IRONCLAD APPS

End to End Security Via Automated Full System Verification

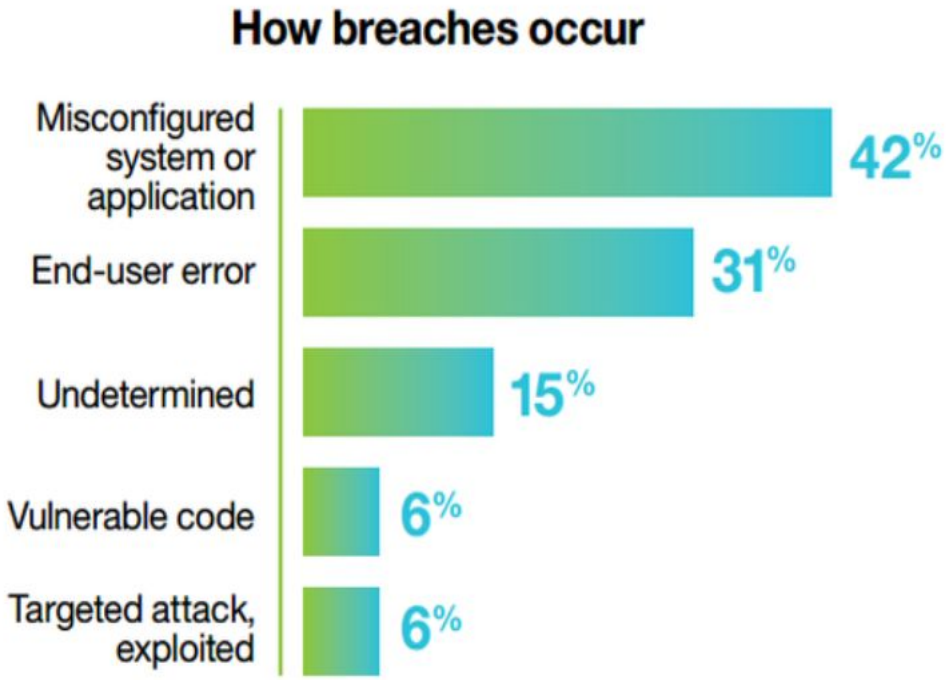
AISHWARYA KOSGI

R11844470

INFORMATION

- What is Ironclad?
- Working Model
- Verification Goals
- Verification Methodology
- Architecture/Rapid Verification
- Advantages/Limitations/Challenges
- Future Developments
- Summary
- References

HOW DATA BREACH OCCURS



- However software's or apps uses Secure Socket Layer (SSL) technology to protect data even with a small code error or end user error here may be a data breach.

WHAT IS IRONCLAD?

IRONCLAD APP

An Ironclad app guarantees to remote parties that every instruction it executes adheres to a high-level security spec.

CHARACTERISTICS

This does more than eliminate implementation vulnerabilities such as buffer overflows, parsing errors, or data leaks

HISTORY

This was a research activity carried out by Microsoft by Chris Hawblitzel, Jon Howell, and Jacob R. Lorch.

HOW THEY ACHIEVE THIS

They achieve this by using complete, low-level software verification.

They then use cryptography and secure hardware to enable secure channels from the verified software to remote users.

Ironclad Combines

LATE LAUNCH

It is a feature developed by Intel and AMD drivers to run software stack in protected environment.

TRUSTED COMPUTING

This actually means that we are bring up software and cryptographic key together.

SOFTWARE VERIFICATION

This is used to prove that the software has some desirable property to turn up by some high level specifications.

SECURE REMOTE EQUIVALENCE

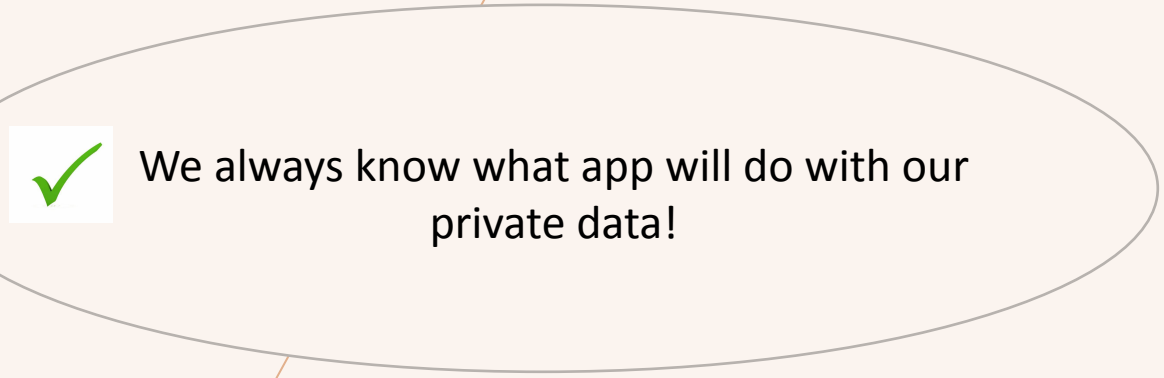
By the combination of all the above three features this secure remote equivalence property is created.

By this property a secure channel is created so that user can have a secure communication directly without any data leak.

WORKING MODEL

VERIFICATION GUARANTEES

- No buffer overflows
- No code injection
- No type safety flaws
- No information disclosures
- No crypto implementation flaws



VERIFICATION GOALS

END TO END SECURITY

This proposed model don’t trust any of the other software running on machine or drivers or libraries or OS or app itself.

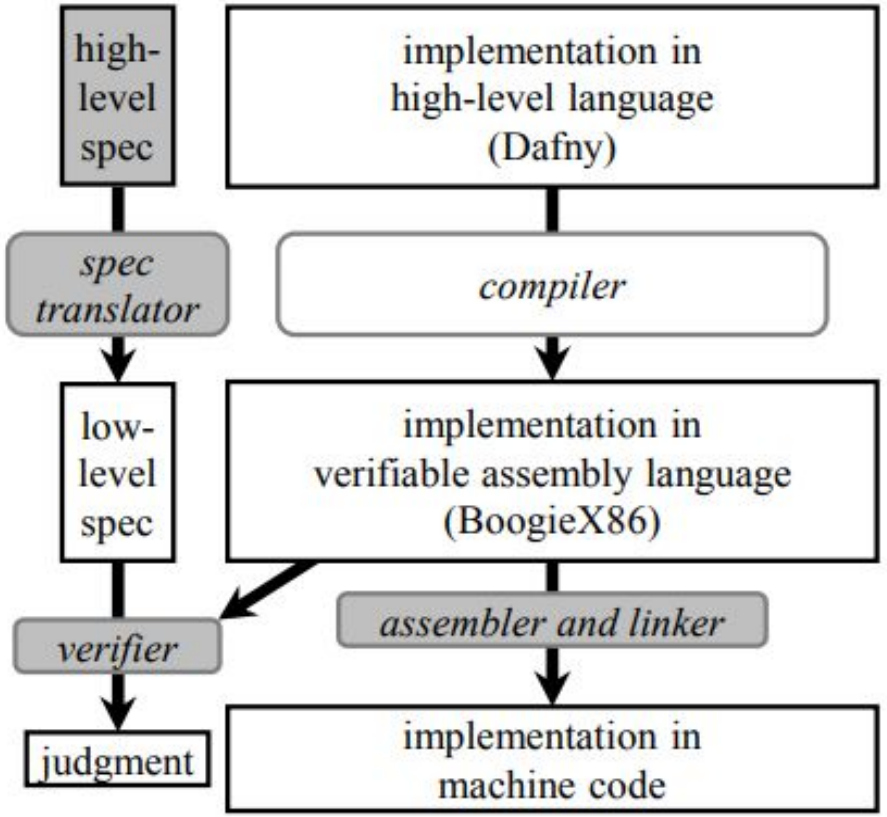
COMPLETE SECURITY

They will verify the entire system including OS to provide complete security.

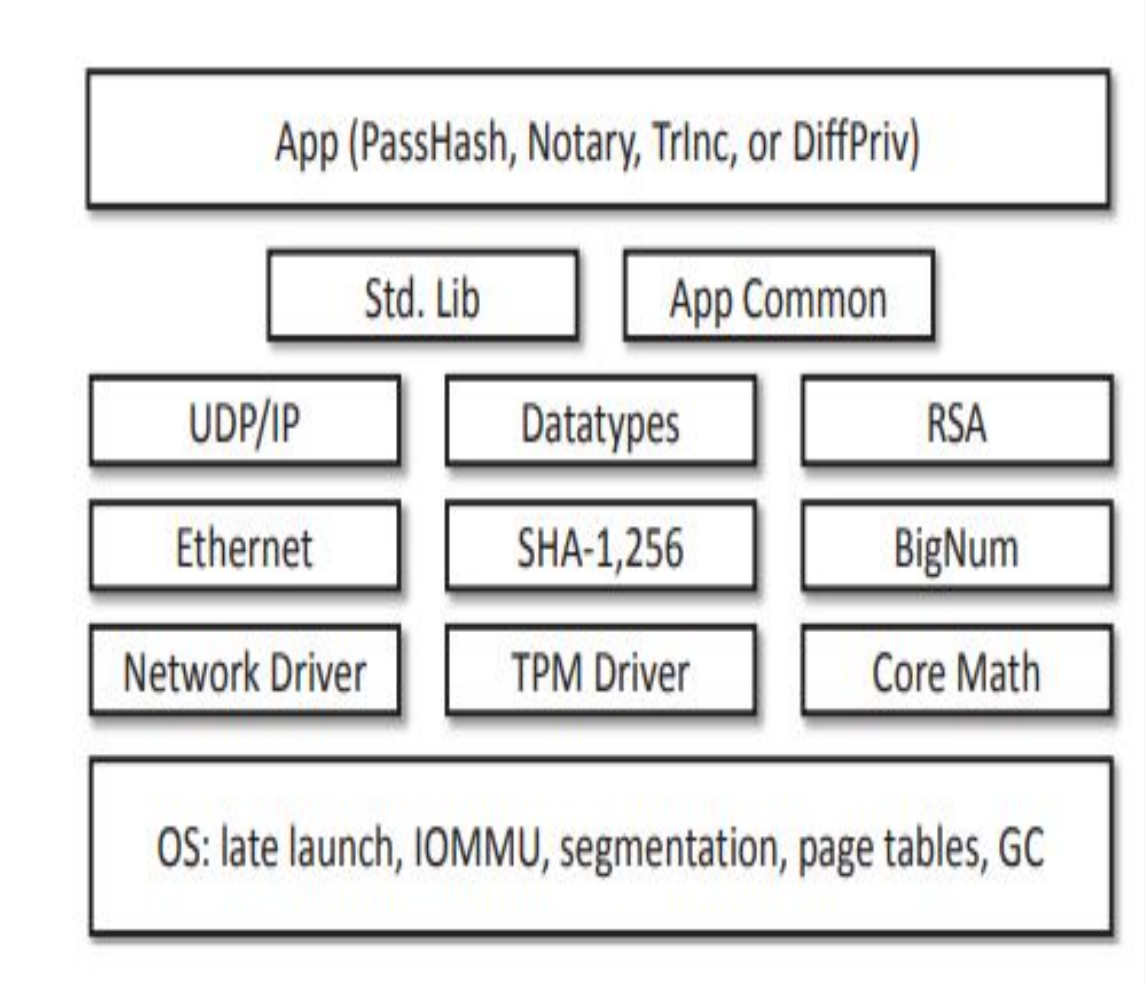
LOW LEVEL

As they don’t trust complier or runtime they maintain low level to verify actual assembly instructions.

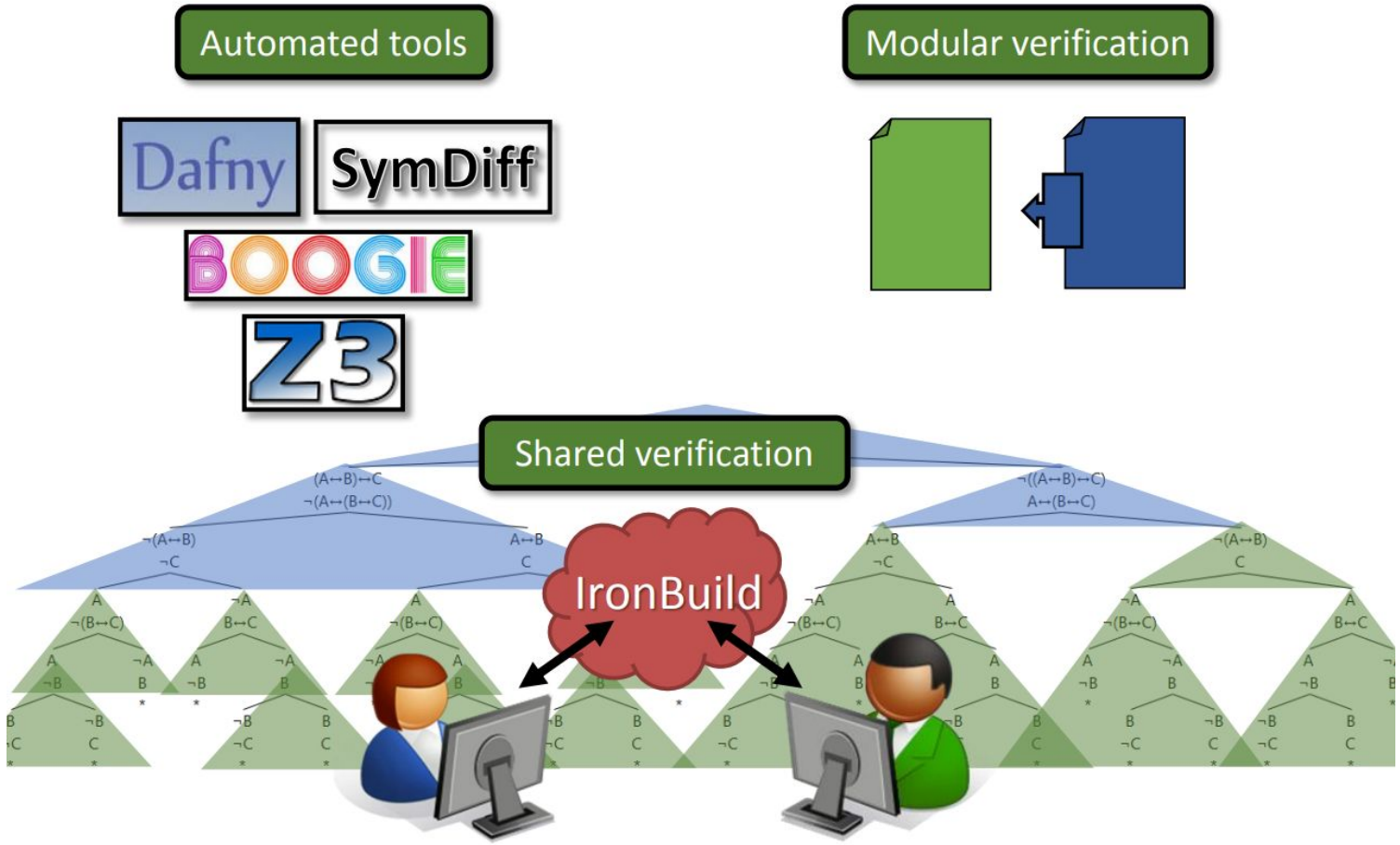
VERIFICATION METHODOLOGY



ARCHITECTURE



RAPID VERIFICATION



ADVANTAGES

Ironclad guarantees end-to-end security to remote parties: Every instruction meets the app's security spec

APPLICATIONS

- Password Protector
- Notary
- Differentially Private DB
- Trusted Incrementer

CHALLENGES

- They can't verify existing code
- For this model they are focusing on performance

LIMITATIONS

- They don't prove absence of side channels
- Liveness
- Physical Security

FUTURE DEVELOPMENTS

Currently, they prove the functional correctness and noninterference of our system, but their future developments include proofs that could be extended in two directions that constitute ongoing work:

1.proving liveness and

2.connecting our guarantees to even higher-level cryptographic protocol correctness proofs.

SUMMARY

- By using automated tools, they have verified full-system, low-level, end-to-end security guarantees about Ironclad Apps.
- To get better usage we expect to see full-system verification scale to larger systems and higher-level properties in the years to come.
- Achieved via: New and modified tools
- A methodology for rapid verification of systems software
- Verification of systems code is quite feasible!

REFERENCES

1. Chris Hawblitzel, Jon Howell, and Jacob R. Lorch, Microsoft Research; Arjun Narayan, University of Pennsylvania; Bryan Parno, Microsoft Research; Danfeng Zhang, Cornell University; Brian Zill, Ironclad Apps: End-to-End Security via Automated Full-System Verification, 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’14), USENIX Association.

THANKYOU

Aishwarya Kosgi
R11844470
akosgi@ttu.edu