Web Agents — Vulnerabilities & Defenses

Agenda

- Threat Model & Attack Surface
- Attacks in Practice (Case Studies)
- Risk Taxonomy & Prioritization
- Defenses: Principles & Controls
- Testing, Red Teaming & Monitoring
- Exercise & Discussion

Recap from Lecture 1

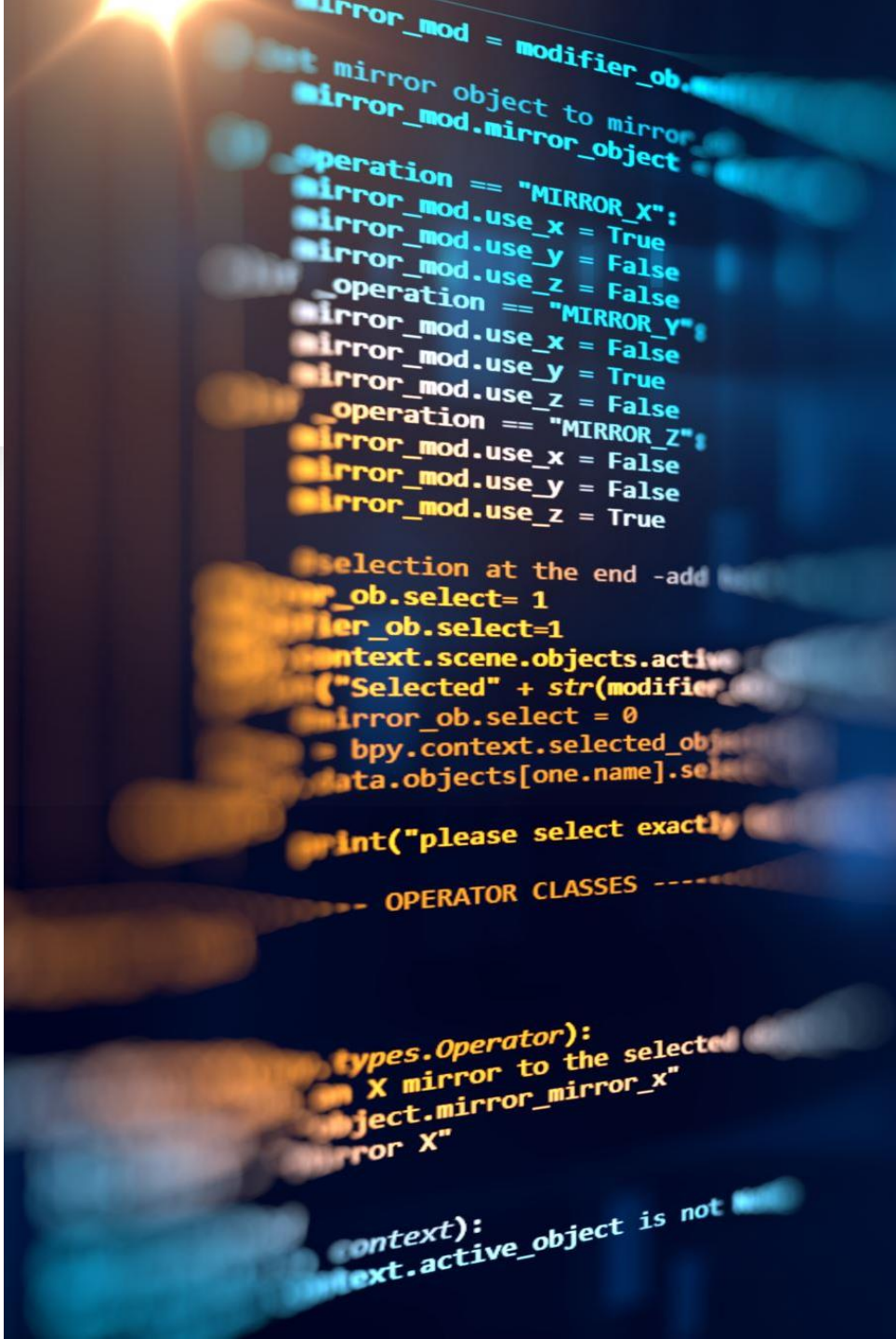
- Agents perceive → plan → act in a browser environment
- Tools and memory expand capability—and risk
- Evaluation helps avoid regressions; security must be part of eval

THREAT MODEL & ATTACK SURFACE

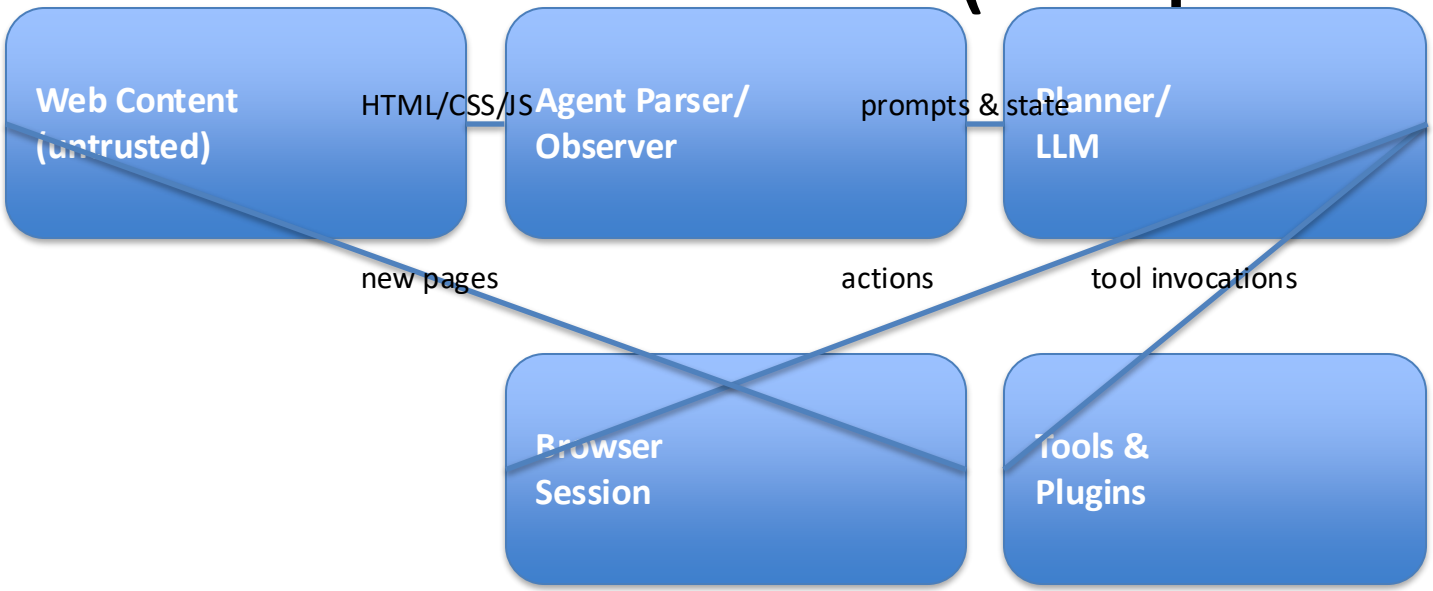
What can go wrong
and why

Threat Model (High Level)

- Inputs from untrusted pages: HTML, JS, CSS, images, PDFs, comments
- Agents often hold high privileges: cookies, tokens, logged-in sessions
- Tool access expands impact: code exec, file I/O, email, APIs
- Adversary goals: steer, exfiltrate, escalate, persist



Attack Surface (Simplified)



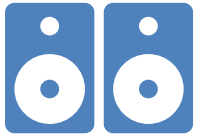
ATTACKS IN PRACTICE

How agents are manipulated

Prompt & Indirect Prompt Injection

Direct: adversarial text instructs the agent to deviate from task

Indirect: instructions embedded in web content (ads, comments, PDFs)



Amplifiers: model jailbreaks, overly broad tools, missing validations

UI/DOM Bait & Traps

Hidden/overlayed elements that capture clicks or inputs

Vision bait: pixels that look benign to humans but lure agents

Dynamic consent dialogs/popups creating race conditions

Insecure Output Handling & Tool Misuse

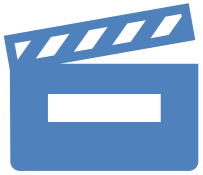
Agent treats untrusted output as code, SQL, or shell

SSRF via tools that fetch URLs provided by content

Privilege escalation via powerful tools without scope limits

Classic Web Vulns through Agent Behavior

CSRF-like effects: agent performs state-changing actions cross-origin



Clickjacking-like flows: framed content steering agent actions

Phishing via autofill tendencies and heuristics

Case Studies (High-level)

Black Hat EU 2023 — Indirect Prompt Injection via files/images

Black Hat USA 2025 — From Prompts to Pwns (agent exploitation demos)

Red Teaming Browser Agents — refusal-trained models still jailbroken

RISK TAXONOMY & PRIORITIZATION

Mapping to OWASP LLM Top 10 (Agent Lens)

OWASP Item	Agent-Specific Example	Primary Mitigation
LLM01 Prompt Injection	Content instructs agent to exfiltrate data	Input isolation; deny-rules; approvals
LLM02 Insecure Output Handling	Treats model output as SQL/JS	Strict tool schemas; sanitization
LLM06 Sensitive Info Disclosure	Agent reveals secrets via chat logs	Secret scanning; redact; compartmentalize
LLM08 Excessive Agency	Powerful tools without scope limits	Least privilege; scoped credentials
LLM10 Monitoring & Logging	No traces for investigations	Structured logs; replay; alerts

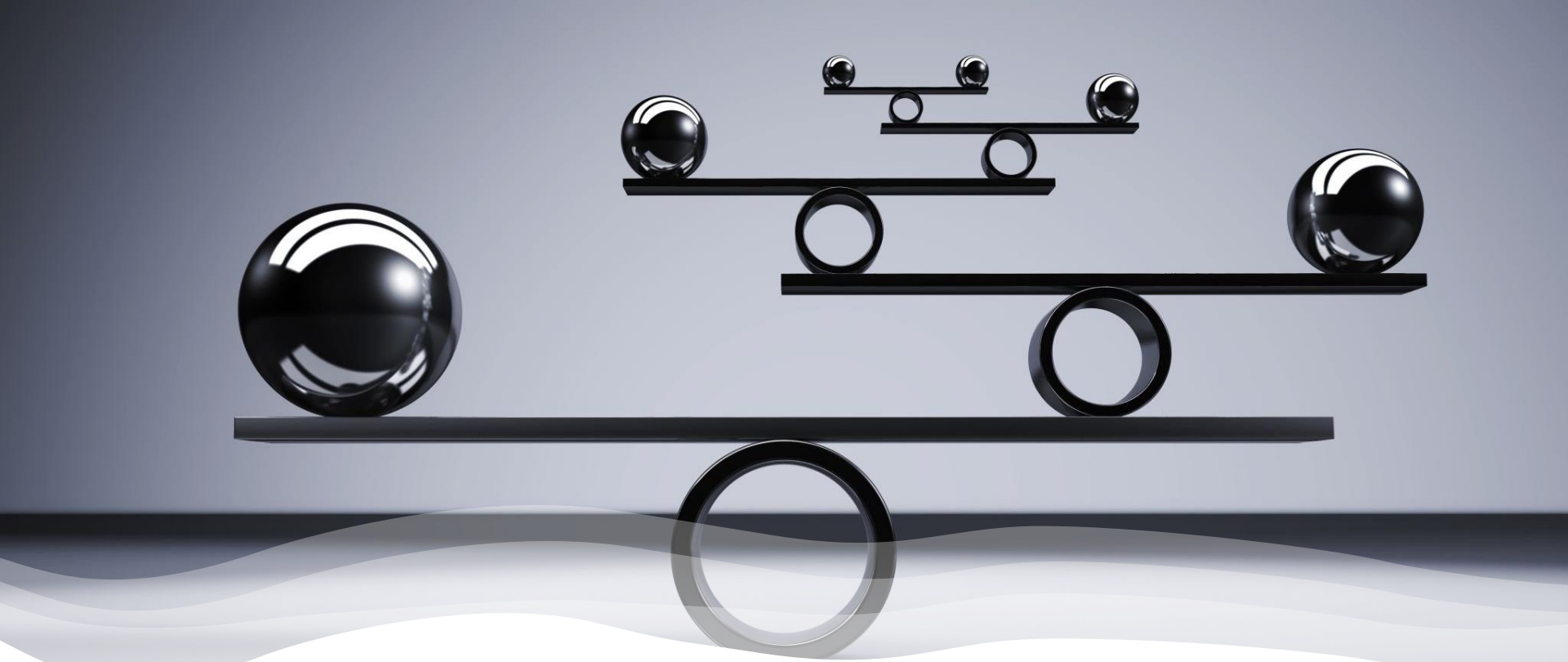
Prioritizing Risks

Likelihood × Impact scoring for each capability and site/domain

Identify blast radius: what can this agent change or access?

Reduce privileges until safe (then add approvals)

Define ‘never’ and ‘always ask’ categories



DEFENSES: PRINCIPLES & CONTROLS

Security Principles for Agents

- Defense-in-depth across observation, planning, and execution
- Least privilege & compartmentalization of credentials/sessions
- Complete mediation: policy gate for every sensitive action
- Secure defaults; explicit opt-in for destructive actions

Input & Prompt Isolation

Separate	Separate untrusted content from system/tool prompts
Annotate	Annotate tainted spans and constrain how they influence tools
Use	Use structured queries (not free-form text) for tools

Action Controls & Approvals

- Allowlist selectors and domains for state-changing actions
- Per-action confirmation with summaries (‘what will happen’)
- Rate limits, budgets, and timeouts to cap damage

Credential & Session Management

- Scoped, short-lived tokens; per-domain storage
- Read-only vs. read-write session separation
- Zero standing privileges outside active tasks

Sandboxing & Containment

- Execute transforms in sandboxes; block network by default
- Out-of-process browser controllers; hardened containers
- File/clipboard permission prompts and quotas

Testing, Red Teaming & Monitoring

- Adversarial test sets (injections, UI traps, consent flows)
- Replay traces; DOM diffs; anomaly alerts on sensitive actions
- Incident response runbooks and kill-switches

EXERCISE & DISCUSSION

Exercise (10–15 minutes)

You maintain a shopping assistant agent with login to two vendors

Identify high-risk actions and where to add approvals

Propose credential scoping and session separation

Define logging you would need to investigate incidents

Discussion Prompts

Where should humans be in-the-loop?

How do we measure ‘safe enough to ship’ for agents?

What telemetry would have prevented your last incident?

Summary & Takeaways

- Agents expand what's possible—and the attack surface
- Treat web content as untrusted; isolate and mediate
- Default to least privilege with explicit approvals
- Continuously test with adversarial scenarios

References (1/2)

- Black Hat EU 2023 — Indirect Prompt Injection (slides): https://i.blackhat.com/EU-23/Presentations/EU-23-Nassi-IndirectPromptInjection.pdf
- Black Hat USA 2025 — From Prompts to Pwns: https://i.blackhat.com/BH-USA-25/Presentations/US-25-Lynch-From-Prompts-to-Pwns.pdf
- AWS re:Invent 2024 — Safeguard GenAI from Prompt Injections: https://reinvent.awsevents.com/content/dam/reinvent/2024/slides/sec/SEC338_Safeguard-your-generative-AI-apps-from-prompt-injections.pdf
- Scale/BrowserART — Red Teaming Browser Agents (paper slides): https://static.scale.com/uploads/6691558a94899f2f65a87a75/_ICLR__Red_Teaming_Browser_Agents%20(14).pdf

References (2/2)

- OWASP Top 10 for LLM Applications: https://genai.owasp.org/llm-top-10/
- Language Agent Tutorial — Apps/Data/Eval (OWASP mapping): https://language-agent-tutorial.github.io/slides/III-Applications-Data-Evaluation.pdf
- LLM Agents Should Employ Security Principles (AgentSandbox): https://kaiyuanzhang.com/slides/LLM_Agents_Kaiyuan_Zhang_2025.pdf