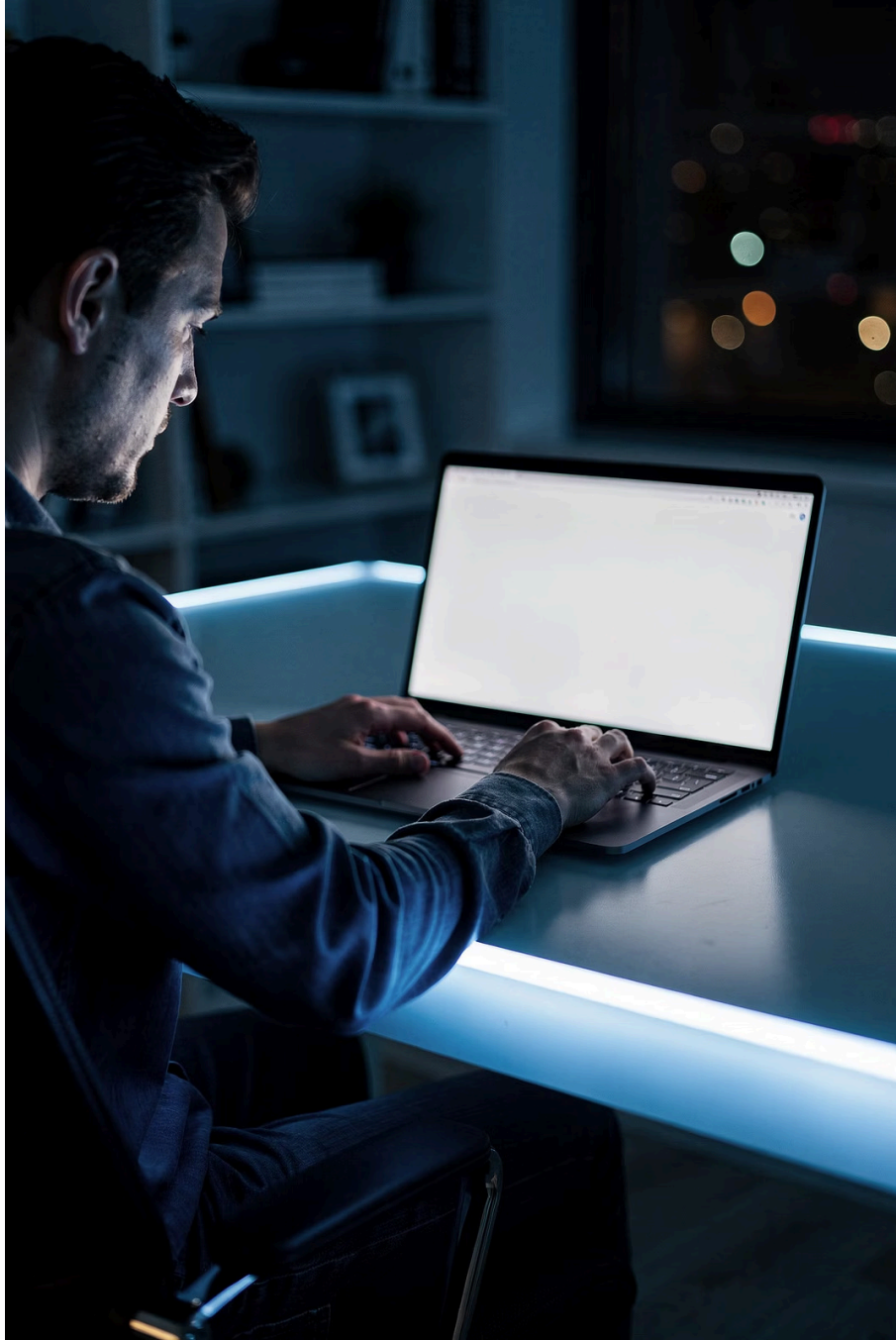
Modul 5: Reporting Like a Professional

Pelajari cara membuat laporan bug bounty yang disukai tim keamanan perusahaan — jelas, terstruktur, mudah diverifikasi, dan bernilai tinggi.

EDY SUSANTO - FOUNDER C-SIX SECURITY



Tujuan Pembelajaran

Di akhir modul ini, peserta akan memiliki kemampuan praktis untuk menulis laporan keamanan yang profesional dan dapat langsung digunakan dalam program bug bounty nyata.

📋 Struktur Laporan

Memahami komponen wajib dalam laporan bug bounty yang baik dan benar.

🔬 Bukti & PoC

Mampu menyusun Proof of Concept dan mengumpulkan evidence yang kuat.

⚖️ Penilaian Dampak

Menganalisis severity dan dampak bisnis dari setiap kerentanan yang ditemukan.

✅ Laporan Siap Submit

Menulis laporan lengkap yang lolos validasi dan menghindari kesalahan umum.

Gambaran Umum

Mengapa Laporan yang Baik Sangat Penting?

Laporan = Nilai Anda

Temuan teknis yang brilian tidak ada artinya jika tidak dapat dikomunikasikan dengan baik. Tim keamanan perusahaan menilai Anda bukan hanya dari kerentanan yang ditemukan, tetapi dari kualitas laporan yang Anda kirimkan.

Dampak Nyata Laporan Berkualitas

- Lebih cepat divalidasi dan dibayar bounty-nya
- Reputasi tinggi di platform bug bounty
- Hubungan profesional yang baik dengan perusahaan
- Peluang mendapat undangan ke program private
- Portofolio yang menarik bagi calon employer

Struktur Laporan Bug Bounty

Setiap laporan profesional mengikuti struktur yang konsisten dan dapat diprediksi — memudahkan triage, validasi, dan eskalasi oleh tim keamanan perusahaan.

Komponen Wajib Laporan Bug Bounty

1	**Judul Vulnerability** Singkat, spesifik, dan deskriptif. Contoh: **"Stored XSS pada parameter komentar /blog/post"** — bukan hanya "XSS Found".
2	**Ringkasan Eksekutif** Penjelasan singkat 2–3 kalimat tentang apa yang ditemukan, di mana, dan mengapa berbahaya. Targetkan pembaca non-teknis sekalipun bisa memahaminya.
3	**Severity & CVSS Score** Nyatakan tingkat keparahan (Critical/High/Medium/Low) beserta justifikasi atau skor CVSS jika relevan.
4	**Langkah Reproduksi** Urutan langkah yang detail, jelas, dan dapat diulang oleh orang lain dari awal hingga eksploitasi terkonfirmasi.
5	**Impact Analysis** Jelaskan konsekuensi nyata: data apa yang bisa dicuri, akun siapa yang bisa diambil alih, atau sistem apa yang bisa dirusak.
6	**Rekomendasi Perbaikan** Berikan saran remediasi yang praktis dan actionable — tunjukkan bahwa Anda memahami root cause dari kerentanan.

Proof of Concept (PoC)

PoC adalah inti dari laporan Anda — bukti konkret bahwa kerentanan benar-benar ada dan dapat dieksploitasi. PoC yang lemah adalah alasan paling umum laporan ditolak.

Apa yang Harus Ada di PoC?

- Payload atau kode eksploitasi yang digunakan
- URL atau endpoint yang rentan (full path)
- Request HTTP lengkap (header + body)
- Response dari server yang membuktikan eksploitasi
- Screenshot atau video rekaman serangan

Prinsip PoC yang Baik

- **Minimal damage** — jangan eksploitasi data nyata pengguna
- **Reproducible** — siapapun bisa mengulangnya step-by-step
- **Targeted** — fokus pada satu kerentanan saja
- **Etis** — hentikan setelah bukti cukup, jangan lanjutkan lateral movement

Reproduction Steps yang Efektif

Reproduction steps yang buruk memaksa tim keamanan membuang waktu mencoba memahami temuan Anda. Tulis seolah-olah panduan untuk seseorang yang tidak tahu apa pun tentang konteks Anda.

01

Tentukan Prasyarat

Sebutkan akun, role, atau kondisi yang diperlukan sebelum memulai. Contoh: "Login sebagai user biasa (bukan admin)".

02

Navigasi ke Titik Rentan

Berikan URL eksak, menu, atau fitur yang harus dikunjungi. Sertakan screenshot halaman awal.

03

Jalankan Aksi Spesifik

Jelaskan input, klik, atau request yang dilakukan. Sertakan payload lengkap dalam format code block.

04

Tunjukkan Hasil

Dokumentasikan output atau behavior yang membuktikan eksploitasi — screenshot, response body, atau log.

Impact Analysis

Impact Analysis mengubah laporan teknis menjadi narasi bisnis — menjawab pertanyaan:
"Seberapa besar kerusakan nyata yang bisa terjadi?"

Cara Menulis Impact Analysis yang Kuat

Dampak terhadap Data

Identifikasi data apa yang terekspos atau dapat dimodifikasi. Sebutkan jenis data (PII, kredensial, data finansial) dan estimasi jumlah record yang berisiko.

Dampak terhadap Pengguna

Jelaskan apakah akun pengguna dapat diambil alih, disadap, atau dimanipulasi. Berikan skenario serangan yang realistis.

Dampak terhadap Bisnis

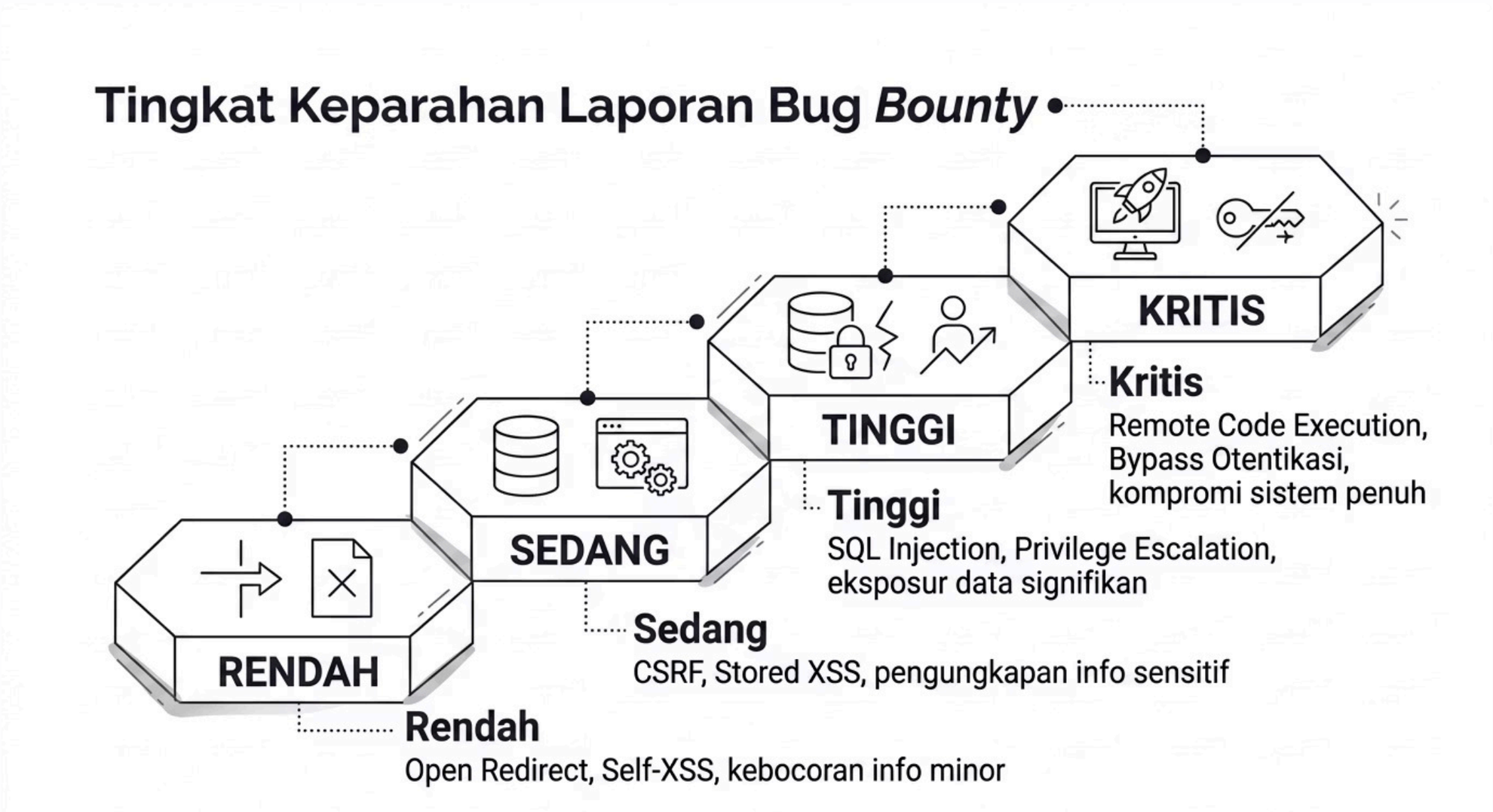
Hubungkan kerentanan dengan konsekuensi bisnis: reputasi, kerugian finansial, kepatuhan regulasi (GDPR, PCI-DSS), atau gangguan layanan.

Chained Attack Scenario

Jika kerentanan dapat digabungkan dengan bug lain untuk dampak yang lebih besar, jelaskan skenario rantai serangan tersebut secara singkat.

Severity Assessment

Penilaian severity yang akurat membantu triager memprioritaskan laporan Anda. Overestimate atau underestimate severity dapat merusak kredibilitas Anda.



Gunakan framework **CVSS v3.1** atau panduan severity resmi dari platform (HackerOne, Bugcrowd) sebagai acuan objektif dalam menentukan tingkat keparahan.

Screenshot & Evidence Collection

Evidence yang lemah = laporan yang ditolak. Kumpulkan bukti secara sistematis sejak pertama kali Anda menemukan kerentanan — jangan coba merekonstruksi setelah sesi berakhir.

Jenis Evidence yang Diperlukan

- **Screenshot** — tampilkan URL bar, timestamp, dan respons server
- **HTTP Logs** — export request/response dari Burp Suite atau proxy
- **Video PoC** — untuk kerentanan yang sulit dijelaskan dengan kata-kata
- **Console Output** — untuk kerentanan logic atau JavaScript

Best Practices Pengumpulan Bukti

- Sensor data pribadi pengguna nyata sebelum melampirkan
- Gunakan akun test / sandbox Anda sendiri sebagai target PoC
- Simpan file asli, bukan hanya tangkapan layar yang dipotong
- Tambahkan anotasi (panah, highlight) untuk memperjelas poin utama
- Sertakan timestamp dan versi browser/tools yang digunakan

Kesalahan Umum dalam Pelaporan Bug

Menghindari kesalahan umum ini dapat secara signifikan meningkatkan rasio penerimaan laporan dan reputasi Anda di platform bug bounty.

Common Reporting Mistakes & Cara Menghindarinya

❌ Judul Terlalu Generik

Buruk: "XSS vulnerability found"
Baik: "Reflected XSS pada parameter `search` di /products — eksekusi JavaScript tanpa autentikasi"

❌ Langkah Reproduksi Tidak Lengkap

Melewatkan prasyarat, menggunakan "dll." atau "dan seterusnya", atau mengasumsikan triager sudah tahu konteksnya. Tulis setiap langkah secara eksplisit.

❌ Severity Terlalu Tinggi / Rendah

Melaporkan Self-XSS sebagai Critical atau SSRF ke internal sebagai Low merusak kepercayaan. Selalu justifikasi severity dengan argumen teknis yang solid.

❌ Tidak Ada Impact Statement

Banyak researcher hanya menjelaskan "apa" tanpa "mengapa penting". Tanpa impact statement yang jelas, laporan cenderung di-downgrade atau ditolak.

❌ Evidence Tidak Relevan atau Kurang

Screenshot yang tidak menampilkan URL, payload yang dipotong, atau video tanpa audio penjelasan membuat triager tidak dapat memverifikasi temuan.

Sesi Praktik: Membuat Laporan Bug Lengkap

Pada sesi praktik ini, peserta akan menyusun laporan bug bounty yang lengkap berdasarkan skenario yang diberikan, kemudian mensimulasikan proses submit ke platform.

1

Fase 1: Analisis Skenario

Pahami temuan yang diberikan — identifikasi jenis kerentanan, endpoint, dan kondisi eksploitasi.

2

Fase 2: Susun Laporan

Tulis semua komponen laporan: judul, ringkasan, severity, reproduction steps, PoC, dan impact analysis.

3

Fase 3: Kumpulkan Evidence

Siapkan screenshot, HTTP log, dan payload yang diperlukan sebagai lampiran laporan.

4

Fase 4: Simulasi Submit

Submit laporan ke platform simulasi dan lakukan peer review antar peserta untuk mendapat feedback konstruktif.

Outcome: Apa yang Anda Kuasai Setelah Modul Ini

🖊️ Laporan Profesional

Mampu menyusun laporan bug bounty yang terstruktur, jelas, dan memenuhi standar industri.

🔍 PoC yang Kuat

Menghasilkan bukti eksploitasi yang dapat diverifikasi dan etis — meningkatkan rasio penerimaan laporan.

📊 Severity Akurat

Menilai dampak kerentanan secara objektif menggunakan framework yang diakui industri.

🚀 Siap Berkompetisi

Memiliki portofolio laporan yang dapat digunakan di platform bug bounty publik maupun private program.

Peserta yang menguasai seni pelaporan memiliki keunggulan kompetitif yang signifikan — laporan yang baik sering kali lebih dihargai daripada temuan yang banyak namun buruk kualitasnya.

EDY SUSANTO - FOUNDER C-SIX SECURITY