Policy Framings for Access Control

Massimo Bartoletti, Pierpaolo Degano, Gian Luigi Ferrari

Dipartimento di Informatica, Università di Pisa

Stack Inspection

- access control mechanism based on the analysis of the execution stack (stack of method frames)
- classes are assigned to *protection domains*, which are granted permissions according to a security policy.
- to check if a permission P is granted to an execution:

```
for each frame in the call stack (starting from top)
    if P is not granted to the frame
        throw an AccessControlException
```

Overview of our approach

- Java Bytecode $\xrightarrow{CFA}$ Control Flow Graph
- control flow + security checks + (dynamic) policy
- basic data flow (e.g. permissions)
- if..then..else.. and dynamic dispatching rendered as nondeterministic choiche
- static analysis to compute *access control contexts*

The language $\lambda^{[]}$ (syntax)

$e, e' \quad ::=$

x	variable
α	access event
$\lambda_z x.\, e$	abstraction
$e\, e'$	application
$\varphi[e]$	policy framing
$\texttt{if}\, b\, \texttt{then}\, e\, \texttt{else}\, e'$	conditional

$\lambda.\, e = \lambda_z x.\, e \qquad (z, x \notin fv(e))$

$e; e' = (\lambda.\, e')e$

The language $\lambda^{[]}$ (semantics)

$$\frac{\eta, e_1 \rightarrow \eta', e_1'}{\eta, e_1\, e_2 \rightarrow \eta', e_1'\, e_2}$$

$$\frac{\eta, e_2 \rightarrow \eta', e_2'}{\eta, v\, e_2 \rightarrow \eta', v\, e_2'}$$

$$\eta, (\lambda_z x.e)v \rightarrow \eta, e\{v/x, \lambda_z x.e/z\}$$

The language $\lambda^{[]}$ (semantics)

$$\frac{}{\eta, \alpha \longrightarrow \eta\alpha, *}$$

$$\frac{\eta, e \longrightarrow \eta', e' \quad \eta, \eta' \models \varphi}{\eta, \varphi[e] \longrightarrow \eta', \varphi[e']}$$

$$\frac{\eta \models \varphi}{\eta, \varphi[v] \longrightarrow \eta, v}$$

Programming in $\lambda^{[]}$ (1)

- $Browser = \lambda u.\, \lambda p.\, \texttt{if}\ html(u)$
  $\texttt{then}\ display(u)$
  $\texttt{else if}\ trusted(u)\ \texttt{then}\ p\ u$
  $\texttt{else}\ \varphi[p\ u;\ W\ *]$
- $Untrusted\ =\ \lambda z.\, \lambda.\, Browser\ z\ (\lambda y.\, y*)$
- $R = \lambda.\, \alpha_r$ (file read)
- $W = \lambda.\, \alpha_w$ (file write)
- $C = \lambda.\, \alpha_c$ (socket connection)
- $\varphi =$ “no connect after a read”
- $\varphi' =$ “no write”

Programming in $\lambda^{[]}$ (2)

$$\varepsilon, \mathit{Browser}\,(\mathit{Untrusted}\ W)\,(\lambda y.\varphi'[y*])$$

Programming in $\lambda^{[]}$ (2)

$$\varepsilon, \mathit{Browser}\,(\mathit{Untrusted}\ W)\,(\lambda y.\varphi'[y*])$$

$$\longrightarrow\ \varepsilon, \varphi[\lambda y.\varphi'[y*]\,(\lambda.\,\mathit{Browser}\,W\ \lambda y.\,y*);\ W*]$$

Programming in $\lambda^{[]}$ (2)

$$
\begin{aligned}
& \varepsilon, \mathit{Browser}\,(\mathit{Untrusted}\ W)\,(\lambda y.\varphi'[y*]) \\
\longrightarrow\ & \varepsilon, \varphi[\lambda y.\varphi'[y*]\,(\lambda.\,\mathit{Browser}\,W\ \lambda y.\,y*);\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[\mathit{Browser}\ W\ (\lambda y.\,y*)];\ W*]
\end{aligned}
$$

Programming in $\lambda^{[]}$ (2)

$$
\begin{aligned}
& \varepsilon, \mathit{Browser}\,(\mathit{Untrusted}\ W)\,(\lambda y.\varphi'[y*]) \\
\longrightarrow\ & \varepsilon, \varphi[\lambda y.\varphi'[y*]\,(\lambda.\,\mathit{Browser}\,W\ \lambda y.\,y*);\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[\mathit{Browser}\ W\ (\lambda y.\,y*)];\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[(\lambda y.\,y*)\,W];\ W*]
\end{aligned}
$$

Programming in $\lambda^{[]}$ (2)

$$
\begin{aligned}
& \varepsilon, \mathit{Browser}\,(\mathit{Untrusted}\ W)\,(\lambda y.\varphi'[y*]) \\
\longrightarrow\ & \varepsilon, \varphi[\lambda y.\varphi'[y*]\,(\lambda.\,\mathit{Browser}\,W\ \lambda y.\,y*);\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[\mathit{Browser}\ W\ (\lambda y.\,y*)];\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[(\lambda y.\,y*)\,W];\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[\alpha_w];\ W*]
\end{aligned}
$$

Programming in $\lambda^{[]}$ (2)

$$\varepsilon, \mathit{Browser}\,(\mathit{Untrusted}\ W)\,(\lambda y.\varphi'[y*])$$

$$\longrightarrow \varepsilon, \varphi[\lambda y.\varphi'[y*]\,(\lambda.\,\mathit{Browser}\,W\ \lambda y.\,y*);\ W*]$$

$$\longrightarrow \varepsilon, \varphi[\varphi'[\mathit{Browser}\ W\ (\lambda y.\,y*)];\ W*]$$

$$\longrightarrow \varepsilon, \varphi[\varphi'[(\lambda y.\,y*)\,W];\ W*]$$

$$\longrightarrow \varepsilon, \varphi[\varphi'[\alpha_w];\ W*]$$

$$\alpha_w \not\models \varphi' \implies \text{security exception!}$$

Programming in $\lambda^{[]}$ (3)

$$\varepsilon, \mathit{Browser}\,(\mathit{Untrusted}\,(\lambda.\, R*;\, C*))\,(\lambda y.\varphi'[y*])$$

Programming in $\lambda^{[]}$ (3)

$$\varepsilon,\ \mathit{Browser}\ (\mathit{Untrusted}\ (\lambda.\ R*;\ C*))\ (\lambda y.\varphi'[y*])$$

$$\rightarrow\ \varepsilon,\ \varphi[\lambda y.\varphi'[y*]\ (\mathit{Untrusted}\ (\lambda.\ R*;\ C*));\ W*]$$

Programming in $\lambda^{[]}$ (3)

$$\begin{aligned}
& \varepsilon,\, \mathit{Browser}\,(\mathit{Untrusted}\,(\lambda.\, R*;\, C*))\,(\lambda y.\varphi'[y*]) \\
\longrightarrow\ & \varepsilon,\ \varphi[\lambda y.\varphi'[y*]\,(\mathit{Untrusted}\,(\lambda.\, R*;\, C*));\, W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[\mathit{Browser}\,(\lambda.\, R*;\, C*)\,(\lambda y.\, y*)];\, W*]
\end{aligned}$$

Programming in $\lambda^{[]}$ (3)

$$
\begin{aligned}
& \varepsilon,\ \mathit{Browser}\,(\mathit{Untrusted}\,(\lambda.\ R*;\ C*))\,(\lambda y.\varphi'[y*]) \\
\longrightarrow\ & \varepsilon,\ \varphi[\lambda y.\varphi'[y*]\,(\mathit{Untrusted}\,(\lambda.\ R*;\ C*));\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[\mathit{Browser}\,(\lambda.\ R*;\ C*)\,(\lambda y.\ y*)];\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[R*;\ C*];\ W*]
\end{aligned}
$$

Programming in $\lambda^{[]}$ (3)

$$
\begin{aligned}
& \varepsilon,\ \mathit{Browser}\,(\mathit{Untrusted}\,(\lambda.\ R*;\ C*))\,(\lambda y.\varphi'[y*]) \\
\longrightarrow\ & \varepsilon,\ \varphi[\lambda y.\varphi'[y*]\,(\mathit{Untrusted}\,(\lambda.\ R*;\ C*));\ W*] \\
\longrightarrow\ & \varepsilon,\ \varphi[\varphi'[\mathit{Browser}\,(\lambda.\ R*;\ C*)\,(\lambda y.\ y*)];\ W*] \\
\longrightarrow\ & \varepsilon,\ \varphi[\varphi'[R*;\ C*];\ W*] \\
\longrightarrow\ & \alpha_r,\ \varphi[\varphi'[C*];\ W*]
\end{aligned}
$$

Programming in $\lambda^{[]}$ (3)

$$\varepsilon,\ Browser\ (Untrusted\ (\lambda.\ R*;\ C*))\ (\lambda y.\varphi'[y*])$$

$$\rightarrow\ \varepsilon,\ \varphi[\lambda y.\varphi'[y*]\ (Untrusted\ (\lambda.\ R*;\ C*));\ W*]$$

$$\rightarrow\ \varepsilon,\ \varphi[\varphi'[Browser\ (\lambda.\ R*;\ C*)\ (\lambda y.\ y*)];\ W*]$$

$$\rightarrow\ \varepsilon,\ \varphi[\varphi'[R*;\ C*];\ W*]$$

$$\rightarrow\ \alpha_r,\ \varphi[\varphi'[C*];\ W*]$$

$$\alpha_r\alpha_c \not\models \varphi \implies \text{security exception!}$$

Programming in $\lambda^{[]}$ (4)

$$\varepsilon,\ Browser\,(Untrusted\ R)\,(\lambda y.\varphi'[y*])$$

Programming in $\lambda^{[]}$ (4)

$$\varepsilon,\ \mathit{Browser}\,(\mathit{Untrusted}\ R)\,(\lambda y.\varphi'[y*])$$

$$\longrightarrow\ \varepsilon, \varphi[(\lambda y.\varphi'[y*])\,(\mathit{Untrusted}\ R);\ W*]$$

Programming in $\lambda^{[]}$ (4)

$$\varepsilon,\ Browser\,(Untrusted\ R)\,(\lambda y.\varphi'[y*])$$
$$\rightarrow \varepsilon, \varphi[(\lambda y.\varphi'[y*])\,(Untrusted\ R);\ W*]$$
$$\rightarrow \varepsilon, \varphi[\varphi'[Browser\ R\,(\lambda y.\ y*)];\ W*]$$

Programming in $\lambda^{[]}$ (4)

$$
\begin{aligned}
& \varepsilon,\ \mathit{Browser}\,(\mathit{Untrusted}\ R)\,(\lambda y.\varphi'[y*]) \\
\longrightarrow\ & \varepsilon, \varphi[(\lambda y.\varphi'[y*])\,(\mathit{Untrusted}\ R); W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[\mathit{Browser}\ R\,(\lambda y.\ y*)]; W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[R*]; W*]
\end{aligned}
$$

Programming in $\lambda^{[]}$ (4)

$$
\begin{aligned}
& \varepsilon,\ Browser\,(Untrusted\ R)\,(\lambda y.\varphi'[y*]) \\
\longrightarrow\ & \varepsilon, \varphi[(\lambda y.\varphi'[y*])\,(Untrusted\ R);\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[Browser\ R\,(\lambda y.\ y*)];\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[R*];\ W*] \\
\longrightarrow\ & \alpha_r, \varphi[\varphi'[*];\ W*]
\end{aligned}
$$

Programming in $\lambda^{[]}$ (4)

$$
\begin{aligned}
& \varepsilon,\ \mathit{Browser}\,(\mathit{Untrusted}\ R)\,(\lambda y.\varphi'[y*]) \\
\longrightarrow\ & \varepsilon, \varphi[(\lambda y.\varphi'[y*])\,(\mathit{Untrusted}\ R);\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[\mathit{Browser}\ R\,(\lambda y.\ y*)];\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[R*];\ W*] \\
\longrightarrow\ & \alpha_r, \varphi[\varphi'[*];\ W*] \\
\longrightarrow\ & \alpha_r, \varphi[\,W*]
\end{aligned}
$$

Programming in $\lambda^{[]}$ (4)

$$
\begin{aligned}
& \varepsilon,\ \mathit{Browser}\,(\mathit{Untrusted}\ R)\,(\lambda y.\varphi'[y*]) \\
\longrightarrow\ & \varepsilon, \varphi[(\lambda y.\varphi'[y*])\,(\mathit{Untrusted}\ R);\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[\mathit{Browser}\ R\,(\lambda y.\ y*)];\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[R*];\ W*] \\
\longrightarrow\ & \alpha_r, \varphi[\varphi'[*];\ W*] \\
\longrightarrow\ & \alpha_r, \varphi[\,W*] \\
\longrightarrow\ & \alpha_r\alpha_w, \varphi[*]
\end{aligned}
$$

Programming in $\lambda^{[]}$ (4)

$$
\begin{aligned}
& \varepsilon,\ \mathit{Browser}\,(\mathit{Untrusted}\ R)\,(\lambda y.\varphi'[y*]) \\
\longrightarrow\ & \varepsilon, \varphi[(\lambda y.\varphi'[y*])\,(\mathit{Untrusted}\ R);\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[\mathit{Browser}\ R\,(\lambda y.\ y*)];\ W*] \\
\longrightarrow\ & \varepsilon, \varphi[\varphi'[R*];\ W*] \\
\longrightarrow\ & \alpha_r, \varphi[\varphi'[*];\ W*] \\
\longrightarrow\ & \alpha_r, \varphi[\,W*] \\
\longrightarrow\ & \alpha_r\alpha_w, \varphi[*] \\
\longrightarrow\ & \alpha_r\alpha_w, *
\end{aligned}
$$

History Expressions

- `if` b `then` α `else` α' $\qquad\qquad \alpha + \alpha'$

History Expressions

- $\texttt{if}\ b\ \texttt{then}\ \alpha\ \texttt{else}\ \alpha'$ $\qquad\qquad \alpha + \alpha'$
- $(\lambda x.\, \alpha')\alpha$ $\qquad\qquad \alpha \cdot \alpha'$

History Expressions

- $\texttt{if}\ b\ \texttt{then}\ \alpha\ \texttt{else}\ \alpha'$ $\qquad \alpha + \alpha'$
- $(\lambda x.\, \alpha')\alpha$ $\qquad \alpha \cdot \alpha'$
- $(\lambda y.\, \alpha; \varphi[y*])(\lambda x.\, \varphi'[\alpha'])$ $\qquad \alpha \cdot \varphi[\varphi'[\alpha']]$

History Expressions

- $\texttt{if}\ b\ \texttt{then}\ \alpha\ \texttt{else}\ \alpha'$ $\qquad \alpha + \alpha'$
- $(\lambda x.\, \alpha')\alpha$ $\qquad \alpha \cdot \alpha'$
- $(\lambda y.\, \alpha; \varphi[y*])(\lambda x.\, \varphi'[\alpha'])$ $\qquad \alpha \cdot \varphi[\varphi'[\alpha']]$
- $(\lambda_z x.\, \alpha; z*)*$ $\qquad \mu h.\, \alpha \cdot h$

History Expressions

- $\texttt{if}\ b\ \texttt{then}\ \alpha\ \texttt{else}\ \alpha'$ $\qquad \alpha + \alpha'$
- $(\lambda x.\, \alpha')\alpha$ $\qquad \alpha \cdot \alpha'$
- $(\lambda y.\, \alpha; \varphi[y*])(\lambda x.\, \varphi'[\alpha'])$ $\qquad \alpha \cdot \varphi[\varphi'[\alpha']]$
- $(\lambda_z x.\, \alpha; z*)*$ $\qquad \mu h.\, \alpha \cdot h$
- $(\lambda_z x.\, \texttt{if}\ b\ \texttt{then}\ \alpha\ \texttt{else}\ \texttt{if}\ b'\ \texttt{then}\ zzx\ \texttt{else}\ \varphi[zx])*$
  $\mu h.\, \alpha + h \cdot h + \varphi[h]$

History Expressions (syntax)

$H, H' ::=$

ε	empty
h	variable
α	access event
$H \cdot H'$	sequence
$H + H'$	choice
$\varphi[H]$	policy framing
$\mu h.H$	recursion

History Expressions (semantics)

$$[\![\varepsilon]\!]_\rho = \varepsilon \qquad [\![\alpha]\!]_\rho = \alpha \qquad [\![h]\!]_\rho = \rho(h)$$

$$[\![H \cdot H']\!]_\rho = [\![H]\!]_\rho \, [\![H']\!]_\rho$$

$$[\![H + H']\!]_\rho = [\![H]\!]_\rho \cup [\![H']\!]_\rho$$

$$[\![\varphi[H]]\!]_\rho = \varphi[[\![H]\!]_\rho]$$

$$[\![\mu h.H]\!]_\rho = \bigcup_{n \in \omega} f^n(\emptyset), \quad f(X) = [\![H]\!]_{\rho\{X/h\}}$$

Access Control Histories

- plain histories + framing events $[_\varphi$ and $]_\varphi$
- notation: $[_\varphi \eta \,]_\varphi = \varphi[\eta]$
- validity: $[_\varphi \alpha_r]_\varphi \alpha_c$ valid, $\alpha_r [_\varphi \alpha_c$ not valid
- $\eta^\flat$ discards the framing events in η
- $\Phi(\eta)$ is the set of φ such that $[_\varphi >]_\varphi$ in η
- $\beta_1 \cdots \beta_n$ is valid iff $\beta_1 \cdots \beta_k \models \Phi(\beta_1 \cdots \beta_k)$, for all $k \in 1..n$
- H valid iff each $\eta \in [\![H]\!]$ is valid

Extracting History Expressions

- type & effect system $\Gamma, H \vdash e : \tau$
- types: $\tau ::= unit \mid \tau \xrightarrow{H} \tau'$
- effects: history expressions H
- correctness of history expressions:

$$\varepsilon, e \rightarrow^* \eta, e' \implies \eta \in prefix(\llbracket H \rrbracket^\flat)$$

- type safety:

$$H \text{ valid} \implies e \text{ will not go wrong}$$

Regularizing History Expressions

- $H\downarrow$ has no redundant framings
- $H\downarrow$ valid iff H valid

Conclusions

- stack inspection + dynamic security policies
  - parametric, sound & complete static analysis
  - elimination of redundant checks and dead code
  - strong/weak method inlining & tail call elimination
  - *complete* decision procedure (for the weak version)

A Type and Effect system for $\lambda^{[]}$ (1)

$$\frac{}{\Gamma, \varepsilon \vdash x : \Gamma(x)} \qquad \frac{}{\Gamma, \alpha \vdash \alpha : unit}$$

$$\frac{\Gamma, H \vdash e : \tau}{\Gamma, \varphi[H] \vdash \varphi[e] : \tau} \qquad \frac{\Gamma, H \vdash e : \tau}{\Gamma, H + H' \vdash e : \tau}$$

$$\frac{\Gamma, H \vdash e : \tau \quad \Gamma, H \vdash e' : \tau}{\Gamma, H \vdash \texttt{if}\, b\, \texttt{then}\, e\, \texttt{else}\, e' : \tau}$$

A Type and Effect system for $\lambda^{[]}$ (2)

$$\frac{\Gamma; x : \tau; z : \tau \xrightarrow{H} \tau', H \vdash e : \tau'}{\Gamma, \varepsilon \vdash \lambda_z x.e : \tau \xrightarrow{H} \tau'}$$

$$\frac{\Gamma, H \vdash e : \tau \xrightarrow{H''} \tau' \quad \Gamma, H' \vdash e' : \tau}{\Gamma, H \cdot H' \cdot H'' \vdash ee' : \tau'}$$

A Type and Effect system for $\lambda^{[]}$ (3)

- $e = \lambda_z x.\, b\,?\,\alpha + (b'\,?\,z z x + \varphi[z x])$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, \varepsilon \vdash z : \tau \xrightarrow{H} \tau} \quad \dfrac{}{\Gamma, \varepsilon \vdash x : \tau}}{\Gamma, H \vdash z\, x : \tau}}{\Gamma, H \cdot H \vdash z\, z\, x : \tau}}{\Gamma, H \cdot H + \varphi[H] \vdash z\, z\, x : \tau} \quad \dfrac{\dfrac{\dfrac{}{\Gamma, \varphi[H] \vdash \varphi[z\, x] : \tau}}{\Gamma, H \cdot H + \varphi[H] \vdash \varphi[z\, x] : \tau}}{\Gamma, H \cdot H + \varphi[H] \vdash b'\,?\,z\, z\, x + \varphi[z\, x] : \tau}}{\Gamma, \alpha + H \cdot H + \varphi[H] \vdash b\,?\,\alpha + (b'\,?\,z\, z\, x + \varphi[z\, x]) : \tau}$$

- $H = \alpha + H \cdot H + \varphi[H] \implies H = \mu h.\, \alpha + h \cdot h + \varphi[h]$
- $\emptyset, \varepsilon \vdash e : unit \xrightarrow{\mu h.\, \alpha + h \cdot h + \varphi[h]} unit$

Regularizing history expressions (1)

$$\varepsilon\downarrow_{\Phi,\Gamma} = \varepsilon \qquad h\downarrow_{\Phi,\Gamma} = h \qquad \alpha\downarrow_{\Phi,\Gamma} = \alpha$$

$$(H \cdot H')\downarrow_{\Phi,\Gamma} = H\downarrow_{\Phi,\Gamma} \cdot H'\downarrow_{\Phi,\Gamma}$$

$$(H + H')\downarrow_{\Phi,\Gamma} = H\downarrow_{\Phi,\Gamma} + H'\downarrow_{\Phi,\Gamma}$$

$$\varphi[H]\downarrow_{\Phi,\Gamma} = \begin{cases} H\downarrow_{\Phi,\Gamma} & \text{if } \varphi \in \Phi \\ \varphi[H\downarrow_{\Phi\cup\{\varphi\},\Gamma}] & \text{otherwise} \end{cases}$$

Regularizing history expressions (2)

$$(\mu h.\, H)\downarrow_{\Phi,\Gamma} = \mu h.\, (H'\sigma'\downarrow_{\Phi,\Gamma\{(\mu h.H)\Gamma/h\}}\sigma)$$

where $H = H'\{h/h_i\}_i$, h_i fresh, $h \notin fv(H')$

$$\sigma(h_i) = (\mu h.H)\Gamma\downarrow_{\Phi\cup guard(h_i,H'),\Gamma}$$

$$\sigma'(h_i) = \begin{cases} h & \text{if } guard(h_i, H') \subseteq \Phi \\ h_i & \text{otherwise} \end{cases}$$