14:52 PM	T.PAGE
IP	59.182.6.234	ADHON	admin	ransomware, system, network	Jun 25, 2025, 9:22:02 PM	T.PAGE
IP	hdcnyfgumstart.webflow.io	CyberHunterAutofeed	admin	No label	Jun 24, 2025, 5:29:02 PM	T.PAGE
IP	145.249.247.546	ADHON	admin	ransomware, system, network	Jun 25, 2025, 9:32:30 PM	T.PAGE

Profil Intrusion Sets: APT33 (HOLMIUM/Elfin), Kimsuky/Lazarus, Phorpiex, ReaverBits, FIN8, Candiru, dan lainnya—lengkap dengan alias, malware, negara target, dan sektor target.

SacTi menyimpan 822 intrusion set. APT33—juga dikenal sebagai HOLMIUM, Elfin, atau Peach Sandstorm—tercatat menargetkan Aerospace, Education, Energy, dan Defense menggunakan NETWIRE, TURNEDUP, Tickler, dan POWERTON. Kimsuky menyerang target di Kanada dengan Comeback dan BLINDINGCAN. ReaverBits dari Rusia menargetkan Finance dan Telecommunications menggunakan ReaverDoor dan Meduza Stealer. FIN8, kelompok bermotif finansial sejak 2016, menggunakan Sardonic dan Ragnar Locker untuk industri hospitality dan retail.

Setiap profil dilengkapi metadata lengkap: alias, malware, negara target, sektor target, dan korelasi ke intrusion set lain. Informasi ini langsung actionable bagi SOC analyst yang perlu menentukan apakah alert mereka mungkin terkait kampanye APT tertentu.

MITRE ATT&CK: DARI FRAMEWORK KE OPERASIONAL

Jika STIX adalah bahasa data SacTi, maka MITRE ATT&CK adalah kerangka analitiknya. Platform memetakan 1.580 lebih attack patterns, mencakup seluruh spectrum taktik dari initial access hingga impact.

ID	NAME	LABELS	ORIGINAL CREATION DATE	MODIFICATION DATE
T1496	[mitre-attack] impact	No label	Jun 24, 2025	Feb 19, 2026
T1572	[mitre-attack] command-and-control	No label	Jun 24, 2025	Feb 19, 2026
T1182	[mitre-attack] command-and-control	No label	Jun 24, 2025	Feb 19, 2026
T1870	[mitre-attack] defense-evasion	No label	Jun 24, 2025	Feb 19, 2026
T1562	[mitre-attack] defense-evasion	No label	Jun 24, 2025	Feb 19, 2026
T1547	[mitre-attack] privilege-escalation	No label	Jun 24, 2025	Feb 19, 2026
T1125	[mitre-attack] collection	No label	Jun 24, 2025	Feb 19, 2026
T1071.001	[mitre-attack] command-and-control	No label	Jun 24, 2025	Feb 19, 2026
T1547.001	[mitre-attack] privilege-escalation	No label	Jun 24, 2025	Feb 19, 2026
T1049	[mitre-attack] discovery	No label	Jun 1, 2017	Feb 19, 2026
T1021.006	[mitre-attack] lateral-movement	No label	Feb 12, 2020	Feb 19, 2026
T1114	[mitre-attack] collection	No label	Jun 24, 2025	Feb 19, 2026
T1133	[mitre-attack] persistence	No label	Jun 24, 2025	Feb 19, 2026
T1021.001	[mitre-attack] lateral-movement	No label	Feb 12, 2020	Feb 19, 2026
T1018	[mitre-attack] discovery	No label	Jun 24, 2025	Feb 19, 2026

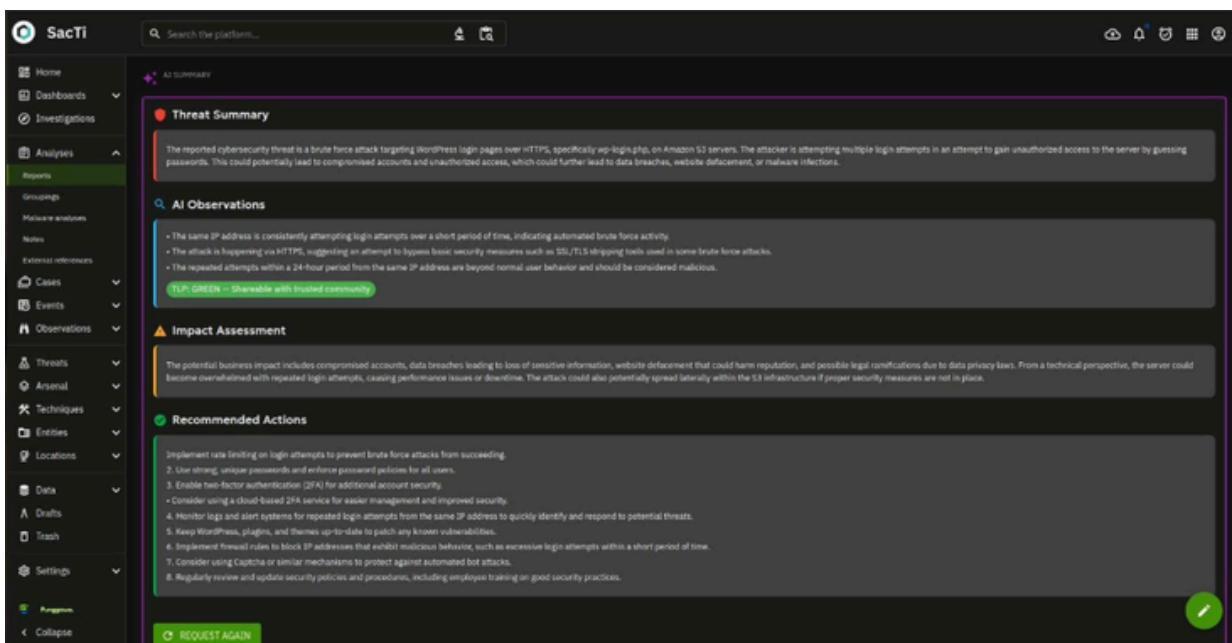
Detail teknik T1496 Resource Hijacking: deskripsi, platform terdampak (macOS, Linux, Windows, Containers, IaaS, SaaS), deteksi, dan sub-teknik (Compute Hijacking, Bandwidth Hijacking, SMS Pumping).

Kami menyelami T1496 Resource Hijacking sebagai contoh. Halaman detail menampilkan deskripsi lengkap tentang bagaimana adversary memanfaatkan sumber daya komputasi korban untuk mining cryptocurrency, platform yang terpengaruh, metode deteksi, dan sub-teknik: T1496.001 Compute Hijacking, T1496.002 Bandwidth Hijacking, dan T1496.003 SMS Pumping.

Nilai operasional mapping ini sangat nyata. Ketika SOC analyst menemukan lonjakan CPU yang tidak wajar, mereka bisa langsung mengkorelasikan temuan ke T1496, mengidentifikasi threat actor yang biasa menggunakan teknik ini, dan memprioritaskan respons berdasarkan risk assessment terstruktur.

28.000 LAPORAN DAN AI YANG MEMBANTU ANALYST BERPIKIR

Repository laporan SacTi menyimpan lebih dari 28.300 laporan dan terus bertambah setiap hari, berasal dari sumber seperti BotnetExposer (data honeypot real-time: SMB, Telnet, SQL Injection, bruteforce WordPress, SSH, RDP) serta AlienVault (analisis RAT dan kampanye cryptojacking). Setiap laporan memiliki label seperti mirai, honeypot, bruteforce, dan malicious IP untuk memudahkan pencarian, sementara fitur utama yang membedakan SacTi dari platform CTI lain adalah AI Insights, yang menganalisis laporan dan menghasilkan ringkasan yang langsung dapat ditindaklanjuti.



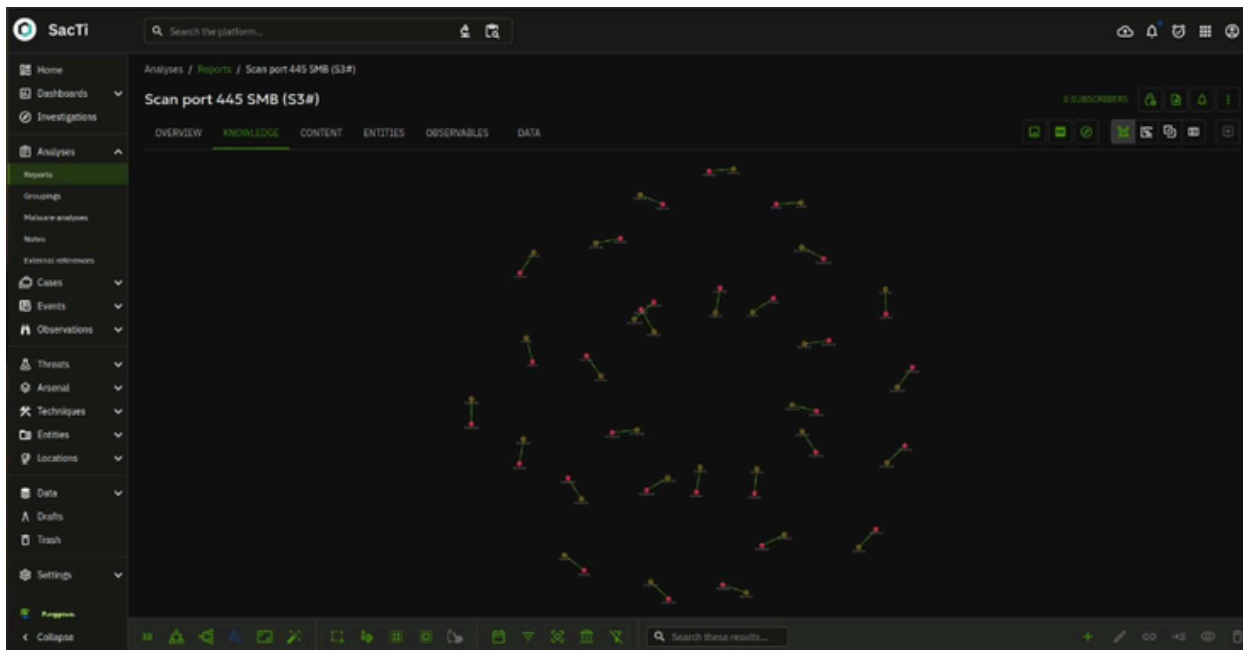
AI Insights pada laporan Bruteforce login WordPress: Threat Summary, AI Observations (pola brute force otomatis, HTTPS bypass), Impact Assessment, dan 8 Recommended Actions terstruktur.

Kami menguji fitur AI Insights pada laporan brute force login WordPress. AI menghasilkan empat komponen utama: Threat Summary yang mengidentifikasi serangan brute force pada wp-login.php melalui HTTPS di server Amazon S3, AI Observations yang menemukan pola login berulang dari IP yang sama dan frekuensi serangan tidak normal, Impact Assessment yang menilai risiko seperti akun terkompromi, kebocoran data, dan kerusakan reputasi, serta Recommended Actions yang mencakup mitigasi seperti rate limiting, password kuat, 2FA, monitoring, update WordPress, firewall rules, Captcha, dan review kebijakan keamanan.

Nilai utama fitur ini terletak pada kecepatan dan konsistensi analisis, sehingga analyst dapat fokus pada investigasi yang membutuhkan intuisi manusia, bukan pada proses peringkasan laporan yang repetitif.

KNOWLEDGE GRAPH: MELIHAT YANG TAK TERLIHAT

Mungkin fitur paling visually striking dari SacTi adalah knowledge graph-nya.



Knowledge Graph pada report Scan port 445 SMB: konstelasi node (IP, domain, hash) yang saling terhubung, membentuk cluster yang mengungkap pola operasional botnet.

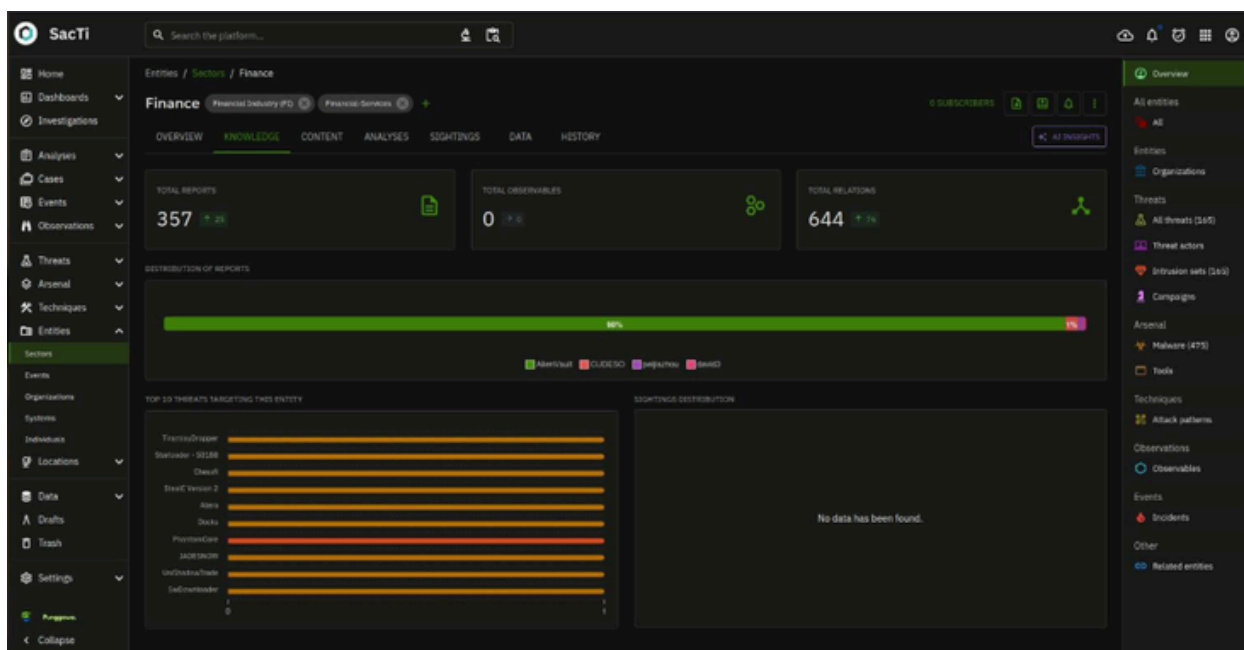
Ketika kami membuka tab Knowledge pada laporan Scan port 445 SMB, layar dipenuhi konstelasi node hijau dan merah yang saling terhubung—setiap node mewakili observable (IP address, domain, hash), dan setiap garis mewakili relasi. Visualisasi ini bukan sekadar eye candy. Ia mengungkap pola yang tidak terlihat dalam format tabular: cluster IP yang beroperasi secara koordinatif, hub node yang menjadi titik koneksi antara kampanye berbeda, dan outlier yang mungkin menunjukkan infrastruktur baru yang belum terkorelasi.

Bagi threat hunter, knowledge graph adalah kanvas investigasi. Dari satu IP mencurigakan, mereka bisa menelusuri ke domain terkait, dari domain ke malware yang pernah di-host, dari malware ke threat actor, dan dari threat actor ke sektor industri targetnya. Perjalanan investigasi yang memakan waktu berjam-jam dalam spreadsheet bisa dilakukan dalam hitungan menit.

ARSENAL, ENTITY MANAGEMENT, DAN CAMPAIGNS

SacTi tidak berhenti di threat intelligence saja. Tab Arsenal menyediakan katalog komprehensif: ratusan malware families (dari Astaroth hingga SwDownloader), 93 tools yang digunakan adversary (dari cmd dan pwdump hingga PoshC2 dan AADInternals), dan 192.590 CVE yang diperbarui mendekati real-time—banyak bertanggal Februari 2026.

Modul Entity Management mengelola taksonomi sektor industri yang terstruktur. Kami membuka sektor Finance dan menemukan granularitas yang mengejutkan: empat subsector (Market Infrastructures, Insurance Services, Financial Organizations, Banking Institutions), 357 laporan terkait, dan yang paling bernilai—Top 10 Threats yang secara spesifik menargetkan sektor finansial: TiramisuDropper, Starloader-SO188, ChessFi, StealC Version 2, hingga PhantomCore dan JADESNOW.



Knowledge tab sektor Finance: 357 reports, 644 relasi, dan Top 10 Threats yang secara spesifik menargetkan industri finansial—intelligence yang memungkinkan CISO memprioritaskan pertahanan berbasis data.

Intelligence semacam ini memungkinkan CISO sektor finansial memprioritaskan pertahanan berdasarkan ancaman yang benar-benar menargetkan industrinya—bukan berdasarkan asumsi generik.

Modul Campaigns mengkatalogasi 52 kampanye siber: dari Operation Spalax (menargetkan pemerintah Kolombia), Night Dragon (spionase energi), FunnyDream (kampanye Tiongkok di Asia Tenggara), hingga Oldsmar Treatment Plant Intrusion. Fitur bookmarking memungkinkan analyst menandai kampanye yang sedang diinvestigasi untuk akses cepat.

CATATAN DAN RUANG PERBAIKAN

SacTi juga memiliki beberapa area yang masih dapat ditingkatkan. Pertama, learning curve cukup curam karena banyaknya modul, sehingga pengguna baru membutuhkan waktu untuk memahami arsitektur data dan workflow; dokumentasi serta training yang lebih terstruktur akan sangat membantu. Kedua, kualitas AI Insights sangat bergantung pada kualitas data input—pada laporan dengan data yang minim, hasil analisis kadang terlalu generik. Ketiga, integrasi dua arah yang lebih seamless dengan SIEM dan SOAR, seperti auto-enrichment alert dan automated playbook triggering, akan meningkatkan nilai operasional platform secara signifikan.

Dalam keamanan siber, konteks adalah segalanya. SacTi memberikan sesuatu yang tidak bisa diberikan oleh SIEM atau EDR: kemampuan untuk memahami bukan hanya apa yang menyerang, tetapi siapa, mengapa, dan bagaimana.

KESIMPULAN

SacTi merupakan platform CTI yang ambisius dengan lebih dari 3,3 juta STIX objects, 80 juta relationships, integrasi MITRE ATT&CK yang komprehensif, serta fitur AI Insights yang memberikan analisis langsung actionable. Kekuatan utamanya bukan hanya pada volume data, tetapi pada cara data diorganisasi, dikorelasikan, dan disajikan—melalui dashboard informatif, knowledge graph untuk menemukan relasi tersembunyi, dan AI yang membantu analyst mengekstrak insight dari ribuan laporan.

Bagi organisasi di Indonesia, khususnya sektor finansial, pemerintahan, dan infrastruktur kritis yang sering menjadi target APT, SacTi dapat menjadi komponen penting dalam ekosistem pertahanan siber dengan menyediakan konteks intelijen yang tidak diberikan oleh **SIEM maupun EDR**.

INFO PRODUK

- ▶ Nama Produk: SacTi (Synergized Analytics of Cyber Threat Intelligence)
- ▶ Pengembang: PT. Punggawa Siber Solusi
- ▶ Basis Platform: Filigran OpenCTI, modified & rebranded by Punggawa
- ▶ Key Capabilities: CTI Dashboard, MITRE ATT&CK Mapping, AI Insights, Knowledge Graph, Threat Actor Profiling, Observable & Indicator Management
- ▶ Data Scale: 3.3M+ STIX Objects | 3.07M+ Observables | 80.89M+ Relationships | 822 Intrusion Sets | 28.3K+ Reports | 192K+ CVEs
- ▶ Intelligence Sources: AlienVault, CyberHunterAutoFeed, BotnetExposer, CUDESQ, dan lainnya
- ▶ Target Pengguna: SOC Analyst, Threat Hunter, CSIRT, CISO, Security Architect
- ▶ Informasi: platform.punggawa.app | www.punggawa.com



HashiCorp Vault dalam Ekosistem Cyber Resilience

Mengamankan Secret di Era Zero Trust: Lebih dari Sekadar Password Manager

BY PAKDE IWAN

Pada awal 2024, Departemen Keuangan Amerika Serikat mengalami insiden kebocoran data yang mengejutkan. Penyerang tidak menggunakan zero-day exploit yang canggih, tidak memanfaatkan kerentanan perangkat lunak yang belum terdeteksi, dan tidak menjalankan serangan brute-force yang memakan waktu berminggu-minggu. Yang mereka butuhkan hanyalah satu API key yang bocor dari platform BeyondTrust. Satu credential. Satu kesalahan pengelolaan. Jutaan dolar investasi keamanan yang sudah dibangun bertahun-tahun — dilewati begitu saja.

Insiden ini bukan anomali. Menurut laporan State of Secrets Sprawl 2025 dari GitGuardian, sebanyak 23,8 juta secrets baru ditemukan bocor di repositori publik GitHub sepanjang tahun 2024 — naik 25% dari tahun sebelumnya. Yang lebih mengkhawatirkan, 70% dari secrets yang bocor pada tahun 2022 masih aktif dan valid hingga hari ini, memberikan penyerang jendela akses yang terus terbuka selama bertahun-tahun.

Data dari Verizon's 2025 DBIR menunjukkan bahwa 22% dari seluruh insiden kebocoran data bermula dari credential yang dicuri atau dikompromikan — lebih tinggi dari kategori serangan lainnya. SpyCloud menambahkan bahwa hampir 80% dari breach di tahun 2024 melibatkan penggunaan credential curian. Rata-rata waktu yang dibutuhkan untuk mengidentifikasi dan meremediasi insiden berbasis credential adalah 292 hari.

Fakta-fakta ini menggambarkan sebuah krisis yang tersembunyi namun sangat nyata: secret sprawl. Dan di sinilah HashiCorp Vault hadir — bukan sekadar sebagai password manager, melainkan sebagai secrets operating system untuk infrastruktur modern.

Anatomi Secret Sprawl: Bom Waktu di Infrastruktur Modern

Secret sprawl terjadi ketika credential, API key, token, certificate, dan data sensitif lainnya tersebar tanpa kontrol di berbagai tempat dalam ekosistem teknologi organisasi. Ini adalah masalah yang tumbuh secara organik seiring bertambahnya kompleksitas infrastruktur, dan hampir setiap organisasi mengalaminya tanpa sepenuhnya menyadarinya.

Dimana Secrets Bersembunyi ?

Secrets tidak hanya bocor dari source code, tetapi tersebar di berbagai sistem kolaborasi organisasi. Laporan GitGuardian 2025 menunjukkan 23,8 juta secrets baru ditemukan di GitHub pada 2024, dengan repository privat 9 kali lebih rentan karena rasa “aman” membuat developer kurang hati-hati. Kebocoran juga muncul di collaboration tools—sekitar 6,1% tiket Jira dan 2,4% channel Slack mengandung credential, biasanya saat troubleshooting. Di container ecosystem, 98% secrets di DockerHub ditemukan dalam image layers, termasuk sekitar 7.000 AWS key yang masih aktif dari perusahaan Fortune 500. Selain itu, 32% secrets yang terdeteksi scanner terkait dengan infrastruktur CI/CD, menjadikan pipeline otomatis sebagai salah satu sumber penyimpanan credential yang sering tidak terkelola.



Mengapa Ini Terjadi?

Akar masalahnya adalah gap antara kecepatan development dan kedisiplinan keamanan. Dalam tekanan deadline dan sprint cycles, developer mengambil jalan pintas: hardcoding API key langsung di source code, menyimpan password database di file konfigurasi, dan membagikan credential melalui Slack atau email.

Penggunaan AI coding assistants seperti GitHub Copilot — yang meningkat 27% antara 2023 dan 2024 — justru memperburuk situasi. Repository yang menggunakan Copilot memiliki leak rate 6,4%, menunjukkan bahwa AI mempercepat development tetapi juga mempercepat terjadinya insiden keamanan.

Konteks Indonesia: Dengan semakin banyaknya organisasi yang mengadopsi cloud-native architecture dan microservices, serta ketatnya regulasi dari OJK, BI, dan PCI DSS yang mewajibkan pengelolaan credential secara aman, kebutuhan akan secrets management yang terstruktur menjadi sangat mendesak.

Mengenal HashiCorp Vault: Secrets Operating System

HashiCorp Vault adalah identity-based secrets management system yang dirancang untuk menyimpan, mengontrol akses, mengenkripsi, dan merotasi secrets secara terprogram di seluruh infrastruktur. Vault bukan sekadar brankas digital — ia adalah sebuah sistem operasi untuk secrets yang mengintegrasikan identity management, access control, encryption, dan audit dalam satu platform terpadu.

Vault dikembangkan oleh HashiCorp, perusahaan yang juga dikenal dengan produk-produk infrastructure automation seperti Terraform, Consul, dan Boundary. Pada Februari 2025, IBM resmi menyelesaikan akuisisi HashiCorp senilai USD 6,4 miliar — sebuah langkah strategis yang memperkuat portofolio hybrid cloud dan keamanan IBM. Dengan akuisisi ini, HashiCorp kini beroperasi sebagai divisi di bawah IBM Software, dan produk-produk utamanya termasuk Vault dan Terraform telah menjadi bagian integral dari portofolio automasi IBM.

Akuisisi oleh IBM membawa dampak signifikan: investasi R&D yang lebih besar, jangkauan global yang lebih luas, dan integrasi yang lebih dalam dengan ekosistem IBM termasuk Red Hat Ansible, OpenShift, dan watsonx. Armon Dadgar, CTO dan Co-founder HashiCorp, menegaskan bahwa HashiCorp akan melanjutkan misinya dengan misi yang sama namun di panggung yang lebih besar.

Sebagai IBM Gold Partner, PT Punggawa Siber Solusi (Punggawa Cybersecurity) memiliki posisi strategis dalam mendistribusikan dan mengimplementasikan Vault bagi organisasi di Indonesia. Status Gold Partner ini menegaskan kompetensi dan komitmen Punggawa dalam ekosistem IBM, memberikan akses ke resources teknis, dukungan prioritas, dan sertifikasi yang memastikan kualitas implementasi. Kolaborasi ini telah dibuktikan antara lain melalui forum diskusi “Strengthening Cyber Defense in the Era of Digital Transformation” yang diselenggarakan bersama IBM pada Desember 2025, yang membahas secara mendalam strategi pengelolaan secrets dan credential menggunakan Vault.



Selain Vault, kemitraan Punggawa dengan IBM juga mencakup Punggawa Cyber Resilience Center yang diperkuat oleh IBM QRadar — mengintegrasikan monitoring, adaptive defense, dan intelligence-led decision making dalam satu framework keamanan terpadu. Kombinasi Vault untuk secrets management dan QRadar untuk security operations memberikan solusi end-to-end yang komprehensif bagi organisasi yang serius membangun postur cyber resilience.

Arsitektur Vault: Tiga Lapisan Pertahanan



1. External Access Layer.

Seluruh interaksi dengan Vault dilakukan melalui HTTPS API yang terenkripsi. Tidak ada akses langsung ke storage backend. Vault menyediakan tiga interface: CLI (command line), Web UI, dan HTTP API untuk integrasi programatik.

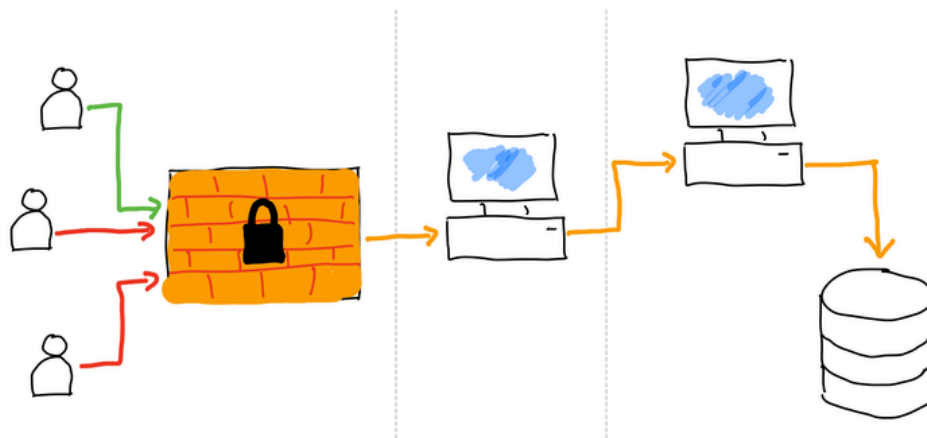
2. Internal Components (The Barrier).

Lapisan ini adalah jantung Vault, berisi komponen-komponen logis yang menyediakan fungsionalitas inti: Auth Methods untuk autentikasi, Secrets Engines untuk mengelola berbagai jenis secrets, Policy Engine untuk access control, Token Store untuk manajemen sesi, dan Audit Devices untuk logging.

3. Storage Backend.

Seluruh data yang disimpan Vault dienkripsi sebelum ditulis ke storage backend. Backend ini bersifat untrusted — artinya, bahkan jika penyerang mendapatkan akses langsung ke storage, mereka hanya akan menemukan data terenkripsi yang tidak bisa dibaca tanpa encryption key Vault.

Konsep kunci: Vault menggunakan Shamir's Secret Sharing algorithm untuk melindungi master encryption key. Key ini dipecah menjadi beberapa shares yang didistribusikan ke beberapa orang. Untuk membuka (unseal) Vault, diperlukan minimum threshold shares memastikan tidak ada satu individu pun yang memiliki akses penuh.



Lima Kapabilitas Inti Vault

1. Dynamic Secrets

Berbeda dari secrets statis yang dibuat sekali dan digunakan selamanya, dynamic secrets dibuat on-demand untuk setiap request, memiliki Time-To-Live (TTL), dan otomatis di-revoke setelah expired. Contoh: sebuah microservice meminta credential database dari Vault → Vault membuat username dan password unik yang hanya valid selama 30 menit → setelah expired, credential otomatis dihapus dari database.

Keuntungan: tidak ada credential yang berlama-lama hidup (long-lived), tidak ada credential yang digunakan bersama (shared), dan blast radius jika credential dicuri sangat minimal karena masa hidupnya yang singkat.

```
# Mengaktifkan database secrets engine
$ vault secrets enable database

# Membaca dynamic credential - unik setiap kali dipanggil
$ vault read database/creds/readonly

Key          Value
---          -
lease_id     database/creds/readonly/a1b2c3d4
lease_duration 30m
username     v-app-readonly-x7k9m2p1
password     A1b-Cd2Ef3Gh4Ij5Kl6
```

Contoh pembuatan dynamic database credential melalui Vault CLI.

2. Encryption as a Service (Transit Engine)

Transit engine memungkinkan aplikasi melakukan enkripsi dan dekripsi tanpa harus mengelola encryption key sendiri. Aplikasi cukup mengirim plaintext ke Vault, menerima ciphertext, lalu menyimpannya di database, sementara seluruh manajemen kunci—termasuk rotation dan versioning—ditangani oleh Vault. Pendekatan ini mengatasi masalah umum ketika developer harus mengimplementasikan kriptografi sendiri yang sering kali tidak aman, karena dengan Transit engine proses enkripsi cukup dilakukan melalui API call sederhana.

3. Identity-Based Access Control

Vault mendukung berbagai metode autentikasi yang berbasis identitas, bukan sekadar credential statis. Ini termasuk Kubernetes service accounts, AWS IAM roles, Azure Active Directory, LDAP, GitHub, dan OIDC. Setiap identitas yang terautentikasi dipetakan ke entity di Vault, yang kemudian diikat dengan policy set yang menentukan secrets mana yang bisa diakses.

Model ini mengimplementasikan prinsip Zero Trust secara literal: setiap request harus di-authenticate dan di-authorize, terlepas dari lokasi jaringan atau status sebelumnya.

4. Policy-Driven Governance

Vault menggunakan policy yang ditulis dalam HashiCorp Configuration Language (HCL) untuk mendefinisikan akses secara fine-grained. Policy menentukan path mana yang bisa diakses, operasi apa yang diizinkan (read, write, delete, list), dan batasan apa yang berlaku.

Pendekatan ini memastikan prinsip least privilege ditegakkan secara konsisten: setiap aplikasi, service, dan user hanya mendapatkan akses ke secrets yang benar-benar mereka butuhkan.

```
# Contoh policy: hanya boleh membaca secrets di path tertentu
path "secret/data/production/database" {
  capabilities = ["read"]
}

path "secret/data/staging/*" {
  capabilities = ["create", "read", "update"]
}
```

Contoh policy HCL yang membatasi akses berdasarkan path dan capability.

5. Comprehensive Audit Trail

Setiap interaksi dengan Vault — setiap request, response, autentikasi, dan akses ke secrets — dicatat dalam audit log yang terperinci. Log ini mencakup siapa yang mengakses, apa yang diakses, kapan, dari mana, dan apakah akses berhasil atau ditolak.

Audit trail ini bukan hanya untuk kebutuhan compliance (OJK, BI, PCI DSS, ISO 27001), tetapi juga menjadi sumber data forensik yang sangat berharga ketika insiden terjadi. Tim SOC dan incident responder dapat dengan cepat menelusuri credential mana yang terpengaruh dan akses apa saja yang perlu di-revoke.



Vault dalam Konteks Cyber Resilience

Vault bukan produk yang berdiri sendiri. Nilai terbesar Vault muncul ketika ia diintegrasikan ke dalam strategi keamanan yang lebih luas. Bagian ini menjelaskan bagaimana Vault berperan sebagai pilar fundamental dalam membangun organisasi yang cyber resilient.

Vault sebagai Pilar Zero Trust

Model keamanan Zero Trust berprinsip “never trust, always verify”, di mana tidak ada user, aplikasi, atau service yang dipercaya secara default—setiap akses harus melalui autentikasi, otorisasi, dan enkripsi. Dalam konteks secrets management, Vault menerapkan prinsip ini dengan mewajibkan setiap request membawa token autentikasi yang valid (never trust), memverifikasi akses melalui policy engine sebelum secrets diberikan (always verify), mengenkripsi seluruh data baik at rest maupun in transit termasuk komunikasi internal dengan TLS (encrypt everything), serta menggunakan dynamic secrets dengan TTL pendek untuk membatasi dampak jika credential bocor (minimize blast radius).

Secrets Lifecycle Management

Dalam pendekatan tradisional, lifecycle sebuah secret terlihat seperti ini: credential dibuat secara manual, disimpan di file konfigurasi atau spreadsheet, dibagikan via email atau chat, tidak pernah dirotasi, dan baru di-revoke setelah insiden terjadi. Pendekatan ini penuh dengan kerentanan di setiap tahapnya.

Vault mengubah lifecycle ini secara fundamental:

Tahap	Tradisional	Dengan Vault
Creation	Manual, ad hoc	Otomatis, on-demand
Storage	File config, spreadsheet, chat	Encrypted vault, centralized
Access	Siapa saja yang tahu	Identity-based, policy-driven
Rotation	Jarang atau tidak pernah	Otomatis dengan TTL
Revocation	Manual, sering terlambat	Otomatis saat expired/incident
Audit	Tidak ada trail	Setiap akses tercatat

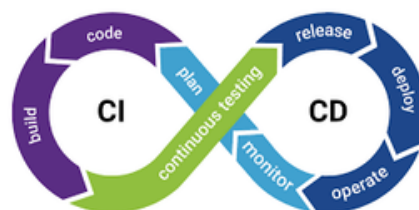
Integrasi dengan Ekosistem DevSecOps

Vault dirancang untuk terintegrasi secara native dengan ekosistem DevSecOps modern. Ini menjadi sangat relevan ketika dihubungkan dengan artikel lain dalam edisi ini tentang DevSecOps: Otomatisasi Vulnerability Scanning dengan Droid LLM Hunter.



Vault + Kubernetes

Vault menyediakan Vault Agent Sidecar Injector yang secara otomatis menyuntikkan secrets ke dalam pod Kubernetes tanpa developer perlu menulis kode khusus. Workflow-nya: pod dibuat → Vault Agent melakukan autentikasi menggunakan Kubernetes service account → secrets yang sesuai dengan policy diambil dari Vault → secrets di-render sebagai file atau environment variable di dalam pod.



Vault + CI/CD Pipeline

Dalam pipeline Jenkins, GitLab CI, atau GitHub Actions, Vault menggantikan kebutuhan untuk menyimpan secrets di environment variables CI/CD. Pipeline melakukan autentikasi ke Vault menggunakan metode yang sesuai (JWT token, AppRole), mengambil credential yang dibutuhkan, menggunakan credential tersebut, dan credential otomatis expired setelah pipeline selesai.



Vault + Cloud Platforms

Vault mendukung secrets engine untuk seluruh cloud platform utama: AWS, Azure, dan Google Cloud. Untuk setiap platform, Vault dapat membuat credential sementara (STS tokens untuk AWS, service principals untuk Azure, IAM credentials untuk GCP) yang secara otomatis di-revoke setelah masa berlakunya habis. Ini menghilangkan kebutuhan untuk menyimpan long-lived cloud credentials di mana pun.

HCP Vault Radar: Menemukan Secrets yang Sudah Terlanjur Bocor

Vault Radar adalah fitur yang memperluas kemampuan Vault dari hanya mengelola secrets menjadi juga menemukan dan meremediasi secrets yang sudah terlanjur tersebar. Vault Radar bekerja dalam empat tahap:

1. Discover. Vault Radar memindai platform seperti GitHub, Confluence, Jira, dan sumber lainnya untuk menemukan secrets yang terekspos.
2. Classify. Setiap secret yang ditemukan diklasifikasikan berdasarkan jenis (AWS key, database credential, API token), tingkat risiko, dan konteks lingkungan.
3. Remediate. Vault Radar menyediakan jalur untuk mengimpor secrets yang ditemukan ke dalam Vault, di mana mereka bisa dikelola secara terpusat dan aman.
4. Prevent. Integrasi dengan CI/CD pipeline memungkinkan Vault Radar memblokir commit yang mengandung secrets sebelum kode masuk ke repositori.

Vault Radar menjadi jawaban langsung untuk masalah secret sprawl yang diuraikan di bagian awal artikel ini: bukan hanya mencegah secrets bocor di masa depan, tetapi juga menemukan dan membersihkan secrets yang sudah bocor.

Skenario Implementasi di Indonesia

Berikut tiga skenario implementasi yang menggambarkan bagaimana Vault dapat diterapkan dalam konteks organisasi di Indonesia:

SKENARIO 1

Perbankan – Rotasi Credential Otomatis untuk Compliance

Sebuah bank nasional menghadapi tantangan compliance: regulasi OJK dan PCI DSS mewajibkan rotasi credential database secara berkala, tetapi proses manual yang ada memakan waktu dua hari per siklus rotasi dan melibatkan tiga tim berbeda (DBA, infrastructure, dan security). Kesalahan manusia dalam proses ini pernah menyebabkan downtime selama empat jam.

Solusi Vault: Bank mengimplementasikan Vault's database secrets engine dengan dynamic credentials. Setiap aplikasi mendapatkan credential unik dengan TTL 24 jam. Rotasi terjadi secara otomatis tanpa intervensi manusia. Audit trail otomatis memenuhi kebutuhan compliance untuk pelaporan ke OJK.

Hasil: Waktu rotasi credential berkurang dari 2 hari menjadi instan. Zero downtime akibat kesalahan rotasi. Laporan audit tersedia secara otomatis untuk regulator.

SKENARIO 2

Fintech – Secrets Management untuk Microservices

Sebuah startup fintech yang menjalankan 40+ microservices di Kubernetes menghadapi masalah credential sprawl: setiap service menyimpan credential di Kubernetes Secrets sebagai base64-encoded strings (yang bukan enkripsi), dan tidak ada mekanisme rotasi otomatis. Saat seorang engineer resign, tim security membutuhkan dua minggu untuk mengidentifikasi dan merotasi semua credential yang pernah diaksesnya.

Solusi Vault:

Fintech mengimplementasikan Vault Agent Sidecar Injector di cluster Kubernetes. Setiap pod mengautentikasi ke Vault menggunakan Kubernetes service account, dan menerima credential yang hanya valid selama lifecycle pod tersebut. PKI engine digunakan untuk mutual TLS antar service.

Hasil: Nol credential statis di Kubernetes Secrets. Offboarding karyawan tidak lagi memerlukan credential rotation manual. Mutual TLS antar service menambah lapisan keamanan komunikasi.

SKENARIO 3

Post-Breach Recovery Membersihkan Secret Sprawl

Setelah mengalami insiden kebocoran data melalui credential yang bocor di repository publik, sebuah perusahaan e-commerce melakukan security assessment dan menemukan ratusan secrets tersebar di GitHub repositories, Confluence pages, dan Slack channels. Tim security kewalahan dengan skala masalahnya.

Solusi Vault: Perusahaan men-deploy Vault Radar untuk memindai seluruh ekosistem dan mengidentifikasi semua secrets yang terekspos. Secrets kritis diimpor ke Vault dan dirotasi secara otomatis. Pipeline CI/CD diintegrasikan dengan Vault Radar untuk mencegah secrets masuk ke repositori di masa depan.

Hasil: 347 secrets teridentifikasi dan diremediasi dalam 72 jam. Proses yang sebelumnya diestimasi memakan waktu 3 bulan manual. Kebijakan preventif mencegah terulangnya insiden.

Vault dan Red Team: Perspektif Ofensif

Hubungan antara Vault dan red team operation tidak bisa diabaikan. Red team yang menggunakan threat intelligence (seperti dibahas dalam artikel #10 edisi ini tentang Threat Intelligence–Driven Red Teaming) secara rutin memanfaatkan platform CTI seperti SacTi (artikel #8) untuk mengidentifikasi threat actor yang relevan. Banyak dari threat actor ini – Lazarus Group, APT33, FIN8 – memulai serangan mereka dengan mencari credential yang bocor.

Vault secara langsung mengurangi attack surface yang bisa dieksploitasi oleh red team maupun adversary nyata:

- Dynamic secrets menghilangkan long-lived credentials yang sering menjadi temuan pentest.
- Identity-based access menghapus kebutuhan untuk credential statis yang bisa dicuri.
- Audit trail memberikan visibility ketika credential digunakan secara mencurigakan.
- Vault Radar menemukan secrets yang sudah bocor sebelum adversary menemukannya.

Koneksi dengan edisi ini: SacTi menyediakan threat intelligence tentang adversary yang menargetkan organisasi. Red team mensimulasikan serangan berbasis TTPs adversary tersebut. Vault (artikel ini) mengamankan aset yang paling sering menjadi target: credentials dan secrets. Bersama, ketiganya membentuk siklus: Detect → Test → Protect.

Penutup: Secrets Management Bukan Lagi Opsional

Pada lanskap ancaman siber 2026, pengelolaan secrets secara manual tidak lagi memadai karena satu credential yang bocor dapat menimbulkan kerugian finansial, kerusakan reputasi, dan sanksi regulasi. HashiCorp Vault menghadirkan solusi komprehensif melalui dynamic secrets yang menghilangkan credential statis, encryption-as-a-service untuk menyederhanakan kriptografi, identity-based access yang mendukung model Zero Trust, serta Vault Radar untuk mengatasi secret sprawl.



Sebagai IBM Gold Partner, Punggawa Cybersecurity dapat mendistribusikan dan mengimplementasikan Vault di Indonesia—didukung oleh akuisisi HashiCorp oleh IBM senilai USD 6,4 miliar pada Februari 2025—dengan layanan konsultasi arsitektur, implementasi lokal, serta integrasi dengan ekosistem keamanan mereka seperti Cyber Resilience Center berbasis IBM QRadar dan platform CTI SacTi.

Pada akhirnya, pertanyaannya bukan lagi apakah organisasi membutuhkan secrets management, tetapi seberapa cepat dapat mengimplementasikannya sebelum kebocoran credential berikutnya terjadi.



Threat Intelligence Driven Red Teaming

Simulasi Serangan Berbasis Ancaman Nyata

BY PAKDE IWAN

Dari Generic Pentest ke Adversary Emulation:

Bagaimana Red Team Modern Meningkatkan Realisme dengan CTI

Paradoks mendasar dari red teaming tradisional adalah engagement yang sukses secara teknis belum tentu relevan secara strategis. Manajemen mungkin merasa aman setelah diuji, namun seringkali kebobolan oleh kelompok APT (Advanced Persistent Threat) yang sesungguhnya karena ancaman yang disimulasikan berbeda dengan realitas. Artikel dari Punggawa Magazine edisi Maret 2026 ini membedah urgensi transisi menuju Threat Intelligence-Driven Red Teaming.

Mengapa Pendekatan Tradisional Tidak Lagi Relevan?

Pendekatan red team konvensional memiliki kelemahan struktural yang membahayakan postur pertahanan:



- Berbasis Tools, Bukan Ancaman Fokus awal selalu pada "tools apa yang akan digunakan?", sehingga skenario serangan cenderung seragam di seluruh industri.



- Mengabaikan Lanskap Ancaman Spesifik Menggunakan teknik generik (seperti phishing standar atau Cobalt Strike) mengabaikan fakta bahwa threat actor di sektor finansial sangat berbeda motivasi dan tekniknya dibandingkan di sektor kesehatan.



- Bias Validasi Deteksi Organisasi seringkali hanya menguji kemampuan mendeteksi tools keamanan standar, bukan kemampuan mendeteksi custom backdoor yang lazim digunakan oleh musuh sesungguhnya.



- Kesenjangan Pelaporan Laporan tradisional hanya menyajikan daftar kerentanan tanpa konteks strategis terkait ancaman nyata, sehingga prioritas mitigasi menjadi tebak-an.

Memahami Threat Intelligence Driven Red Teaming

Threat Intelligence Driven Red Teaming adalah pendekatan operasional yang menggunakan Cyber Threat Intelligence (CTI) sebagai fondasi setiap aspek engagement—mulai dari ruang lingkup, pengembangan skenario, hingga metrik keberhasilan. Alih-alih bertanya "bisakah kita masuk?", pendekatan ini fokus pada "bisakah adversary X masuk dengan cara Y, dan apakah organisasi bisa mendeteksinya?".

Pendekatan ini bertumpu pada tiga pilar utama:

Pilar	Deskripsi	Peran dalam Engagement
Threat Intelligence	Data dan analisis tentang adversary yang relevan: siapa mereka, apa motivasinya, bagaimana mereka beroperasi	Menjadi input utama untuk merancang skenario serangan yang realistis dan relevan
MITRE ATT&CK Framework	Bahasa universal untuk mendokumentasikan dan mengkomunikasikan TTPs adversary	Memetakan teknik adversary ke teknik yang akan disimulasikan, dan mengukur detection coverage
Adversary Emulation	Proses mereplikasi atau mengadaptasi TTPs spesifik dari threat actor teridentifikasi	Eksekusi simulasi serangan yang merepresentasikan ancaman nyata, bukan skenario hipotetis

Validasi dari framework ini dapat dilihat pada adopsi TIBER-EU (Threat Intelligence–Based Ethical Red Teaming) di Eropa, yang prinsip pengujian berkala dan realistisnya sangat relevan untuk diadopsi lembaga keuangan di Indonesia guna memenuhi regulasi OJK dan BI.

TIBER-EU (Threat Intelligence–based Ethical Red Teaming) adalah kerangka kerja Eropa untuk pengujian keamanan siber yang komprehensif. TIBER-EU memandu perusahaan dalam melakukan serangan siber terkontrol yang meniru taktik aktor ancaman nyata untuk menguji dan meningkatkan kemampuan perlindungan, deteksi, serta respons terhadap serangan pada sistem kritis mereka.

Workflow Operasional: 5 Fase Eksekusi

Pelaksanaan TI-Driven Red Teaming yang matang mengikuti lima tahapan terstruktur:

1. Threat Landscape Assessment

- Tim CTI membedah profil organisasi (industri, geolokasi, tech stack) untuk mengidentifikasi ancaman paling spesifik.
- Contoh konkret: untuk sebuah bank di Indonesia, analisis melalui platform CTI menunjukkan bahwa threat actor paling relevan mencakup Lazarus Group (motivasi finansial, sejarah menargetkan institusi keuangan di Asia), FIN8 (spesialisasi di sektor hospitality dan finansial), serta kampanye Astaroth yang menargetkan lembaga keuangan di Amerika Latin dan mulai berekspansi ke Asia Tenggara.
- Informasi ini menjadi fondasi untuk seluruh engagement.

2. Threat Actor Profiling & TTP Mapping

- Setiap teknik yang digunakan adversary diekstrak menjadi blueprint serangan dan dipetakan langsung ke dalam taksonomi MITRE ATT&CK.
- Sebagai ilustrasi, profiling Lazarus Group untuk engagement di sektor finansial akan menghasilkan matriks TTP seperti berikut:

Taktik (ATT&CK)	Teknik Lazarus	ID ATT&CK
Initial Access	Spear phishing dengan lampiran berisi macro weaponized	T1566.001
Execution	PowerShell script execution dan Windows Command Shell	T1059.001, T1059.003
Persistence	Registry run keys dan scheduled tasks	T1547.001, T1053.005
Defense Evasion	Process injection dan DLL side-loading	T1055, T1574.002
Credential Access	LSASS memory dumping dan credential harvesting	T1003.001
Lateral Movement	SMB/Windows Admin Shares dan remote services	T1021.002
Collection	Data from local system dan screen capture	T1005, T1113
Exfiltration	Exfiltration over C2 channel	T1041
Impact	Data encrypted for impact (ransomware)	T1486

3. Skenario Development & Planning Pengembangan

- Fidelity vs. feasibility, yaitu menjaga realisme TTP adversary namun tetap dapat dieksekusi dalam batas waktu dan scope engagement.
- Objective alignment, di mana tujuan serangan mencerminkan motivasi threat actor—misalnya kelompok finansial seperti Lazarus Group akan menargetkan sistem bernilai tinggi seperti SWIFT atau core banking.
- Detection opportunity mapping untuk mengidentifikasi titik deteksi yang seharusnya terlihat oleh blue team.
- Deconfliction plan untuk memastikan aktivitas red team tidak bertukar dengan insiden keamanan nyata.

5. Execution & Purple Team Collaboration

- Fase eksekusi adalah tahap ketika skenario yang telah dirancang dijalankan oleh red team sementara blue team tetap beroperasi normal tanpa mengetahui detail engagement (kecuali pada pendekatan kolaboratif).
- Berikut adalah pendekatan eksekusi TI-Driven Red Teaming yang bisa dipilih berdasarkan maturitas organisasi:

Pendekatan	Deskripsi	Cocok Untuk
Full Blind (Red Only)	Blue team tidak mengetahui engagement berlangsung. Menguji detection dan response sesungguhnya.	Organisasi dengan maturitas keamanan tinggi yang ingin validasi realistis.
Assumed Breach	Red team memulai dari titik tertentu (misalnya, sudah memiliki initial foothold). Menghemat waktu untuk fokus pada post-exploitation.	Ketika initial access sudah diketahui rentan dan fokusnya pada lateral movement dan detection.
Purple Team Integration	Pada titik-titik tertentu, red team dan blue team berkolaborasi. Red team menjelaskan teknik yang dieksekusi, blue team memeriksa apakah terdeteksi.	Organisasi yang ingin memaksimalkan learning effect dan detection improvement.
Hybrid	Fase awal dijalankan blind, kemudian beralih ke purple teaming untuk teknik yang tidak terdeteksi.	Pendekatan paling umum yang menyeimbangkan realisme dan learning.

5. Reporting & Detection Gap Analysis

Titik akhir dari proses ini bukanlah sekadar rilis tingkat keparahan celah keamanan, melainkan laporan analisis kesenjangan deteksi. Deliverable utamanya adalah Detection Gap Analysis sebuah analisis sistematis tentang teknik mana yang terdeteksi, mana yang lolos, dan mengapa. Berikut adalah komponen utama Detection Gap Analysis:

- **ATT&CK Coverage Heat Map** – Visualisasi cakupan deteksi terhadap teknik adversary untuk melihat teknik yang terdeteksi, sebagian terdeteksi, atau tidak terdeteksi oleh SOC.
- **Detection Timeline Analysis** – Mengukur waktu dari eksekusi serangan hingga deteksi serta efektivitas proses triage dan eskalasi.
- **Root Cause Analysis** – Mengidentifikasi penyebab teknik tidak terdeteksi, seperti kekurangan log, rule deteksi yang tidak ada, atau konfigurasi yang tidak optimal.
- **Prioritized Remediation Roadmap** – Menentukan prioritas perbaikan berdasarkan kritikalitas teknik dan kemudahan implementasi mitigasi.
- **Detection Rule Recommendations** – Memberikan rekomendasi spesifik seperti sumber log yang diperlukan, logika deteksi, atau rule yang dapat langsung diimplementasikan oleh tim SOC.

Peran Platform CTI Dalam Red Team Operation

Keberhasilan Threat Intelligence–Driven Red Teaming sangat bergantung pada ketersediaan threat intelligence yang terstruktur dan actionable, sehingga platform Cyber Threat Intelligence (CTI) berfungsi sebagai force multiplier bagi red team. Platform CTI mengagregasi berbagai sumber intelijen dan menstrukturkannya (misalnya dalam format STIX) untuk menghasilkan blueprint adversary yang dapat digunakan dalam perencanaan serangan.

Nilai utama platform CTI bagi red team meliputi:

- **Threat Actor Identification** – Mengidentifikasi intrusion sets yang aktif menargetkan sektor dan wilayah organisasi sehingga perencanaan serangan berbasis data, bukan asumsi.
- **TTP Extraction** – Mengekstrak teknik serangan yang dipetakan ke framework seperti MITRE ATT&CK untuk membangun rencana adversary emulation yang komprehensif.
- **IOC Contextualization** – Memahami pola infrastruktur adversary seperti domain, hosting, dan protokol command-and-control untuk menciptakan simulasi yang lebih realistis.
- **Campaign Correlation** – Mengkorelasikan kampanye serangan dengan sektor dan wilayah target agar skenario red team mencerminkan ancaman yang sedang berlangsung.

Studi Kasus: TI–Driven Red Teaming di Sektor Finansial

Sebuah lembaga keuangan besar di Asia Tenggara dengan infrastruktur hybrid dan tim SOC beranggotakan 15 orang menjalankan pentest tahunan selama tiga tahun, namun hasilnya semakin terbatas. Untuk meningkatkan efektivitas, dilakukan pendekatan Threat Intelligence–Driven Red Teaming.

Phase 1–2: Intelligence Gathering & Profiling Analisis threat landscape mengidentifikasi tiga threat actor prioritas:

- Lazarus Group – dikenal menargetkan institusi finansial melalui supply chain attack, spear phishing, dan teknik living-off-the-land.
- FIN8 – berfokus pada pencurian data kartu kredit dan sistem pembayaran, namun juga melakukan reconnaissance pada sektor finansial.
- Astaroth – malware banking trojan yang menggunakan teknik fileless dan LOLBins.

Phase 3: Scenario Development Dikembangkan dua skenario utama:

- Skenario A (Lazarus Emulation) – spear phishing ke tim treasury dengan macro weaponized, diikuti PowerShell post-exploitation, credential dumping LSASS, lateral movement via SMB, dan upaya akses sistem SWIFT.
- Skenario B (Astaroth Adaptation) – serangan fileless melalui watering hole, menggunakan LOLBins seperti BitsAdmin, certutil, dan mshta untuk payload execution tanpa menyentuh disk, dengan target workstation teller.

Phase 4: Execution (Hybrid Approach)

- Engagement dilakukan dua minggu full blind dan dua minggu purple teaming.
- Fase blind: Skenario A berhasil mendapatkan initial access dan credential dumping (terdeteksi setelah 4 jam) namun terblokir oleh network segmentation pada sistem SWIFT. Skenario B berhasil menjalankan payload fileless tanpa terdeteksi karena aktivitas LOLBins tidak dimonitor EDR.
- Fase purple teaming: Analisis bersama menemukan root cause—EDR hanya memonitor executable baru, bukan pola command-line LOLBins. Tim kemudian mengembangkan 12 detection rules baru.

Phase 5: Detection Gap Analysis Dari 24 teknik yang disimulasikan, 42% terdeteksi real-time, 25% terdeteksi terlambat, dan 33% tidak terdeteksi sama sekali. Kesenjangan terbesar ditemukan pada fileless execution dan penyalahgunaan LOLBins, yang jarang diuji dalam pentest tradisional.

Rekomendasi Memulai TI-Driven Red Teaming

Bagi organisasi yang ingin mengadopsi pendekatan ini, berikut adalah rekomendasi praktis berdasarkan maturitas keamanan:

Organisasi Maturitas Awal

- **Lakukan Threat Landscape Assessment** – Identifikasi adversary yang paling relevan menggunakan CTI atau layanan threat intelligence yang berbasis data.
- **Fokus pada Satu Threat Actor** – Lakukan adversary emulation secara mendalam, jangan mencoba semua sekaligus.
- **Gunakan Purple Teaming** – Lebih efektif untuk meningkatkan kemampuan deteksi tim SOC yang masih berkembang.

Organisasi Maturitas Menengah-Tinggi

- **Integrasikan CTI ke Program Red Team** – Setiap engagement harus berbasis threat intelligence yang relevan.
- **Gunakan Pendekatan Hybrid** – Kombinasikan fase blind (validasi realistis) dan purple teaming (peningkatan deteksi).
- **Ukur Detection Coverage** – Gunakan heat map dari MITRE ATT&CK untuk memantau peningkatan deteksi.

Untuk Semua Organisasi

- **Bangun Kolaborasi CTI-Red Team-SOC** – CTI menyediakan intelijen, red team menguji, SOC memperbaiki.
- **Lakukan Secara Berkala** – Jalankan engagement berulang dengan threat actor berbeda untuk memperluas coverage deteksi dan meningkatkan resilience.



Punggawa Cybersecurity: Partner untuk TI-Driven Red Teaming

Metodologi Threat Intelligence-Driven Red Teaming yang dibahas dalam artikel ini telah diimplementasikan secara nyata oleh Punggawa Cybersecurity sejak 2019 melalui layanan keamanan siber end-to-end yang digunakan berbagai organisasi di Indonesia, yaitu:



- **Red Team Operation** – Simulasi serangan dunia nyata berbasis threat intelligence, menguji ketahanan people, process, dan technology dari initial access hingga objective completion.



- **Vulnerability Assessment & Penetration Testing (VAPT)** – Evaluasi kerentanan yang komprehensif oleh tim ahli, menjadi fondasi sebelum organisasi meningkatkan maturitas ke red teaming berbasis adversary.



- **Cyber Threat Hunting & Intelligence** – Deteksi ancaman proaktif yang terintegrasi dengan CSOC untuk meningkatkan kemampuan detection dan response secara berkelanjutan.



- **Social Engineering Assessment** – Simulasi serangan manipulasi untuk menguji kesiapan karyawan dan memperkuat faktor manusia sebagai lini pertahanan.



- **Cybersecurity Operation Center (CSOC) 24x7** – Monitoring, deteksi, dan respons insiden berkelanjutan yang mengintegrasikan hasil red team untuk memperkuat aturan deteksi.

Seluruh layanan didukung oleh platform **SacTi (Synergized Analytics of Cyber Threat Intelligence)** hasil riset tim Punggawa Cybersecurity yang menyediakan intelijen tentang lebih dari 822 intrusion sets, 3.13K malware families, serta jutaan indikator ancaman. Platform ini memungkinkan Punggawa memberikan wawasan mendalam kepada pelanggan mengenai kampanye phishing, malware, dan aktivitas kelompok APT, sekaligus menjadi sumber intelijen utama bagi operasi red teaming dan threat hunting.

Integrasi antara threat intelligence, red team, dan CSOC menciptakan siklus keamanan berkelanjutan **Detect → Test → Protect** yang membantu organisasi meningkatkan ketahanan terhadap ancaman siber nyata.

Penutup: Era Baru Red Teaming

Lanskap ancaman siber tidak pernah bersifat generik setiap organisasi menghadapi adversary yang berbeda, dengan motivasi, kapabilitas, dan TTPs yang unik. Maka, pendekatan untuk menguji pertahanan organisasi pun seharusnya tidak generik. **Threat Intelligence-Driven Red Teaming** adalah jawaban atas paradoks yang telah lama menghantui industri keamanan siber: bagaimana memastikan bahwa investasi dalam pengujian keamanan benar-benar menguji apa yang perlu diuji.

Dengan mengintegrasikan **cyber threat intelligence** ke dalam setiap fase red team engagement dari scoping hingga reporting—organisasi mendapatkan lebih dari sekadar daftar vulnerability. Mereka mendapatkan pemahaman mendalam tentang kesiapan mereka menghadapi adversary yang sesungguhnya, **detection gaps yang spesifik dan terukur**, serta **roadmap** perbaikan yang berbasis ancaman nyata.

Ini bukan tentang membuat **red teaming** menjadi lebih rumit. Ini tentang membuatnya menjadi lebih **relevan**. Dan di dunia di mana threat actor terus berevolusi dan memperbaiki taktik mereka, relevansi adalah satu-satunya metrik yang benar-benar penting. Dengan ekosistem layanan yang terintegrasi dari platform **CTI SacTi**, red team operation berbasis intelligence, hingga CSOC yang beroperasi 24x7 **Punggawa Cybersecurity** hadir sebagai mitra yang siap membantu organisasi Anda membangun cyber resilience yang sesungguhnya.





Scan for download
MAGAZINE PUNGGAWA VOLUME 7



ask.sales@punggawa.com



[punggawacyber](https://www.instagram.com/punggawacyber)



[PunggawaCyber](https://www.facebook.com/PunggawaCyber)



[Punggawa Cybersecurity](https://www.linkedin.com/company/Punggawa%20Cybersecurity)