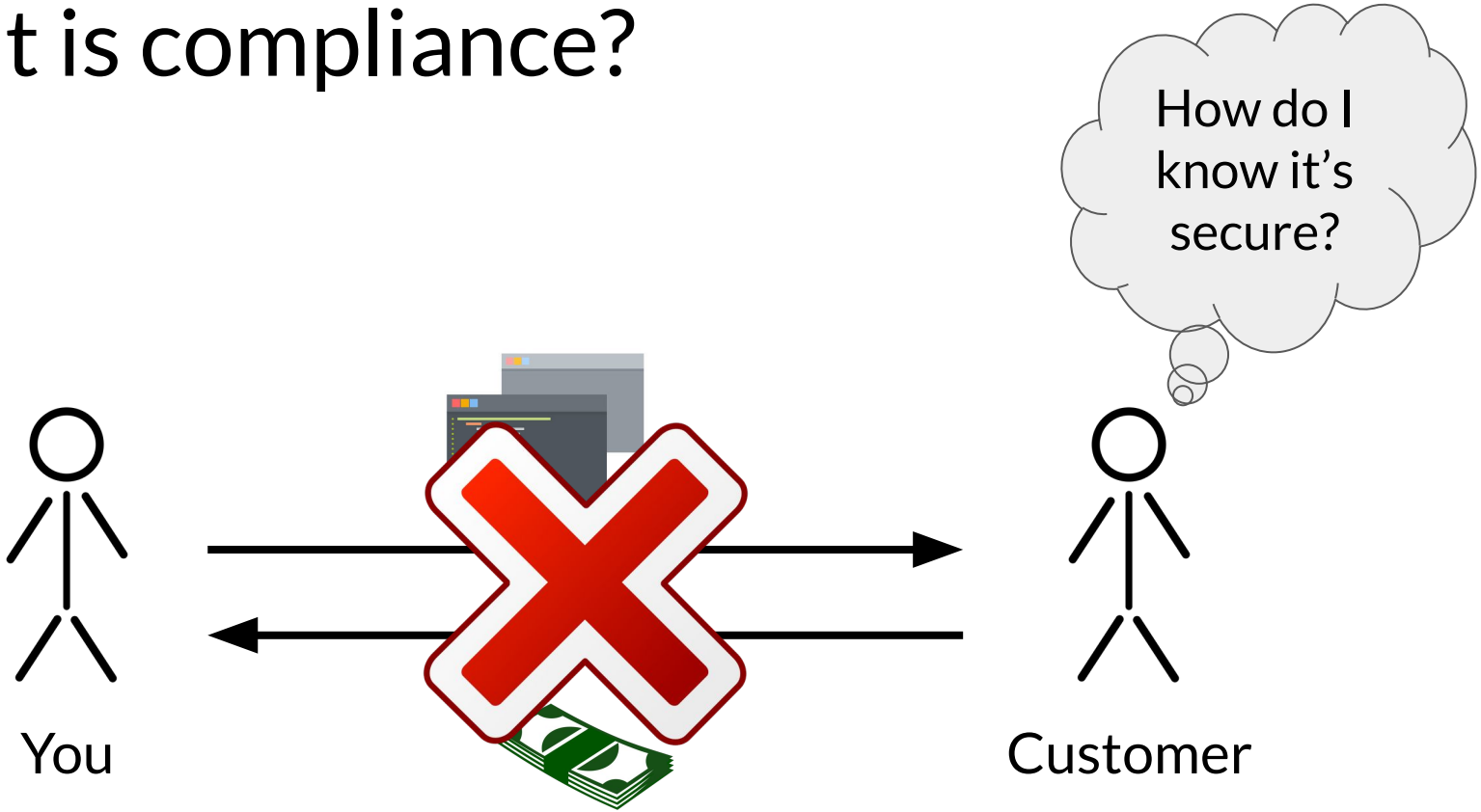
Continuous Compliance

Martin Kellogg[a], Martin Schäf[b], Serdar Tasiran[b], Michael D. Ernst[a,b]

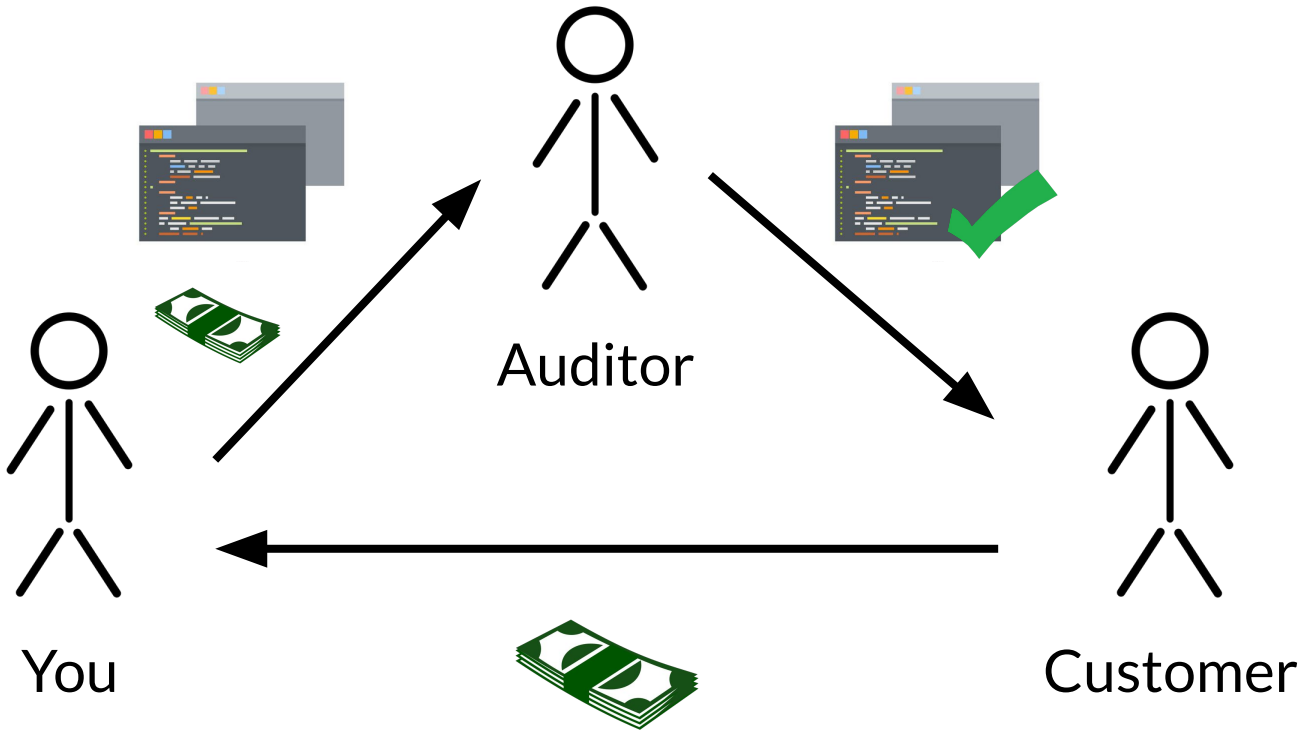
[a]University of Washington [b]Amazon Web Services

What is compliance?

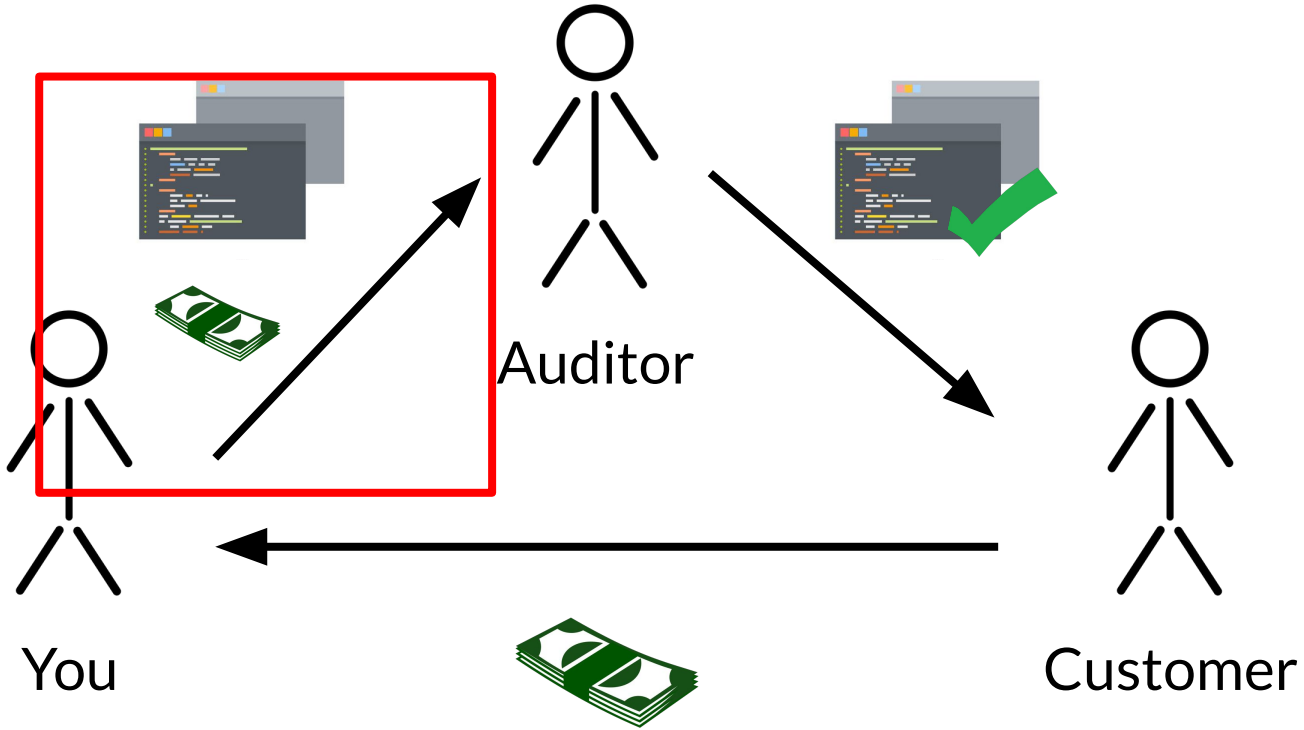
What is compliance?



What is compliance?



What is compliance?



Audit workflow

	Traditional
Development	code review, keep compliance in mind
Preparation	
Review	

Audit workflow

	Traditional
Development	code review, keep compliance in mind
Preparation	gather evidence from each engineering team
Review	

Audit workflow

	Traditional
Development	code review, keep compliance in mind
Preparation	gather evidence from each engineering team
Review	randomly sample, manually check evidence

Audit workflow

	Traditional
Development	code review, keep compliance in mind
Preparation	gather evidence from each engineering team
Review	randomly sample, manually check evidence

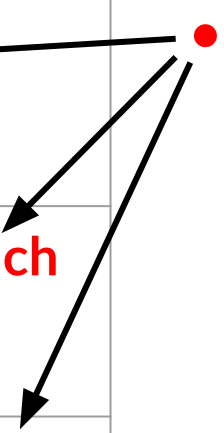
Problems:

Audit workflow

	Traditional
Development	code review, **keep compliance in mind**
Preparation	gather evidence from **each engineering team**
Review	randomly sample, **manually check evidence**

Problems:

- **Cost**

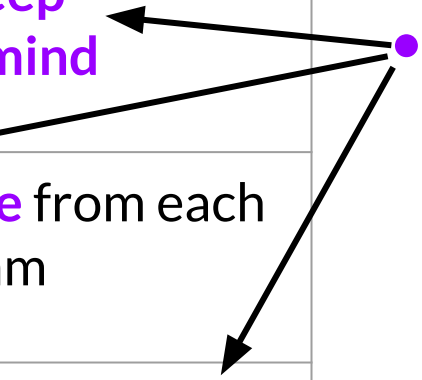


Audit workflow

	Traditional
Development	code review, keep compliance in mind
Preparation	gather evidence from each engineering team
Review	randomly sample, manually check evidence

Problems:

- Cost
- Judgment



Audit workflow

	Traditional
Development	code review, keep compliance in mind
Preparation	gather evidence from each engineering team
Review	**randomly sample**, manually check evidence

Problems:

- **Cost**
- **Judgment**
- **Sampling**

Audit workflow

	Traditional
Development	code review, keep compliance in mind
Preparation	gather evidence from each engineering team
Review	randomly sample, manually check evidence

Problems:

- Cost
- Judgment
- Sampling
- Regressions

Continuous Compliance

- Build verification tools for compliance controls
- On each commit, run verifier in continuous integration
- Report failures directly to developers

Audit workflow

	Traditional	Continuous
Development	code review, keep compliance in mind	
Preparation	gather evidence from each engineering team	
Review	randomly sample, manually check evidence	

Audit workflow

	Traditional	Continuous
Development	code review, keep compliance in mind	write specifications, verifier runs in CI
Preparation	gather evidence from each engineering team	
Review	randomly sample, manually check evidence	

Audit workflow

	Traditional	Continuous
Development	code review, keep compliance in mind	write specifications, verifier runs in CI
Preparation	gather evidence from each engineering team	none
Review	randomly sample, manually check evidence	

Audit workflow

	Traditional	Continuous
Development	code review, keep compliance in mind	write specifications, verifier runs in CI
Preparation	gather evidence from each engineering team	none
Review	randomly sample, manually check evidence	auditor checks output of verifier

Audit workflow

- Cost
- Judgment
- Sampling
- Regressions

	Traditional	Continuous
Development	code review, keep compliance in mind	write specifications, verifier runs in CI
Preparation	gather evidence from each engineering team	none
Review	randomly sample, manually check evidence	auditor checks output of verifier

Audit workflow

~~Cost~~
~~Judgment~~
~~Sampling~~
~~Regressions~~

	Traditional	Continuous
Development	code review, keep compliance in mind	write specifications, verifier runs in CI
Preparation	gather evidence from each engineering team	none
Review	randomly sample, manually check evidence	auditor checks output of verifier

Contributions

- **Idea**: verification is a good fit for compliance

Contributions

- **Idea**: verification is a good fit for compliance
- **Engineering**: we built verifiers for five compliance controls

Contributions

- **Idea**: verification is a good fit for compliance
- **Engineering**: we built verifiers for five compliance controls
- **Experimental**: open-source experiments and comparisons

Contributions

- **Idea**: verification is a good fit for compliance
- **Engineering**: we built verifiers for five compliance controls
- **Experimental**: open-source experiments and comparisons
- **Experiential**: verifiers in the compliance process at AWS

Compliance controls

Controls:

- HTTP vs HTTPS
- Cryptographic key length
- Cryptographic algorithm selection
- Cloud data store initialization
- Hard-coded credentials

Compliance controls

Controls:

- HTTP vs HTTPS
- Cryptographic key length
- Cryptographic algorithm selection
- Cloud data store initialization
- Hard-coded credentials

Techniques:

Compliance controls

Controls:

- HTTP vs HTTPS
- Cryptographic key length
- Cryptographic algorithm selection
- Cloud data store initialization
- Hard-coded credentials

Techniques:

- Constant propagation

Compliance controls

Controls:

- HTTP vs HTTPS
- Cryptographic key length
- Cryptographic algorithm selection
- Cloud data store initialization
- Hard-coded credentials

Techniques:

- Constant propagation
- + enum analysis

Compliance controls

Controls:

- HTTP vs HTTPS
- Cryptographic key length
- Cryptographic algorithm selection
- Cloud data store initialization
- Hard-coded credentials

Techniques:

- Constant propagation
- + enum analysis
- + regex matching

Compliance controls

Controls:

- HTTP vs HTTPS
- Cryptographic key length
- Cryptographic algorithm selection
- Cloud data store initialization
- Hard-coded credentials

Techniques:

- Constant propagation
- + enum analysis
- + regex matching
- + accumulation analysis

Compliance controls

Controls:

- HTTP vs HTTPS
- Cryptographic key length
- Cryptographic algorithm selection
- Cloud data store initialization
- Hard-coded credentials

Techniques:

- Constant propagation
- + enum analysis
- + regex matching
- + accumulation analysis
- + dataflow

Analysis strategy

Analysis strategy: type systems

- Familiar to developers
- Predictable
- Scalable
- Sound

Evaluation

1. Run all verifiers on 492 open-source projects
2. Compare verifiers to existing tools
3. Case study of a verifier in a real, industrial compliance workflow
4. Case study of two verifiers as part of industrial security scans

Evaluation

1. **Run all verifiers on 492 open-source projects**
2. Compare verifiers to existing tools
3. Case study of a verifier in a real, industrial compliance workflow
4. Case study of two verifiers as part of industrial security scans

Open-source projects

- 492 projects from GitHub, 5.7 million LoC
  - Use type inference and build scanning to automate process

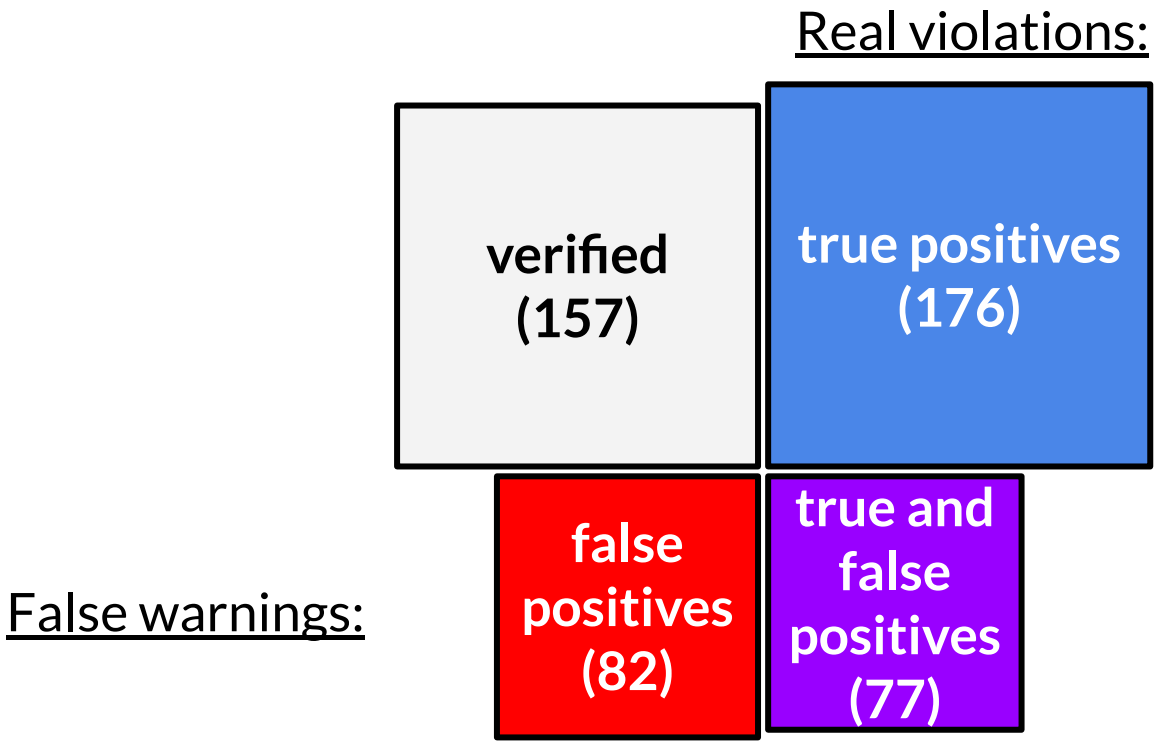
Open-source projects

- 492 projects from GitHub, 5.7 million LoC
  - Use type inference and build scanning to automate process

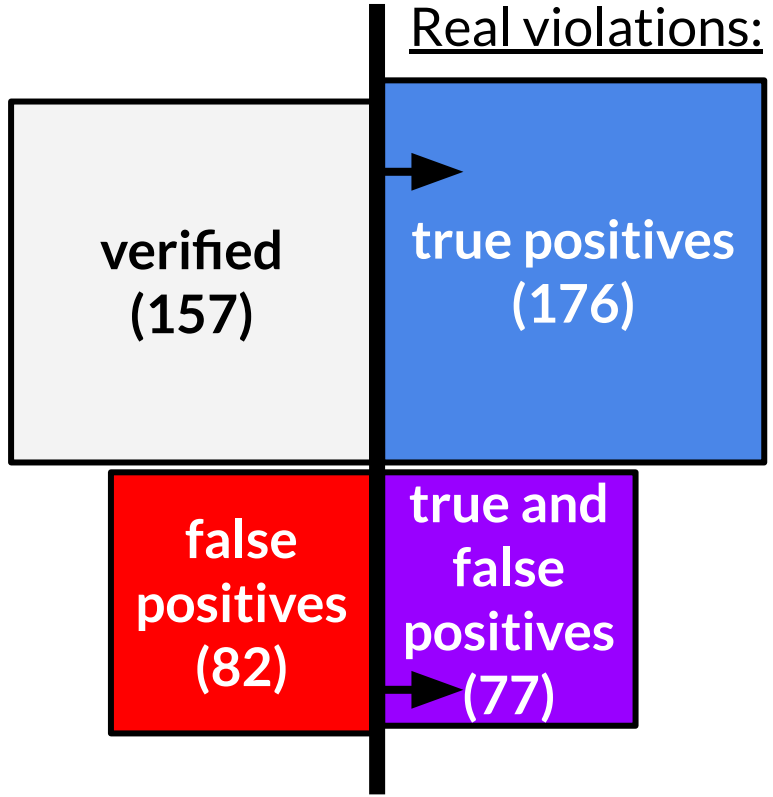
- Triage into 4 categories:

		Real violations:
	verified, no warnings	**true positives**: all warnings are real violations
False warnings:	**false positives**: warnings, no real violations	**true and false positives**: some warnings are real

Open-source projects



Open-source projects

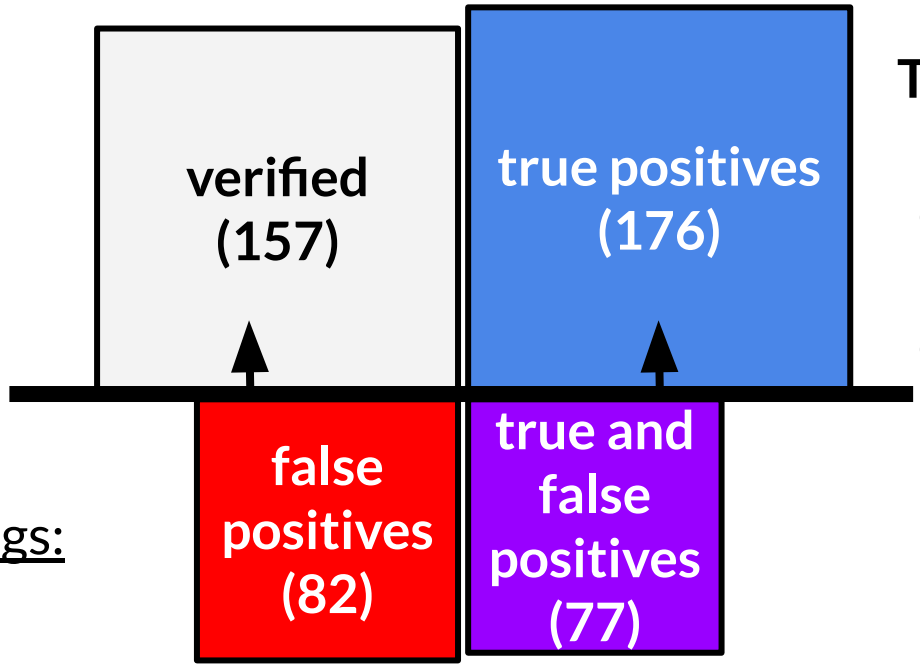


Takeaways:

- ~1/2 open-source projects have compliance violations

Open-source projects

Real violations:



False warnings:

Takeaways:

- ~1/2 open-source projects have compliance violations
- ~2/3 projects cause no false positives from our tools

Evaluation

1. Run all verifiers on 492 open-source projects
2. **Compare verifiers to existing tools**
3. Case study of a verifier in a real, industrial compliance workflow
4. Case study of two verifiers as part of industrial security scans

Comparison with other tools

- Used a CryptoAPIBench, a previously-published benchmark
- Only compared on categories covered by our tools (11/16)
- Four other tools:
  - SpotBugs
  - Coverity
  - $\text{CogniCrypt}_{\text{SAST}}$ (CrySL)
  - CryptoGuard

Comparison with other tools

Tool	SpotBugs	Coverity	CrySL	CryptoGuard	Ours
Precision					
Recall					

Comparison with other tools

Tool	SpotBugs	Coverity	CrySL	CryptoGuard	Ours
Precision	0.69	**1.0**	0.79	**1.0**	0.97
Recall					

Comparison with other tools

Tool	SpotBugs	Coverity	CrySL	CryptoGuard	Ours
Precision	0.69	**1.0**	0.79	**1.0**	0.97
Recall	0.32	0.38	0.61	0.88	**1.0**

Comparison with other tools

Tool	SpotBugs	Coverity	CrySL	CryptoGuard	Ours
Precision	0.69	**1.0**	0.79	**1.0**	0.97
Recall	0.32	0.38	0.61	0.88	**1.0**

Only ours are suitable for compliance: auditors won't accept a tool that has **false negatives**

Evaluation

1. Run all verifiers on 492 open-source projects
2. Compare verifiers to existing tools
3. **Case study of a verifier in a real, industrial compliance workflow**
4. Case study of two verifiers as part of industrial security scans

AWS case study 1: auditor acceptance

- key-length verifier
- verified in CI for 7 core AWS services
- **replaced** existing manual compliance workflow
- auditors accepted output of tool: **all services compliant**

AWS case study 1: auditor acceptance

- key-length verifier
- verified in CI for 7 core AWS services
- **replaced** existing manual compliance workflow
- auditors accepted output of tool: **all services compliant**

“It eliminates [the need for] a lot of trust”

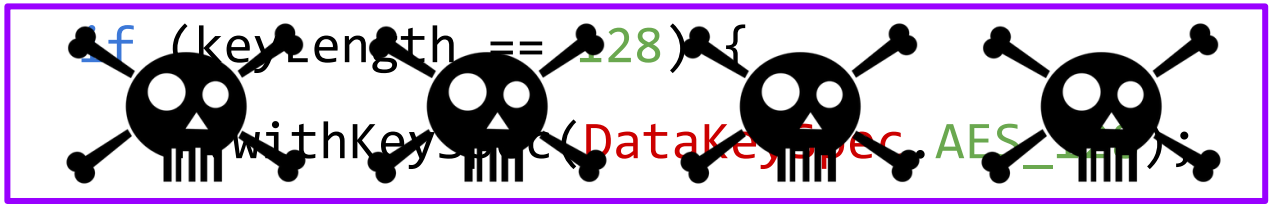
- external auditor

Why does it eliminate the need for trust?

```
public SecretKey getKMSKey(int keyLength) {
  GenerateDataKeyRequest r = new GenerateDataKeyRequest();
  if (keyLength == 128) {
     r.withKeySpec(DataKeySpec.AES_128);
  else {
     r.withKeySpec(DataKeySpec.AES_256);
  }
  ...
```

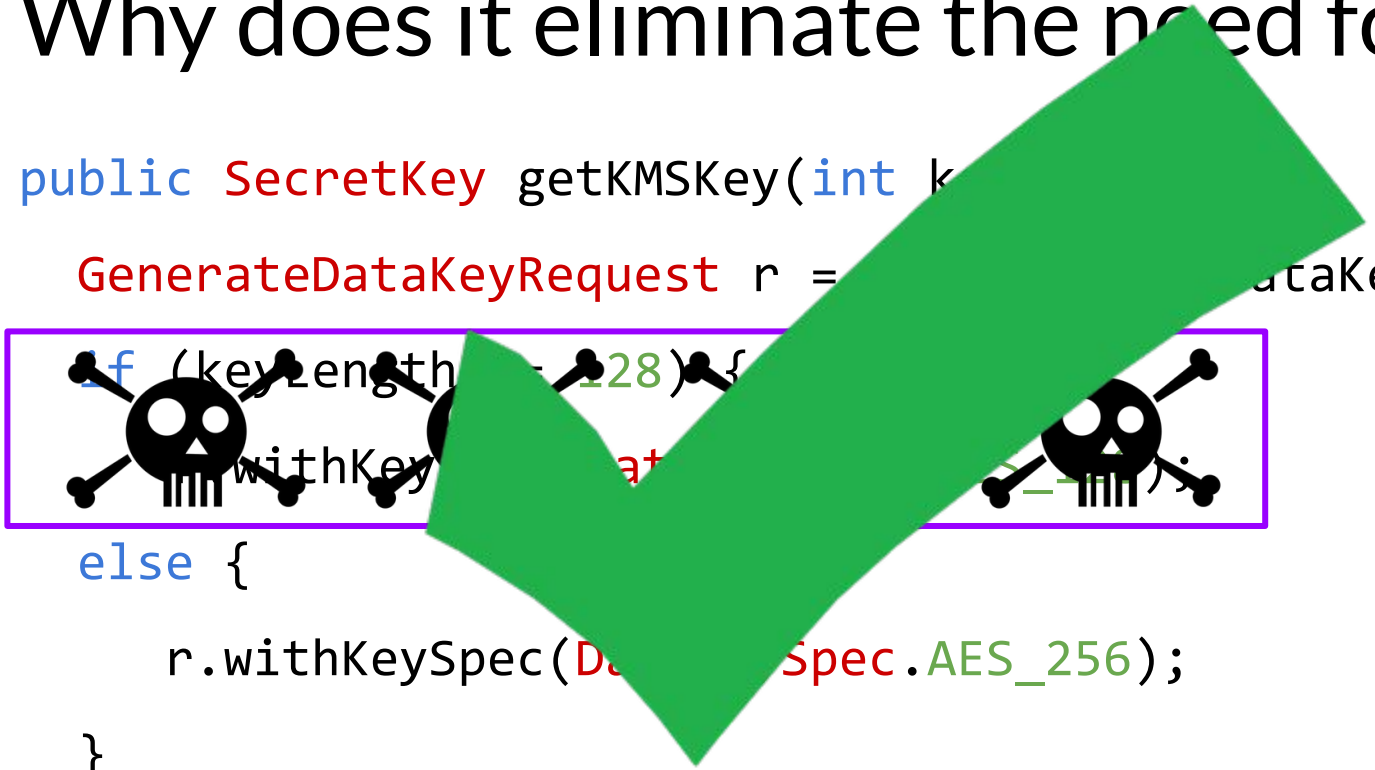
Why does it eliminate the need for trust?

```
public SecretKey getKMSKey(int keyLength) {
  GenerateDataKeyRequest r = new GenerateDataKeyRequest();
  if (keyLength == 128) {
     r.withKeySpec(DataKeySpec.AES_128);
  else {
     r.withKeySpec(DataKeySpec.AES_256);
  }
  ...
```



Why does it eliminate the need for trust?

```
public SecretKey getKMSKey(int k
  GenerateDataKeyRequest r =                    ataKeyRequest();
  if (keyLength == 128) {
     withKey          at                    _128);
  }
  else {
     r.withKeySpec(D        Spec.AES_256);
  }
  ...
```



AWS case study 1: auditor acceptance

- key-length verifier
- verified in CI for 7 core AWS services
- **replaced** existing manual compliance workflow
- auditors accepted output of tool: **all services compliant**

“It eliminates [the need for] a lot of trust”

- external auditor

AWS case study 1: auditor acceptance

- key-length verifier
- verified in CI for 7 core AWS services
- **replaced** existing manual compliance workflow
- auditors accepted output of tool: **all services compliant**

"It eliminates [the need for] a lot of trust"

- external auditor

"This has saved my team 2 hours every 6 months and we also don't have to worry about failing an audit control."

- developer

AWS case study 1: auditor acceptance

- key-length verifier
- verified in CI for 7 core AWS services
- **replaced** existing manual compliance workflow
- auditors accepted output of tool: **all services compliant**

"It eliminates [the need for] a lot of trust"

- external auditor

"This has saved my team 2 hours eve
months and we also don't have to wo
about failing an audit control."

- developer

per team,
per audit,
per control

Evaluation

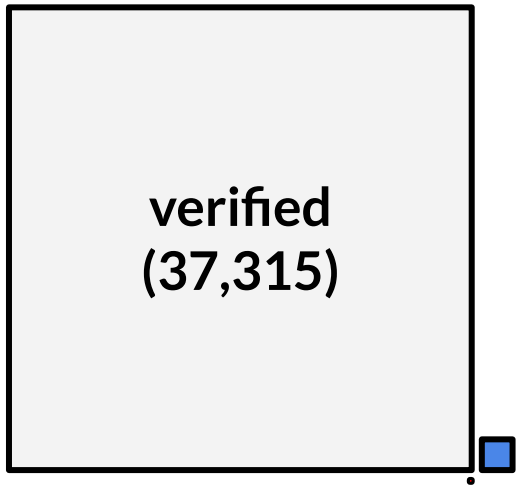
1. Run all verifiers on 492 open-source projects
2. Compare verifiers to existing tools
3. Case study of a verifier in a real, industrial compliance workflow
4. **Case study of two verifiers as part of industrial security scans**

AWS case study 2: security scanning

- key-length and crypto-algorithm verifiers
- scan all security-relevant (not just compliance relevant) code

Industrial projects

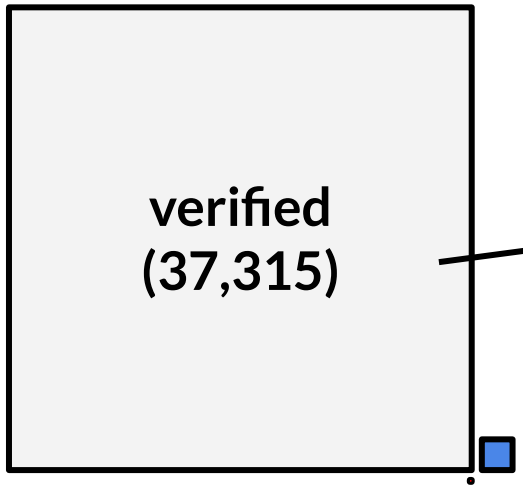
Real violations:



true positives (173)

False warnings: **false positives (1)** **true and false positives (0)**

Industrial projects



Real violations:

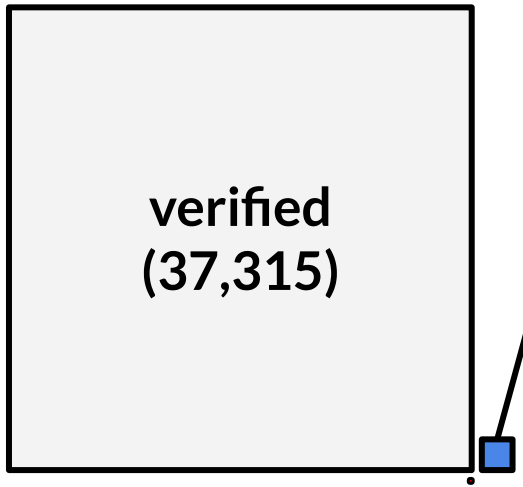
99.94% required no annotations

true positives (173)

False warnings: **false positives (1)** **true and false positives (0)**

Industrial projects

Real violations:



All validated by security engineers; none compliance relevant

true positives (173)

False warnings: **false positives (1)** **true and false positives (0)**

Lessons learned

Lessons learned

1. **Verification is a good fit for compliance**
    a. auditors require soundness (no false negatives)
    b. most controls are local and simple (human-checkable)

Lessons learned

1. **Verification is a good fit for compliance**
   a. auditors require soundness (no false negatives)
   b. most controls are local and simple (human-checkable)
2. **Verification is useful for stakeholders other than programmers**
   a. auditors, managers, security reviewers, etc.
   b. research impact from focusing on other stakeholders

Lessons learned

1. **Verification is a good fit for compliance**
   a. auditors require soundness (no false negatives)
   b. most controls are local and simple (human-checkable)
2. **Verification is useful for stakeholders other than programmers**
   a. auditors, managers, security reviewers, etc.
   b. research impact from focusing on other stakeholders
3. **Verification can save time for developers**
   a. don't add a new task, replace an existing task
   b. verification is easier than tasks developers already do

Contributions

- **Idea**: verification is a good fit for compliance
- **Engineering**: we built verifiers for five compliance controls
- **Experimental**: open-source experiments and comparisons
- **Experiential**: verifiers in the compliance process at AWS

Tools and data are publicly available: see paper for links

Problems with traditional audits

- **Cost**: lost engineering time, paying auditors, failed audits, etc.

Problems with traditional audits

- **Cost**: lost engineering time, paying auditors, failed audits, etc.
- **Judgment**: humans can make mistakes

Problems with traditional audits

- **Cost**: lost engineering time, paying auditors, failed audits, etc.
- **Judgment**: humans can make mistakes
- **Sampling**: not a proof that there is not a violation

Problems with traditional audits

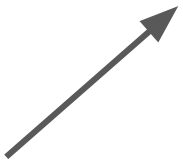
- **Cost**: lost engineering time, paying auditors, failed audits, etc.
- **Judgment**: humans can make mistakes
- **Sampling**: not a proof that there is not a violation
- **Regressions**: only checked at audit-time

Compliance code example

```
void makeCipher() {
    Cipher.getInstance(“AES”);
}
```

Compliance code example

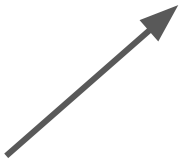
```
void makeCipher() {
    Cipher.getInstance(“AES”);
}
```



String

Compliance code example

```
void makeCipher() {
    Cipher.getInstance(“AES”);
}
```



```
@StringVal(“AES”) String
```

Compliance code example

```
void makeCipher() {
    Cipher.getInstance(“AES”);
}
```