Software Security (SWSec) is a critical aspect of the computer security problem\textsuperscript{[1,2]}

However, it often lacks guidance on how much and when to invest
\begin{itemize}
  \item SWSec can be specialised, variable, and costly\textsuperscript{[3]}
  \item This cost can be a barrier to security practice adoption\textsuperscript{[4]}
\end{itemize}

The result is an environment ripe for over- and under-investment
PROBLEM STATEMENT

How much to invest, and when? (Given a fixed budget B over time t)
**MODELS**

How can secure software engineering be modelled in order to support decision making?

Assumptions on models
- All models are wrong, some are useful
  (George E. P. Box, statistician, 1919 – 2013)
- Security is enacted by “rational” actors
  (in the economic sense)
- Modelling human behaviour to some fidelity can be accomplished
- Data underlying the model is indicative of reality
- The primary goal of a system is not just to be secure

Example post-deployment models:
- Gordon-Loeb (GL)\textsuperscript{10}
- The Iterated Weakest Link (IWL)\textsuperscript{11}

Current models fall short relative to SWSec
- Ill-defined, surrogate measures\textsuperscript{6}, lack of data\textsuperscript{7}
- Gap between risk and implementation\textsuperscript{8}
- Often need process artefacts\textsuperscript{9}
- Fail to balance pre-/post-deployment investment – Not easily extended to SWSec\textsuperscript{5}
EXAMPLE:

ITERATED WEAKEST LINK (IWL)
A model for post-development defence investments

Main Points
- Attackers operate strategically with known (“true”) costs
- Plays out as an iterative process of defender investment against attacks to successive weakest links
- Defenders face uncertainty about the order of the weakest links, which lie along an attack gradient

Actions (played out over $t_{\text{max}}$ periods...)
- Defender chooses $k_1$ initial proactive defences & deploys them
- Attacker succeeds if net gain $a \cdot z$ (asset value $\cdot$ fraction looted) is less than actual cost of attack
- Defender responds with reactive defences, with defined conditions for refraining (or even divestment)

EXAMPLE:

ITERATED WEAKEST LINK (IWL) (CONT.)

Measure approaches using ROSI

$$\text{ROSI} = \text{ALE}_0 - \text{ALE}_1 - \frac{\text{Ave Sec Spend}}{\text{Ave Sec Spend}}$$

Primary aspects of interest

- $k^*$: Optimal proactive defence
- $\sigma$: Defender’s uncertainty
- $\Delta x$: Increase in cost for successive attack; the attack gradient (always 1 in IWL)
- Iterated, discrete time, and encapsulated model

Additional aspects not considered here: sunk costs ($\lambda$), interdependent defences

Parameters used: $a = 1000$, $r = 5\%$, $z = 2.5\%$, $\Delta x = 1$, $x_i = 15$, $n = n_{\text{max}} = 25$
EXAMPLE: IWL-SSE

SWSEC EXTENSIONS TO IWL

Premise: SWSec addresses uncertainty & attack gradient absent attacker input (efficiently)

Model Secure Software Engineering (SSE) starting at time $t = (-1, -2)$
- $t = -2$: Architecture & Design (AD)
- $t = -1$: Implementation & Test (IT)

Removal of two sources of vulnerability
- flaws (errors in design) during AD or IT
- bugs (errors in logic) during IT

Practices employed have an associated effectiveness, $\alpha$ (AD) and $\beta$ (IT)

Each phase is iterated zero or more times (e.g. the number of reviews, tests conducted)
EXAMPLE: IWL-SSE

COSTS OF SWSEC

Each iteration induces costs

- Cost to execute test (c)
- Cost to remediate any errors discovered (e)

Costs scale according to established software engineering measurements (normalised at t = 0)

\[ e_{fXX} = \text{Cost of error (flaw) at phase } \{AD=0.01, IT=0.1\} \]
\[ e_{bXX} = \text{Cost of error (bug) at phase } \{IT=0.01\} \]
\[ c_r, c_t = \text{Cost of conducting a review or test} \]

\((t = -2)\): Architecture & Design (AD)

\[ I_{AD} = i_{AD} \cdot c_r + i_{AD}(ae_{fAD}) \]

\((t = -1)\): Integration & Test (IT)

\[ I_{IT} = i_{IT} \cdot c_t + i_{IT}(\beta[(e_{fIT}/2^{q_{iAD}})+e_{bIT}]) \]

Overall investment

\[ I_P = I_{AD} + I_{IT} \]
EXAMPLE: IWL-SSE

BENEFITS OF SWSEC

(t = 0): Deployment
- Residual uncertainty
  - Maximum Uncertainty
    - Per Phase
      \[ \sigma_{\text{max}} = \sigma_{\text{AD}} + \sigma_{\text{IT}} \]
      \[ \sigma_{\{\text{AD,IT}\}} = (\sigma_{\text{max}} / 2) \{\alpha, \beta\}^{(1/i_{\{\text{AD,IT}\}})} \]

- Attack gradient
  - Increase in attack (normalised to 1)
  \[ \Delta x = \sqrt{1 + \alpha i_{\text{AD}} + \beta i_{\text{IT}}} \]

(t = 1..n): Execute IWL
- Apply SSE values to IWL
- Optimise over (t = -2 ... 0 ... n)

Convex
- Diminishing returns
- Asymptotic to 0

Concave
- Diminishing returns
- Unbounded
What happens to security investment when we consider pre-deployment options?

Example: ROI for static & dynamic cases: \( a = 1000, \ r = 5\%, \ z = 2.5\%, \ Δx=1, \ x_1=15, \ n=t_{\text{max}}=25 \)

- No process (equivalent to original IWL)
- Process iterated over 25 runs of (review \| test): (60\% | 27\%) effective, (3 | 1) cost
RESULTS: IWL-SSE EXECUTION

ROSI comparison for k proactively deployed defences

\[ a = 1000, r = 5\%, z = 2.5\%, x_t = 15, n = t_{\text{max}} = 25 \]

Return on Security Investment (ROSI)

- 0 reviews, 0 tests, sigma = 0
- 0 reviews, 0 tests, sigma = 16
RESULTS: IWL-SSE EXECUTION

ROSI comparison for \( k \) proactively deployed defences

\[ a = 1000, \ r = 5\%, \ z = 2.5\%, \ x_1 = 15, \ n = t_{\text{max}} = 25 \]
RESULTS: IWL-SSE EXECUTION

What happens to security investment when we consider pre-deployment options?

Example: ROI for static & dynamic cases: \( a = 1000, \ r = 5\%, \ z = 2.5\%, \ \Delta x=1, \ x_1=15, \ n=t_{\text{max}}=25 \)

- No process (equivalent to original IWL)
- Process iterated over 25 runs of (review | test): (60% | 27%)[15,16] effective, (3 | 1) cost

A new security measure: The Return on Secure Software Process (ROSSP)

\[
\text{ROSSP} = \text{ROSI}_{\text{SSE}} - \text{ROSI}_{\text{NoSSE}}
\]

Comparison of optimal investments:

IWL: \( n = 25, \ \sigma = 0 \)  \hspace{1cm} ROSI_{\text{NoSSE}}: 33.5 (k=11)

IWL-SSE: \( i_{\text{AD}}=8, \ i_{\text{IT}}=24 \)  \hspace{1cm} ROSI_{\text{SSE}}: 44.6 (k=3)  \hspace{1cm} \text{ROSSP}: 44.6 - 33.5 = 11.1
RESULTS: IWL-SSE EXECUTION

ROSSP comparison for \( k \) proactively deployed defences

CONCLUSION

What do we stand to gain?

- **Make SWSec argument as a business case**
  
  “For business leaders, the biggest motivation for implementing new processes, procedures, or expanding budgets boils down to how much money they can make on the initiative.”

- **A means of scientifically examining SWSec theory and practice**
  
  SWSec effects and outcomes
  Role of pre- and post-deployment security
  Examine the impact of process, approach, and language

Directions & challenges

- **Data.** Stronger ties to the empirical software engineering community; culture of quantitative research

- **Models.** SWSec practice effectiveness and its relationship to investment and benefit; best practices and guidelines to effective SWSec – ongoing work with GL-SSE, IWL-SSE extensions

- **Practice.** Usable SWSec tools and techniques; relationship between theory and practice

---

*For more, see: “A Case for the Economics of Software Security” at the New Security Paradigms Workshop (NSPW ’16), September 2016*
REFERENCES

QUESTIONS?

Chad Heitzenrater
UK: Chad.heitzenrater@cs.ox.ac.uk
US: Chad.Heitzenrater@us.af.mil
https://www.cs.ox.ac.uk/people/chad.heitzenrater/
WEIS 2016 CHARTS
Gartner estimates cybersecurity expenditure topped $75 billion worldwide in 2015, and is expected to increase to $170 billion by 2020.

“When it came to measuring confidence in cybersecurity, half of respondents said they are not confident in their current security products. Almost 1 in 5 feel that effective endpoint security isn’t even possible”

“…54 percent of respondents have low confidence in their company’s ability to demonstrate the ROI of security. For business leaders, the biggest motivation for implementing new processes, procedures, or expanding budgets boils down to how much money they can make on the initiative.”
Current security investment approaches …

emphasise prevention and detection measures\(^1\)
favour third-party additions over built-in solutions\(^1\)
fails to reflect the various sources of vulnerabilities\(^2\)

Alternatives?

Invest in **Secure Software Engineering**: “Build Security In”\(^3\)

Preventing the introduction of vulnerabilities prior to release, rather than patching vulnerabilities afterwards is THE challenge SSE rises to meet\(^2\)

Applications are a “prime vector into an organisation” and “next logical step” for attackers to target, in order to bypass perimeter defences\(^3\)

**How can information security investment models reflect this mindset?**
BACKGROUND:
SOFTWARE ENGINEERING

NIST System Development Lifecycle (SDLC)
Inception – **Acquisition/Development** – Implementation – Operations/Maintenance – Disposal

Waterfall, V-Model

Evolutionary, Iterative, Incremental

Spiral

RAD, JAD, Prototyping

Agile, Lightweight (e.g. Scrum, XP)

Greenfield | Open vs. Closed Source | Product vs. Product Line | Managed vs. community
Traditionally, security engineering has been compliance-focused
- PCI-DSS, ISO 27001
- Orange Book / Common Criteria
- UK Cyber Essentials

However, these approaches are...
- Perpetually incomplete
- Static and prescriptive
- Technology focused
- Example (vs. context) driven
- Can be expensive

SWSec seeks to shift this paradigm to become process-focused
- OWASP Top 10, IEEE CSD 10 Arch Flaws
- Microsoft SDL, Adobe SPL
- Touchpoints, BSIMM

Addresses many issues...
- Proactive to root cause of vulnerability
- Less prescriptive (but requires interpretation)
- Technology independent

...but some remain
- Example (vs. context) driven
- Expensive (and is now unbounded!)
**Problem Statement (cont)**

**Goals**
- Optimise investment over the SDLC
- Inform the project management decisions
- Improve & integrate qualified security approaches
- Demonstrate the return on SWSec investment

**Approach**
- Model secure software process
- Leverage existing models
- Inform starting conditions & analyse output

**Focus & Assumptions (This work)**
- Development \rightarrow Operations (+)
- Managed planning & development
- Single, greenfield, “traditional” development
- Return on Secure Software Process (ROSSP)

- Focus on 2 BSIMM Touchpoints, abstractly: “Reviews” (for flaws) and “Test” (for bugs)
- Link to Integrated Weakest Link (IWL)
- Uncertainty & attack gradient vs. investment return
ITERATED WEAKEST LINK (IWL)

Model for post-development defence investments

Main Points
- Attackers operate strategically with known (“true”) costs
- Plays out as an iterative process of defender investment against attacks to successive weakest links
- Defenders face uncertainty about the order of the weakest links, which lie along an attack gradient

Actions (played out over $t_{\text{max}}$ periods…)
- Defender chooses $k_1$ initial proactive defences & deploys
- Attacker succeeds if gain ($a \cdot z$) is less than actual cost of attack
- Defender responds with reactive defenses, with defined conditions for refraining (or even divestment)

ITERATED WEAKEST LINK (IWL) (CONT.)

RELATIONSHIP BETWEEN UNCERTAINTY AND GRADIENT

$\sigma \Delta x$

True cost of attack

Expected threat ordering

Expected cost of attack

$\sigma \Delta x$

Expected threat ordering

Expected cost of attack
ITERATED WEAKEST LINK (IWL) (CONT.)

Measure approaches using ROSI

\[
\text{ROSI} = \text{ALE}_0 - \text{ALE}_1 - \text{Ave Sec Spend} \\
\text{Ave Sec Spend}
\]

Primary aspects of interest
- \( k^* \): Optimal proactive defence
- \( \sigma \): Defender’s uncertainty
- \( \Delta x \): Increase in cost for successive attack; the attack gradient (always 1 in IWL)
- Iterated, discrete time, and encapsulated model

Additional aspects not considered here: sunk costs (\( \lambda \)), interdependent defences
**SSE EXTENSIONS TO IWL**

**APPLICATION OF SSE**

Premise: SSE addresses uncertainty & attack gradient *absent attacker input* (efficiently)

Model SSE starting at time $t = (-1, -2)$
- $t = -2$: Architecture & Design (AD)
- $t = -1$: Implementation & Test (IT)

Removal of two sources of vulnerability
- flaws (errors in design) during AD or IT
- bugs (errors in logic) during IT

Practices employed have an associated effectiveness, $\alpha$ (AD) and $\beta$ (IT)

Each phase is iterated $\theta$ or more times (e.g. the number of reviews, tests conducted)
SSE EXTENSIONS TO IWL
COSTS OF SSE

Each iteration induces costs
- Cost to execute test (c)
- Cost to remediate any errors discovered (e)

Costs scale according to established software engineering measurements (normalised at $t = 0$)

- $e_{fXX} =$ Cost of error (flaw) at phase $\{AD=0.01, IT=0.1\}$
- $e_{bXX} =$ Cost of error (bug) at phase $\{IT=0.01\}$
- $c_r, c_t =$ Cost of conducting a review or test

(t = -2): Architecture & Design (AD)

- $I_{AD} = i_{AD} \cdot c_r + i_{AD}(ae_{fAD})$

(t = -1): Integration & Test (IT)

- $I_{IT} = i_{IT} \cdot c_t + i_{IT}(\beta[\frac{(e_{fIT}/2^{q_{iAD}})+e_{bIT})]{})$

Overall investment

- $I_P = I_{AD} + I_{IT}$
**SSE EXTENSIONS TO IWL**

**BENEFITS OF SSE**

\[(t = 0)\): Deployment

- Residual uncertainty
  - Maximum Uncertainty
    - Per Phase
  - Attack gradient
    - Increase in attack (normalised to 1)

\[
\begin{align*}
\sigma_{\text{max}} &= \sigma_{\text{AD}} + \sigma_{\text{IT}} \\
\sigma_{\{\text{AD,IT}\}} &= (\sigma_{\text{max}}/2)^{\alpha,\beta}(1/i_{\{\text{AD,IT}\}})
\end{align*}
\]

\[(t = 1..n)\): Execute IWL

- Apply SSE values to IWL
- Optimise over \((t = -2 \ldots 0 \ldots n)\)

---

Convex
Diminishing returns
Asymptotic to 0

Concave
Diminishing returns
Unbounded
RESULTS: MODEL VALIDITY

1. High $\sigma$, Low $\Delta x$ : Poor process
2. Low $\sigma$, Low $\Delta x$ : Sure it is bad
3. Low $\sigma$, High $\Delta x$ : Ideal!
4. High $\sigma$, High $\Delta x$ : Really lucky

Defender Benefits

Normalised Uncertainty ($\sigma/\Delta x$)

Attacker Benefits

Defender Benefits

Attack Gradient ($\Delta x$)
What happens to IWL when we introduce process?

Example: ROI for static & dynamic cases: \( a = 1000, \ r = 5\%, \ z = 2.5\%, \ \Delta x=1, \ \chi_1=15, \ n=t_{\text{max}}=25 \)
- No process (equivalent to original IWL)
- Process iterated over 25 runs of (review | test): (60% | 27%) effective, (3 | 1) cost
RESULTS: MODEL EXECUTION

ROSİ comparison for k proactively deployed defences

\[ a = 1000, \ r = 5\%, \ z = 2.5\%, \ x_1=15, \ n=t_{\text{max}}=25 \]
RESULTS: MODEL EXECUTION

RosI comparison for \( k \) proactively deployed defences

\[ a = 1000, \ r = 5\%, \ z = 2.5\%, \ x_1 = 15, \ n = t_{\text{max}} = 25 \]
RESULTS: MODEL EXECUTION

What happens to IWL when we introduce process?

Example: ROI for static & dynamic cases: \( a = 1000, r = 5\%, z = 2.5\%, \Delta x = 1, x_1 = 15, n = t_{\text{max}} = 25 \)

- No process (equivalent to original IWL)
- Process iterated over 25 runs of (review|test): (60\%|27\%) effective, (3|1) cost

A new security measure: The **Return on Secure Software Process (ROSSP)**

\[
\text{ROSSP} = \text{ROSI}_{\text{SSE}} - \text{ROSI}_{\text{NoSSE}}
\]

Comparison of optimal investments:

- IWL: \( n=25, \sigma=0 \) \hspace{1cm} \text{ROSI: 33.5} (k=11)
- IWL-SSE: \( i_{\text{AD}}=8, i_{\text{IT}}=24 \) \hspace{1cm} \text{ROSI: 44.6} (k=3) \hspace{1cm} \text{ROSSP: 44.6 - 33.5 = 11.1}
RESULTS: MODEL EXECUTION

ROSSP comparison for $k$ proactively deployed defences

$a = 1000, r = 5\%, z = 2.5\%, x_1 = 15, n = t_{\text{max}} = 25$
CONCLUSION & NEXT STEPS

Initial model to examine the role of SWSec in security investment
Demonstrate conditions where SWSec investment can improve overall ROI
Introduce the Return on Secure Software Process (ROSSP) measure

Refinements

- Validate functional forms against further software engineering data
- Expand the notion of bugs and flaws to capture engineering nuance
  - Flaws: role of review activities
  - Bugs: incorporate bug fix literature
- Incorporate sunk costs, defense inter-dependency (IWL)

Extensions

- More expressive model of the software process; relax assumptions
- Impact of process on the initial attack gradient
- Other sources of vulnerability (e.g. post-deployment misconfigurations, IWL-PT)
- Application to other models (e.g. Gordon-Loeb)

Practice: Apply to projects; examine secure software engineering decision-making
QUESTIONS?

Chad Heitzenrater
UK: Chad.heitzenrater@cs.ox.ac.uk
US: Chad.Heitzenrater@us.af.mil
https://www.cs.ox.ac.uk/people/chad.heitzenrater/
NEW WORK: PROCESS
HOW DOES PROCESS AFFECT THE MODEL?

We will consider four classes of model:
- **Waterfall (baseline), V-Model, Dual-V Model**
- **Incremental-Linear**
- **Prototype**
- **Incremental-Iterative (IID), to include agile processes/”continuous” development**
- **Others: Formal, Spiral**

Aspects of the model
- Additional costs incurred
- Process and remediation costs of the phases (and when they converge/diverge/exchange)
- Effectiveness
- Additional considerations

Assumptions
- Faithful execution of the process
- Costs normalised to deployment costs = 1 (to adhere to the IWL)
- Collapsed version of SDLC steps
  - Planning & Requirements
  - Architecture & Design
  - Implementation & Test
  - Operations & Maintenance
HOW DOES PROCESS AFFECT THE MODEL?

ORIGINAL MODEL (WATERFALL, V/DUAL-V MODEL)

Aspects
- Additional costs: None
- Process Costs: Generally, review > test
- Fix Costs: 10x phase increase, starting at 0.01 (e.g. Review = 0.01, Test = 0.1, Deploy = 1)
- Effectiveness: vary

Assumptions
- In-phase fix costs lower than out of phase
- Lack of sequential phase feedback incurs high cost

Additional considerations:
- V-Model: Would investment in Review/Test offset costs in Test/Deploy?
HOW DOES PROCESS AFFECT THE MODEL?

PROTOTYPE MODEL

Aspects
- **Additional costs:** Planning & requirements (-3)
- **Process Costs:** Generally prototype > review > test
  - Throw Away: 100% of prototype costs $P$ are sunk
  - Evolutionary: $g\%$ of prototype costs $gP$ are sunk
  - Result in advance reduction of $\sigma$
- **Fix Costs:** Determined by subsequent model
- **Effectiveness:** Vary. Value of $g$ set by approach, project type

Assumptions
- Same as waterfall (see below)

Additional considerations:
- Prototype placed ahead of other models
- Role of prototype: functional vs security uncertainty?
- (Inverse) relationship between $g$ and additional security uncertainty?
HOW DOES PROCESS AFFECT THE MODEL?
LINEAR-INCREMENTAL MODEL

Aspects
- Additional costs: None
- Process Costs: Generally review > test
- Fix Costs: Assumed to be waterfall
- Effectiveness: Vary

Assumptions
- Same as waterfall
- Deployment at end (after all iterations)

Additional considerations:
- What amount of the previous codebase can be expected to be operated on at a given time?
- Do disconnected iterations raise the uncertainty/lower attack gradient?
- Incremental release = incremental deployment?
HOW DOES PROCESS AFFECT THE MODEL?
IID & AGILE MODEL

Aspects
- **Additional costs:** None
- **Process Costs:** review $>, <, or = test
- **Fix Costs:** review $\approx$ test (little/no differentiation between in-phase and out-of-phase fixes)
- **Effectiveness:** Vary

Assumptions
- The effectiveness of the processes is similar and remain differentiable

Additional considerations:
- Would a true agile process then encompass the deployment phase as well?