Ensuring Host Integrity With Cryptographic Provenance Verification

RUTGERS UNIVERSITY

Deian Stefan, Chehai Wu, Danfeng (Daphne) Yao, and Gang Xu

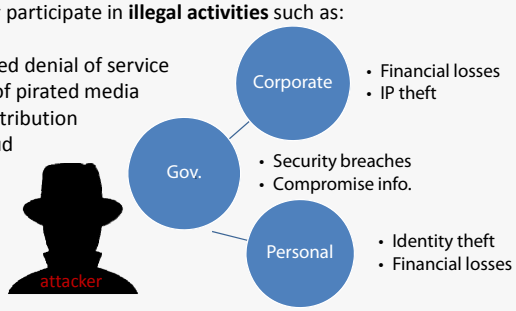
The Cooper Union, AppFolio, Inc., Rutgers University, AT&T

Background and Objective

Studies show that **millions of computers are infected by malware** and controlled by cyber criminals.

These *bots* participate in **illegal activities** such as:

- Distributed denial of service
- Storage of pirated media
- Spam distribution
- Click fraud



Existing detection solutions are in an arm race with continuously changing malware → **ineffective!**

Our Approach

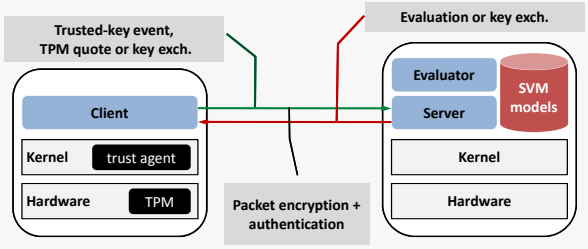
There are intrinsic difference between human and bot interaction with computers, thus **detection of anomalies in behavior patterns** results in **detection of malware!**

We focus on the characteristic behaviors of humans and propose a **cryptographic provenance verification approach**, and demonstrate its use in

- **keystroke-based bot identification,**
- **rootkit traffic detection.**

TUBA Integrity Service

We design a TPM-based infrastructure which cryptographically ensures the integrity of user input events.



Cryptographically trusted input approach:

1. Trust agent signs each keystroke event and machine state.
2. Server verifies the signed keystroke event and client state.
3. Server analyzes keystroke events to identify non-human characteristics.

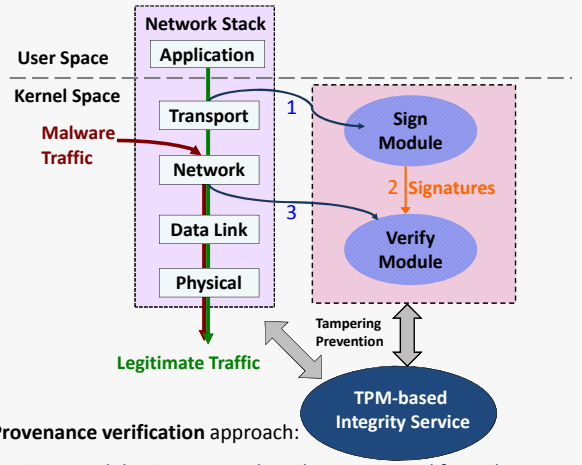
The integrity service **defends against advanced malware attacks** including:

- replay attacks
- fake key-event injections
- tampering of the TUBA client

} **very difficult in non-TPM environment!**

Rootkit Detection

CompareView is an add-on to the host's network stack enforcing outbound network traffic to pass through the entire stack:

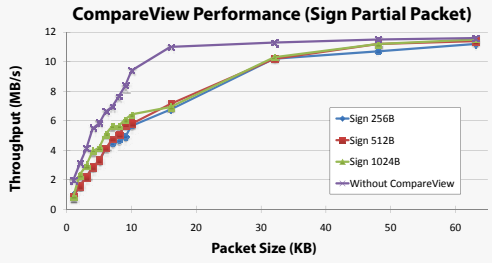


Provenance verification approach:

1. Sign Module signs network packets generated from the application layer.
2. Sign Module sends encrypted signatures as packet provenance information to the Verify Module directly.
3. Verify Module verifies the packet provenance.

Rootkits and other malware circumventing higher layers of the network stack are easily **detected with CompareView!**

Malware	Capability	Detected?
FU_Rootkit	Hide process information	✓
AFXRootkit	Hide process information	✓
hxdef	Hide process information	✓
proof-of-concept malware	Hide outbound traffic from the transport layer	✓



Conclusions

We propose a malware detection approach based on the characteristic behaviors of human users.

Our cryptographic provenance approach is used to

- **provide a trusted input infrastructure**
- **identify bots using keystroke biometric**
- **rootkit traffic detection with very little overhead**