



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Protezione e sicurezza nei sistemi operativi: AI e Cybersecurity

Ozalp Babaoglu
Professore Alma Mater
Università di Bologna
www.cs.unibo.it/babaoglu/

Da tool ad attore

- 2023–2024: l'AI assiste l'attaccante — phishing migliore, codice suggerito; l'umano resta l'operatore
- 2025–2026: l'AI esegue — agent con accesso a tool, shell e browser lavorano da soli per ore
- Lo stesso salto in difesa: da copilot a sistemi che trovano e riparano vulnerabilità senza supervisione
- La conseguenza è economica: la competenza offensiva smette di essere un bene scarso
- Il costo marginale di un attacco competente tende a zero. Il costo di difendersi no

@ Babaoglu

ALMA MATER STUDIORUM

2

Mappa della lezione

- Parte 1 — AI offensiva: cinque case study documentati, 2024–2026 (35 min)
- Parte 2 — AI difensiva: AlxCC, Big Sleep, pentest autonomo, SOC agentico (25 min)
- Parte 3 — Sicurezza dell'AI: l'AI come bersaglio (35 min)
- Parte 4 — Strumenti: toolbox open source e laboratorio (10 min)
- Parte 5 — Governance e futuro: EU AI Act, previsioni (15 min)
- Prerequisiti: Lezioni 1–3. Non serve conoscere il machine learning

@ Babaoglu

ALMA MATER STUDIORUM

3

Tre domande guida

- Chi avvantaggia? L'attaccante deve trovare una strada, il difensore deve chiuderle tutte: l'automazione cambia il bilancio?
- Cosa è davvero nuovo? Quali sono nuove classi di minaccia e quali vecchie minacce più veloci e più economiche
- Di chi è la colpa? Se un agent autonomo causa un danno: il modello, chi lo ha integrato, o chi lo ha eseguito?
- Nessuna delle tre ha oggi una risposta unanime nella comunità

@ Babaoglu

ALMA MATER STUDIORUM

4

Quattro anni, in scala

- Nov 2022 — ChatGPT pubblico: phishing linguisticamente perfetto a costo zero
- Gen 2024 — frode Arup: deepfake in videocall, 25,6 M\$ in un giorno
- Ago 2025 — finale DARPA AlxCC; ESET documenta PromptLock
- Nov 2025 — GTG-1002: prima campagna di spionaggio orchestrata da AI
- Mag 2026 — primo zero-day scoperto e weaponizzato con AI
- Set 2026 — agent che riscrivono il malware finché l'EDR non lo rileva più

Tre assi da non confondere

- AI for offense — l'AI è lo strumento dell'attaccante: ricognizione, exploit, social engineering, orchestrazione
- AI for defense — l'AI è lo strumento del difensore: vulnerability discovery, patching, triage, pentest continuo
- Security of AI — l'AI è il bersaglio: prompt injection, poisoning, abuso di tool, supply chain dei modelli
- Il terzo asse è il più trascurato nei corsi e il più probabile nella vostra carriera

Glossario minimo

- LLM — modello che predice la continuazione di un testo; non distingue nativamente istruzioni da dati
- Tool use — il modello emette una richiesta strutturata: è il runtime, non il modello, a eseguirla
- Agent — loop osserva, decide, chiama tool, ripete: centinaia di iterazioni senza umano
- MCP — standard aperto per esporre tool e dati a un agent: di fatto un bus di driver per LLM
- Analogia col corso: il modello è un processo non fidato, il runtime è il kernel che fa enforcement

Parte 1 AI offensiva

- Cinque casi documentati, non ipotetici
- Per ognuno: cosa è successo, cosa ha fatto l'AI, cosa ci insegna

La kill chain, compressa

- Reconnaissance — OSINT su scala: social, leak, repo pubblici, organigrammi
- Weaponization — scoperta di vulnerabilità e scrittura dell'exploit, incluso zero-day
- Delivery — spear phishing personalizzato, voce e video clonati, pretexting multi-canale
- Exploitation — adattamento a ambienti sconosciuti senza script predefiniti
- Installation e C2 — malware che si rigenera; orchestrazione autonoma di più host
- Nessuna fase è nuova: cambia il rapporto fra costo e risultato

GTG-1002: attacco orchestrato da AI

- Nov 2025 — attore valutato da Anthropic, con alta confidenza, come state-sponsored cinese
- Circa 30 bersagli: aziende tech, istituzioni finanziarie, industria chimica, agenzie governative
- L'AI ha eseguito l'80–90% delle operazioni tattiche in autonomia, a ritmi impossibili per un umano
- L'umano autorizzava solo il passaggio a sfruttamento attivo e l'esfiltrazione finale
- Alcune intrusioni sono riuscite prima del rilevamento
- Rilevato non dalle vittime ma dal fornitore del modello, sui pattern di utilizzo

Come hanno aggirato le protezioni

- Pretesto — il modello è convinto di lavorare per una società di sicurezza, su un pentest autorizzato
- Decomposizione — l'obiettivo è spezzato in micro-task, ciascuno innocuo se letto da solo
- Esecuzione — scansione, exploit, movimento laterale, raccolta credenziali
- Funziona perché il filtro valuta la singola richiesta, non la sequenza
- Il limite dell'attaccante: il modello allucinava credenziali e dati inventati
- L'inaffidabilità dell'AI è oggi un freno all'attacco automatizzato, e non durerà

Il primo zero-day armato con AI

- 11 mag 2026 — Google GTIG: un gruppo criminale usa un modello per scoprire una vulnerabilità sconosciuta e scriverne l'exploit
- La falla: bypass del 2FA per un'assunzione di fiducia hardcoded nella logica di autenticazione
- Obiettivo dell'attore: una campagna di sfruttamento di massa
- Google coordina la disclosure: patch arrivata prima dell'esecuzione
- Attribuito all'AI da indizi stilistici: CVSS allucinato, docstring didattiche, Python da manuale
- Il codice generato lascia impronte stilistiche. Per ora

Arup: 25,6 M\$ in una videocall

- Gen 2024 — spear phishing che impersona il CFO britannico: transazione riservata
- Il dipendente sospetta ed esita: il controllo umano funziona
- Viene organizzata una videoconferenza: il CFO e i colleghi, volto e voce, sono deepfake
- Il sospetto svanisce. Quindici bonifici, circa 25,6 M\$, in un solo giorno
- Non è fallita la tecnologia: è fallito il protocollo di verifica
- Il video è un canale non autenticato: usarlo come fattore di autenticazione è un errore di progettazione

Deepfake: rilevare non è difendersi

- Circa 1,1 miliardi di dollari di perdite USA da frodi con deepfake nel 2025
- Gli umani riconoscono i video deepfake di alta qualità nel 24,5% dei casi
- Difese che invecchiano male: detector, euristiche visive, formazione basata sul sospetto
- L'attaccante può testare offline finché il suo deepfake non passa il detector
- Difese che reggono: parola d'ordine fuori banda, callback su numero da rubrica interna
- Doppia autorizzazione sopra soglia; firma crittografica dei contenuti (C2PA)

PromptLock: malware senza payload

- Ago 2025, ESET — scritto in Go, incorpora prompt fissi, non logica malevola
- Interroga un LLM locale via Ollama, che genera al volo script Lua
- Gli script enumerano il filesystem, decidono cosa esfiltrare e cosa cifrare, ed eseguono
- È un proof-of-concept accademico (NYU Tandon) trovato su VirusTotal: nessuna vittima
- Conta lo stesso: il codice malevolo non esiste finché non è generato sulla macchina della vittima
- Corollario difensivo: rilevare comportamenti batte rilevare artefatti

Il malware che itera contro l'EDR

- Set 2026, report Anthropic — sospetto attore di spionaggio russo
- Agent che sorvegliano i prodotti di sicurezza in cerca di rilevamenti del proprio malware
- A ogni rilevamento l'agent riscrive il malware e riprova, finché non viene più rilevato
- È un loop di ottimizzazione avversaria: genera, testa, osserva, muta — senza addestrare nulla
- L'EDR funge da funzione di fitness: espone un oracolo binario interrogabile a costo nullo
- Ogni difesa che dà feedback osservabile all'attaccante è ottimizzabile contro

Cosa l'AI non cambia

- L'accesso iniziale passa ancora in larga parte da credenziali e software non aggiornato
- MFA resistente al phishing (FIDO2 / WebAuthn) blocca ancora intere classi di attacco
- Least privilege e segmentazione limitano il danno, chiunque guidi l'attacco
- È cambiata la scala: un operatore conduce trenta intrusioni in parallelo
- È cambiata la velocità, e la soglia di competenza richiesta per un exploit funzionante
- L'AI ha reso conveniente attaccare bersagli che prima non valevano lo sforzo

Parte 2 AI difensiva

- Gli stessi progressi, dall'altra parte
- Distinguendo i risultati misurati dalle promesse dei vendor

AlxCC: trovare e riparare da soli

- Finale ago 2025 — sette Cyber Reasoning Systems, 143 ore di funzionamento totalmente autonomo
- 53 progetti derivati da software di infrastruttura critica
- 54 vulnerabilità sintetiche scoperte su 63 challenge, 43 riparate con patch
- 18 vulnerabilità reali scoperte incidentalmente (6 in C, 12 in Java), con responsible disclosure
- Team Atlanta primo, Trail of Bits secondo, Theori terzo; sistemi rilasciati open source
- Il dettaglio che conta: la patch è accettata solo se supera i test — l'LLM propone, il verificatore dispone

Big Sleep e CodeMender

- Big Sleep (Google Project Zero + DeepMind) — agent di vulnerability research
- CVE-2025-6965 in SQLite: falla nota solo agli attaccanti, chiusa prima dello sfruttamento
- Primo caso rivendicato in cui un agent AI sventa uno sfruttamento in preparazione
- CodeMender (DeepMind) — propone la patch, non solo il report
- Riscrive il codice per rendere strutturalmente impossibile la classe di vulnerabilità
- Nel 2026 le CVE su Chrome esplodono: è migliorato lo strumento di misura, non peggiorato Chrome

Claude Mythos (aprile 2026)

- Classe di modelli sopra Opus, non rilasciata al pubblico
- Motivazione dichiarata: la capacità stessa di trovare vulnerabilità
- CVE-2026-4747: RCE in FreeBSD, root via NFS — bug rimasto 17 anni
- Bug di 27 anni in OpenBSD, di 16 anni nel codec H.264 di FFmpeg
- Su browser: un exploit concatena 4 vulnerabilità ed esce da due sandbox
- 181 exploit funzionanti su una vulnerabilità Firefox, contro i 2 di Opus 4.6
- Costo di un run che produce un exploit funzionante: meno di 50 dollari
- Oltre il 99% delle vulnerabilità trovate non era ancora patchata
- Fonte unica: Anthropic, insieme produttore, valutatore e venditore

Project Glasswing

- Niente rilascio pubblico: l'accesso va a una coalizione di difensori
- 12 partner: AWS, Apple, Google, Microsoft, NVIDIA, CrowdStrike, Palo Alto
- Più la Linux Foundation, e oltre 40 organizzazioni di infrastruttura critica
- 100 M\$ in usage credit; 4 M\$ a Alpha-Omega, OpenSSF, Apache Foundation
- A listino dopo la preview: 25\$/125\$ per milione di token
- La tesi: i difensori arrivano per primi sul software critico
- La domanda: chi decide chi entra? Governance privata, senza mandato pubblico
- Chi resta fuori riceve le patch, non lo strumento — e le riceve dopo

Trovare non è riparare

- Primo mese di Glasswing: oltre 10.000 vulnerabilità high/critical trovate
- Cloudflare: 2.000 bug, 400 high/critical, falsi positivi sotto quelli umani
- Mozilla: 271 vulnerabilità in Firefox 150
- Validazione indipendente: 90,6% di veri positivi su 1.752 casi esaminati
- Ma solo 75 patch su 530 bug high/critical divulgati
- Due settimane in media per patchare; i manutentori non hanno capacità
- La scoperta scala, la riparazione no: è il nuovo collo di bottiglia
- Ogni falla trovata e non chiusa è una vulnerabilità nota e documentata

Fable e Mythos: due accessi

- Fable 5 e Mythos 5 (giugno 2026) sono lo stesso modello
- A cambiare non sono i pesi, ma i safeguard attivi
- Fable è generalmente disponibile; Mythos è ad accesso ristretto
- Con la 5.1: Fable può trovare vulnerabilità, non svilupparne l'exploit
- Controllo degli accessi applicato a una capacità, non a un oggetto
- Tre incidenti nelle valutazioni interne di Anthropic:
- Test con accesso a Internet non previsto: compromessa un'azienda reale
- Pacchetto Python malevolo su PyPI, eseguito su ~15 sistemi reali
- Scansione di ~9.000 bersagli reali e SQL injection su un'app in produzione
- Confinare un agent è un problema di isolamento, non di prompt

XBOW: una macchina in classifica

- Giu 2025 — primo sistema autonomo al 1° posto della leaderboard USA di HackerOne
- Circa 1.060 report di vulnerabilità su programmi reali in novanta giorni
- Ma i bug bounty premiano volume e velocità su vulnerabilità web riconoscibili per pattern
- Non dimostra superiorità su logica di business o contesti che richiedono comprensione del dominio
- Il dato “85 volte più veloce di un pentester” è del fornitore, non di terzi indipendenti
- Effetto collaterale: programmi di bug bounty sommersi da report automatici di bassa qualità

Il SOC agentico

- Il problema del SOC non è l'intelligenza ma il volume: migliaia di alert al giorno, quasi tutti falsi positivi
- L'analista di primo livello si assuefa e chiude in fretta: è così che passano gli incidenti veri
- Il triage è lavoro ripetitivo e contestuale: il fit con un LLM è genuino
- Piattaforme 2026: Microsoft Security Copilot, CrowdStrike Charlotte AI, SentinelOne Purple AI
- I vendor dichiarano triage oltre il 95% e MTTR ridotto del 50–80%; le misure indipendenti sono più modeste

Domanda 1: chi avvantaggia l'AI?

- Attacco: basta un successo, e l'AI aumenta i tentativi per unità di costo
- Attacco: deployment istantaneo — nessun procurement, change management o compliance
- Attacco: gli errori costano poco, mentre la difesa deve fidarsi dell'output per agire
- Difesa: il collo di bottiglia è lavoro noioso su scala, ciò che l'AI fa meglio
- Difesa: riparare una vulnerabilità la chiude per tutti gli attaccanti, per sempre
- Difesa: si può verificare in modo deterministico — la patch passa i test o no

Parte 3 Difendere l'AI

- La parte che vi riguarda più da vicino
- Sarete voi a costruire i sistemi che questi attacchi colpiscono

OWASP Top 10 for LLM Apps (2025)

- LLM01 Prompt Injection — primo posto per la seconda edizione consecutiva
- LLM02 Sensitive Information Disclosure · LLM03 Supply Chain · LLM04 Data and Model Poisoning
- LLM05 Improper Output Handling · LLM06 Excessive Agency · LLM07 System Prompt Leakage
- LLM08 Vector and Embedding Weaknesses · LLM09 Misinformation · LLM10 Unbounded Consumption
- LLM01 è la causa: il modello non separa istruzioni e dati
- LLM06 è l'impatto: senza agency un'injection produce testo sbagliato, con agency produce azioni

Prompt injection non è SQL injection

- SQL injection è risolta: i prepared statement danno due canali, codice e dati
- Il parser non può promuovere un dato a istruzione: la classe scompare
- Nella prompt injection istruzioni e dati arrivano nello stesso canale: il testo
- Non esiste l'equivalente del prepared statement, perché non esiste un secondo canale
- Ogni mitigazione — delimitatori, classificatori, prompt severi — è probabilistica
- È il problema di von Neumann: aspettatevi mitigazioni, non soluzioni

Injection diretta e indiretta

- Diretta: l'utente scrive "ignora le istruzioni precedenti" — attacca una sessione propria
- Indiretta: il payload è nel contenuto che il modello legge — email, pagina web, PDF, ticket
- Scenario: un assistente ha accesso alla posta e a un tool invia_messaggio
- Una email con testo bianco su bianco gli ordina di cercare allegati e inviarli a un terzo
- L'utente chiede solo "riassumi la posta": per il modello i due testi sono indistinguibili
- Nessun exploit, nessuna credenziale rubata, nessun controllo di accesso violato

EchoLeak: zero-click su Copilot

- CVE-2025-32711, giu 2025, CVSS 9,3 — esfiltrazione remota da una singola email
- Zero-click: basta che Copilot includa l'email nel contesto, la vittima non deve aprirla
- Payload in un commento HTML o testo bianco su bianco; superficie Word, Excel, Outlook, Teams
- Catena: evasione del classificatore XPIA, aggiramento della redazione dei link
- Poi immagini auto-caricate come canale di uscita e abuso di un proxy Teams già in CSP
- Quattro difese aggirate: la difesa in profondità fallisce se i livelli condividono l'assunzione errata

La “lethal trifecta”

- Dati privati — l'agent legge email, documenti, database, repository interni
- Contenuto non fidato — l'agent elabora web, email in arrivo, file condivisi, ticket
- Canale di uscita — l'agent può emettere dati via HTTP, immagini, link, messaggi
- Se ci sono tutti e tre, il sistema è compromettibile per costruzione: è l'architettura, non il modello
- In EchoLeak: la casella dell'utente, l'email dell'attaccante, l'immagine auto-caricata
- Difendersi significa rimuovere un lato: isolare i dati, vincolare l'uscita, o approvazione umana

MCP: il tool stesso è il payload

- In MCP ogni tool si presenta al modello con una description testuale
- Quel campo non è sanificato e finisce nel contesto: un server può nascondervi istruzioni
- L'agent le segue senza che l'utente veda nulla: è il tool poisoning
- Classi correlate: rug pull, tool shadowing, attacchi cross-server, confused deputy su OAuth
- Oltre 40 CVE contro implementazioni MCP fra gennaio e aprile 2026, su SDK Python, TS, Java, Rust
- Giu 2026: la NSA pubblica una guida MCP Security Design

La SOP non vale per un agente

- Studio dell'Università di Washington: su sette browser agentici, in quattro un sito può aggirare la same-origin policy
- Brave su Comet: istruzione nascosta in uno spoiler tag di Reddit
- Alla richiesta “riassumi la pagina” l'agente legge la posta, preleva un OTP e lo pubblica
- LayerX: un singolo link fa raccogliere dati dalle app connesse e li invia all'attaccante
- La SOP isola origini e non è rotta: ma l'agente non è uno script, è un utente
- Opera sopra il livello della SOP, con le sessioni di tutte le origini già aperte

Poisoning: bastano 250 documenti

- Ott 2025 — Anthropic, UK AI Security Institute e Alan Turing Institute: lo studio più ampio finora
- Modelli da 600 milioni a 13 miliardi di parametri, con 100, 250 o 500 documenti malevoli
- 250 documenti bastano a impiantare una backdoor, indipendentemente dalla dimensione del modello
- Si assumeva servisse una percentuale dei dati: per un modello grande sarebbe stato proibitivo
- Se la soglia è assoluta e piccola, l'attacco è alla portata di chiunque: 250 pagine web
- La provenienza dei dati di training è supply chain security, non data science

Slopsquatting

- Su 2,23 milioni di campioni di codice generato da 16 modelli, il 19,7% cita un pacchetto inesistente
- Il 43% di quei nomi si ripete a ogni esecuzione: l'attaccante può prevederli
- Li pre-registra su npm o PyPI con codice malevolo
- È typosquatting senza typo: il pacchetto lo sceglie l'AI, non l'utente che sbaglia a digitare
- Veracode su oltre 100 LLM: il 45% del codice generato introduce vulnerabilità OWASP Top 10
- Difesa: verificare che il pacchetto esista da prima della vostra domanda; lockfile, allowlist, SBOM

Agent sicuri: principi

- Least privilege sui tool — chi riassume la posta non deve poter inviare messaggi
- Credenziali con scope ristretto, mai il token dell'utente
- Mediazione completa — ogni tool call passa da un policy engine esterno al modello
- Action gating — approvazione umana o allowlist per ogni azione irreversibile o verso l'esterno
- Separazione dei contesti (dual-LLM, CaMeL) — chi ha privilegi non vede mai contenuto non fidato
- Sono i principi di Saltzer e Schroeder (1975), applicati a un componente non deterministico

Agent sicuri: controlli operativi

- Sandboxing: container effimero, filesystem read-only, egress filtering
- L'egress filtering è il terzo lato della trifecta: toglietelo e la catena si rompe
- Audit log di ogni prompt, tool call e risultato: è la vostra unica forensics
- Rate limiting e budget: un agent in loop è anche un incidente di costo
- Red teaming continuo nel CI, non una valutazione una tantum
- Provenienza di modelli, dataset e server MCP: pinning, niente "ultima disponibile"

Parte 4 Strumenti

- Cosa potete installare e provare questa settimana
- Tutto open source o con tier gratuito

Toolbox

- garak (NVIDIA) — scanner per LLM, oltre 120 probe per jailbreak, injection, leakage. Il più facile
- PyRIT (Microsoft) — orchestrazione di red teaming: attacchi adattivi, converter, scoring
- promptfoo — valutazione e red teaming da CLI, integrabile nel CI
- Cyber Reasoning Systems di AlxCC — fuzzing guidato da LLM e patch automatica, da leggere
- Gandalf / Lakera — CTF didattico sulla prompt injection, ottimo primo contatto in aula
- OWASP GenAI e MITRE ATLAS — i riferimenti; Ollama per eseguire un LLM in locale

Laboratorio: tre esercizi, tre ore

- Rompere (40 min) — Gandalf fino al livello 7, documentando quale difesa c'era e come è caduta
- Obiettivo: capire che ogni difesa nota ha un bypass noto
- Misurare (50 min) — Ollama con un modello piccolo, poi garak; confrontare due modelli
- Obiettivo: la sicurezza di un LLM è misurabile, non opinabile
- Costruire (90 min) — mini-agent con leggi_file e http_get, bucarlo, poi ripararlo
- Obiettivo: si ripara l'architettura (egress allowlist, approvazione), non il prompt

Parte 5 Governance e futuro

- Cosa dice la legge, cosa dicono gli analisti
- e cosa penso io che succederà davvero

EU AI Act: le date che contano

- 2 ago 2025 — obblighi per i GPAI, i modelli di uso generale
- Per i GPAI con rischio sistemico: mitigazione dei rischi, model evaluation, segnalazione incidenti, cybersecurity del modello
- 2 ago 2026 — obblighi di trasparenza (art. 50); l'AI Office può esercitare vigilanza e sanzione
- 2 dic 2027 — regole sui sistemi ad alto rischio, rinviate dall'AI Omnibus
- 2 ago 2028 — AI incorporata in prodotti regolati da altra normativa settoriale
- Da conoscere anche: NIST AI RMF, ISO/IEC 42001, Cyber Resilience Act

Cosa dicono gli analisti

- 2027 — gli agent AI dimezzano il tempo per sfruttare l'esposizione di un account
- 2027 — oltre il 40% dei progetti agentic AI sarà cancellato: costi e controlli inadeguati
- 2027 — la compliance AI manuale esporrà a sanzioni il 75% delle organizzazioni regolate
- 2028 — le applicazioni AI guideranno il 50% delle attività di incident response
- 2028 — il 40% del social engineering userà deepfake audio e video
- Tutte Gartner: sono proiezioni di spesa, e tendono ad autoavverarsi perché orientano i budget

Cinque scommesse, al 2030

- La prompt injection non verrà risolta: sarà gestita per architettura, come la memoria in C
- Il patching automatico verificato arriva prima dell'attacco automatico end-to-end
- Sparirà l'analista di livello 1 — non il mestiere, il livello: si passerà a supervisionare agent
- L'identità delle macchine diventa centrale: "chi ha compiuto questa azione" non ha più risposta
- Il primo incidente sistemico nascerà da un'integrazione, non da un modello
- Hype: l'AI che si difende da sola, la detection di deepfake, il system prompt come controllo

Sette cose da portare a casa

- L'AI non ha creato nuove classi di attacco: ha abbattuto il costo
- Gli agent sono il salto: la sicurezza sta nel runtime, non nel modello
- La prompt injection è architetturale: istruzioni e dati condividono un solo canale
- La lethal trifecta è la checklist: dati, contenuto non fidato, uscita
- In difesa i risultati verificabili esistono: chiedete sempre la metodologia
- La supply chain resta il vettore sistemico: training, MCP, pacchetti
- I fondamentali delle Lezioni 1–3 reggono anche sotto un componente non deterministico