The Observer Effect
&
Cyber Feng-Shui

Mr. Jacob I. Torrey
@JacobTorrey
Assured Information Security

Thesis Statement

- “Through non-conventional uses of existing technology, it is possible to build and combine capabilities to: detect general behaviors of other software on the system, create and seal keying material to a specific device, and bootstrap trusted and opaque implants”

- If you only remember one thing about this talk, make it that!

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Disclaimer

devastating capability, revolutionary advantage

- This is a technical talk, but covers a wide breadth of topics; if you want to go deeper, follow up in person or on Twitter
  - I could talk about this stuff for days ☺
- If you have a clarifying question about an acronym or other concept, please ask during talk as I guarantee others are too shy to ask the same
- If you have more broad questions, please hold until the end

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Who am I?

devastating capability, revolutionary advantage

- Advising security researcher at Assured Information Security
  - Leads Denver, CO office
  - Leads the low-level computer architectures group
  - Plays in:
    - SMM
    - VMM
    - BIOS
- LangSec Co-conspirator
- Avid outdoorsman/traveler

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Outline

- Problem & Introduction
- The Triforce
  - A software dynamic root-of-trust to detect introspection
    - Trusted Computing
  - Reliable device-specific keys
    - PUFs
    - Shamir's Secret Sharing & Error Correcting Codes
  - Secure execution enclave
    - Encrypted Execution
- Putting it all together
- Conclusions

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Problems

- Post-exploitation and persistence is a challenge with the advances in next-gen AV, hypervisor analysis and outsourced RE
- During extensive red-team campaign, obtaining and maintaining trust in your implants is critical to prevent subjugation or loss of tactics, techniques, & procedures (TTPs)
- Reputation-based analysis finds odd files and exfiltrates to AV for RE analysis, must prevent “burning” of capabilities

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Introduction

- Need three tools to bootstrap a trusted implant
- 1. Detect introspection / determine presence of hypervisor
  - Presence of introspection leaves indicators (observer effect)
- 2. Generate device-specific keys to use for attestation and protection
  - Getting cozy! (feng shui)
- 3. Provide opaque execution environment
  - HARES from last year!

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Detecting Introspection

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Background: Trusted Computing

- Field of security aiming to establish a *root-of-trust* on a system, and extend that via a *trust chain*
- Championed by the Trusted Computing Group, providing TPM, etc.
- Helps network administrators ensure malicious software is not running on their systems

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Background: Trusted Computing II

- TC is slowly becoming popular and provided by OEMs
  - UEFI SecureBoot native on Windows
  - TPMs common on x86
- Only signed/approved boot-loaders can execute
- Remote attestation provides ability to ensure remote systems are in a trusted state

Background: Trusted Computing III

- Four main TC measurements
  - Static and dynamic
  - HW and SW-based for each
- Static measurement
  - Starting from initial boot, and each element must be measured before execution
- Dynamic measurement
  - Allows for the TCB to be shrunk by measured launch

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

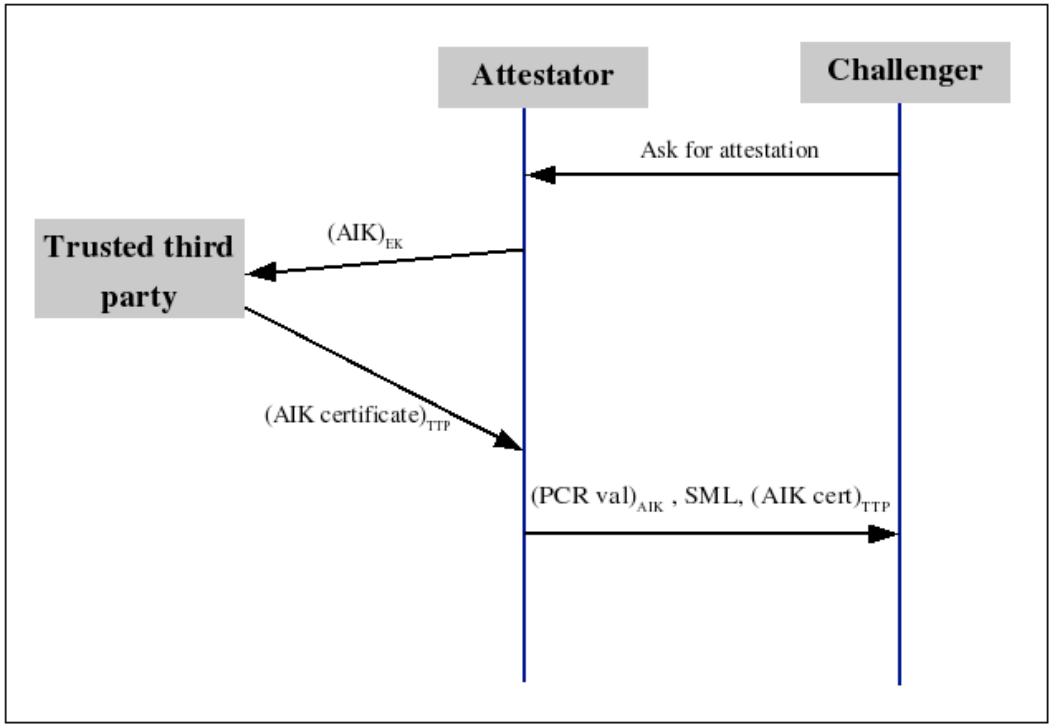
Background: Chain-of-Trust

- Platform configuration registers in TPM provide only reset and *extend*
  - Extend(hash): PCR_n' := SHA1(PCR_n || hash)
- Provides a way to ensure that every "link" in the chain is correct
- Cannot set arbitrary values without breaking SHA1

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Background: Remote Attestation

- Uses TPM attestation identity key (AIK) to sign PCR values to prove chain-of-trust to remote entities



https://www.cs.bham.ac.uk/~mdr/teaching/modules/security/lectures/TrustedComputingTCG.html

Detecting Introspection

- Need to determine if another process, even with more privileges is attempting to cross isolation boundary to introspect

- If introspection is detected, next steps may be different:
  - Avoid detection
  - Shrink forensics footprint
  - Prevent loss of capabilities or TTPs

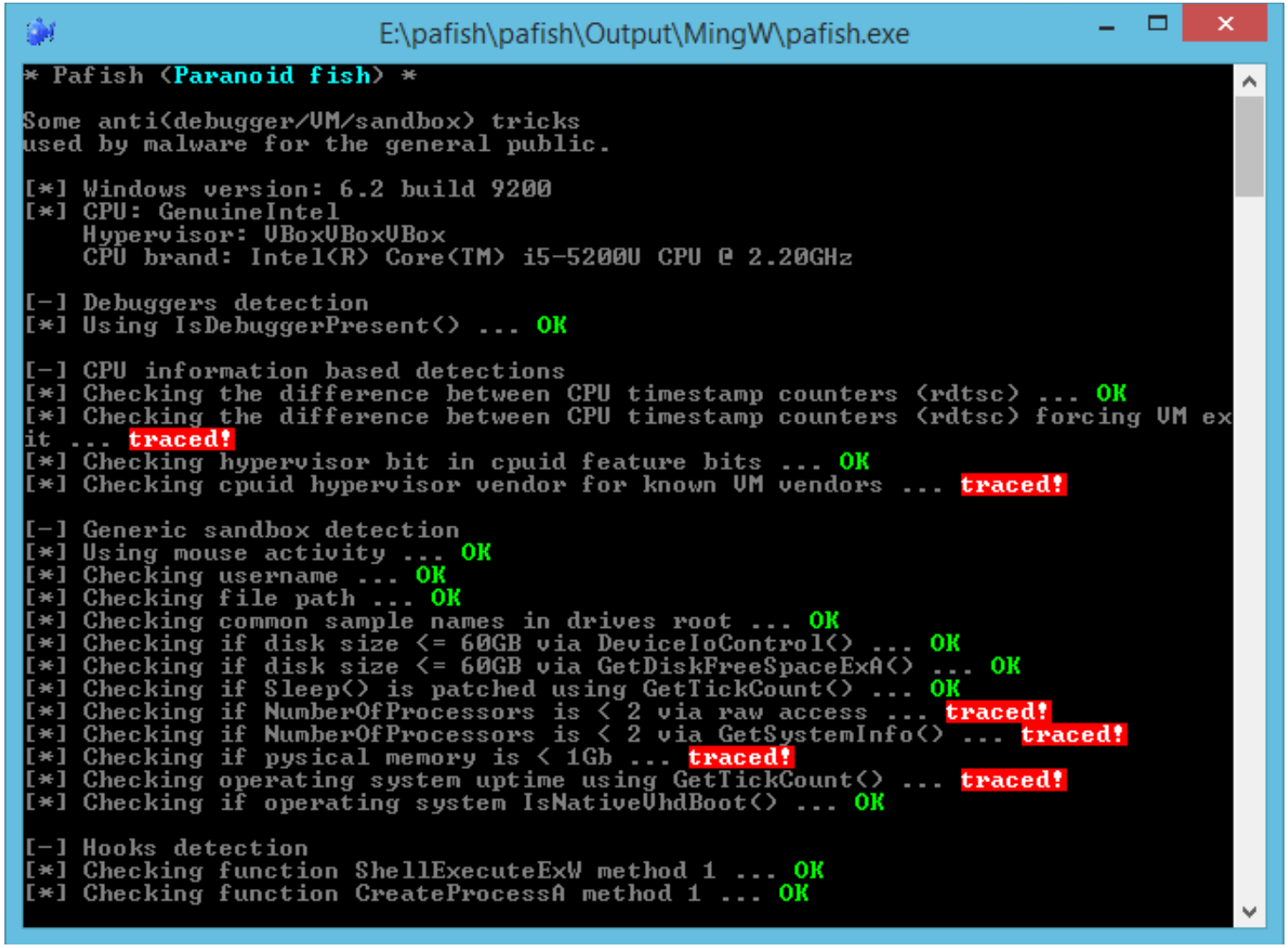
5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Paranoid Fish

- A FOSS tool to collect techniques used by malware to determine if a sandbox or debugger is being used
- Good basis to look for common sandboxes
- Open-source eases integration, though helps sandbox vendors avoid detection

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Paranoid Fish



Shared CPU Resources

- CPUs work hard to provide the illusion of isolation between processes
  - Virtual memory
  - Transparent interrupt-driven multi-tasking
  - Etc…
- Many of these shared resources provide ability to stealthily measure execution and determine what other processes are doing
  - See also Anders' talk and Sophia's closing keynote

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Shared CPU Resources II

- Some examples:
  - CPU cache
  - CPU pipeline
  - Timing/load information

- By measuring these share resources, other process' behavior can be inferred

- Deltas from expected can provide insight into other process' actions

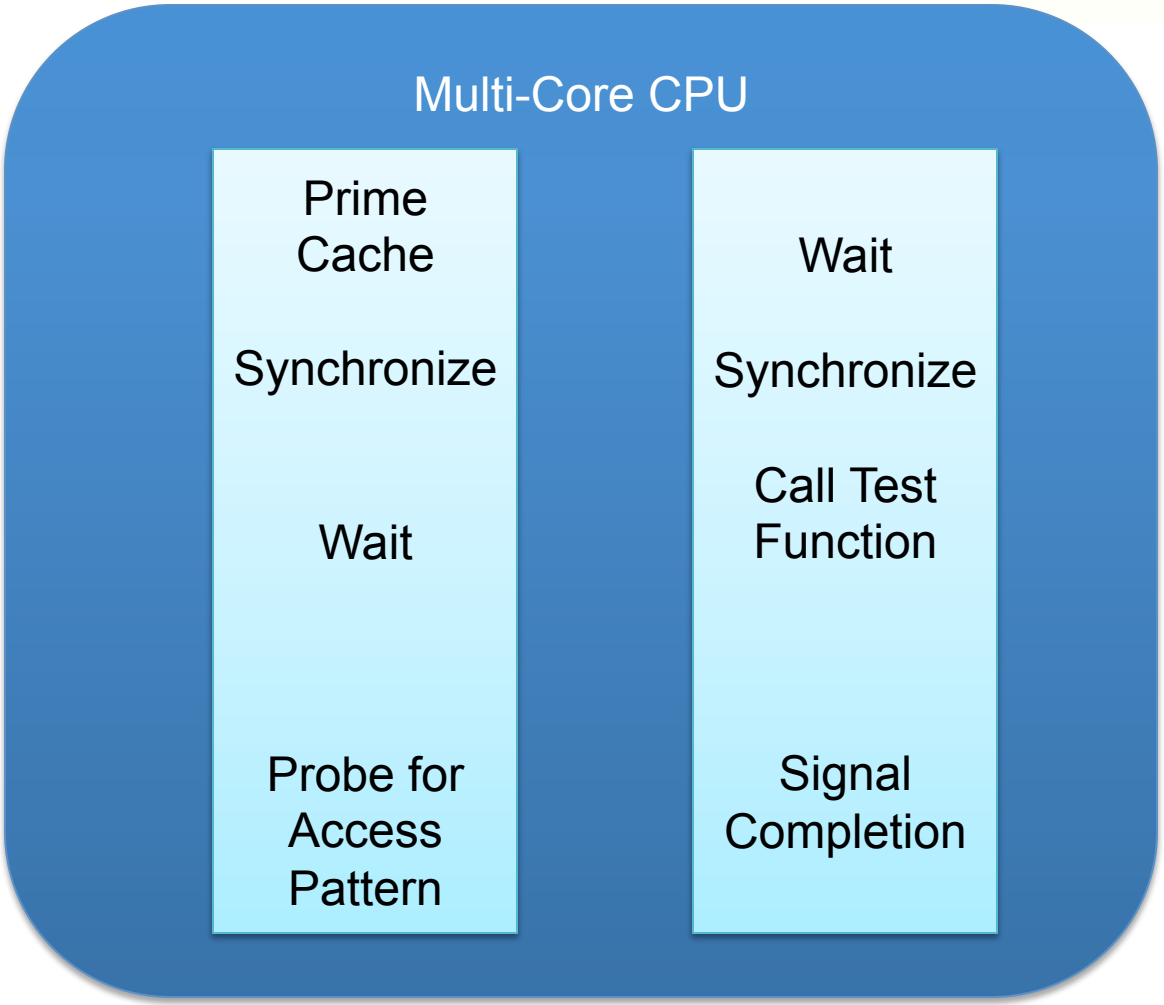
5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Cache Teller I

- Using shared cache to determine instruction access patterns of other threads
- One thread monitors cache using prime and probe
- Other thread triggers function that may be changed
  - Trigger VM exit, OS interrupt or SMI
- Binary classifier to determine if functionality is normal

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Cache Teller II



5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

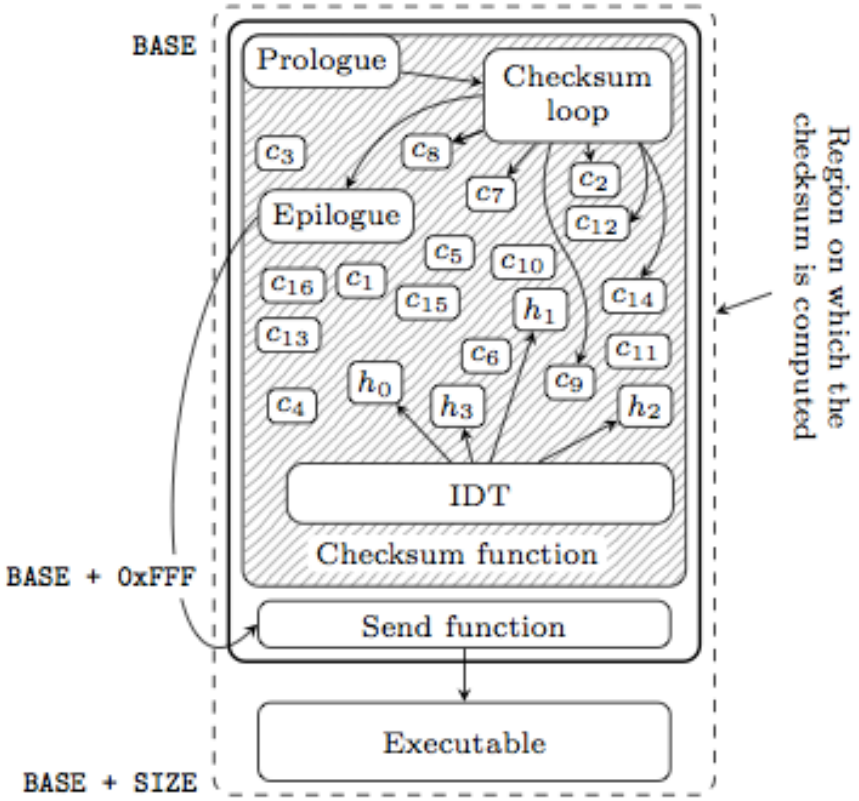
Cache Teller III

- Can be used to detect abnormal execution flow:
  - Hooked functions
  - VM introspection (does accessing this page of memory invoke the EPT handler?)
  - Is the SMI cache impact consistent through testing
- Runs in ring-3 (user space)
- Provides granular data to feed to classifier

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Conqueror

- Randomized gadgets to measure self and system state



Conqueror II

- Requires trusted time source and random seed to prevent replay attacks and detect if challenge has been reversed or emulated

- Some gadgets are critical:
  - CPUID to trigger VM exit
  - Self-modifying code to detect split-TLB
  - Interrupt handler to ensure not running in V8086 mode

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Conqueror III

- Use the resulting checksum to detect if there is debugging, VM presence or even SMI
- Either use with local hardware that is trusted or over network to prevent timing source emulation
- Using resulting checksum to derive the inputs to the PUF challenges or extend with other measurements

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

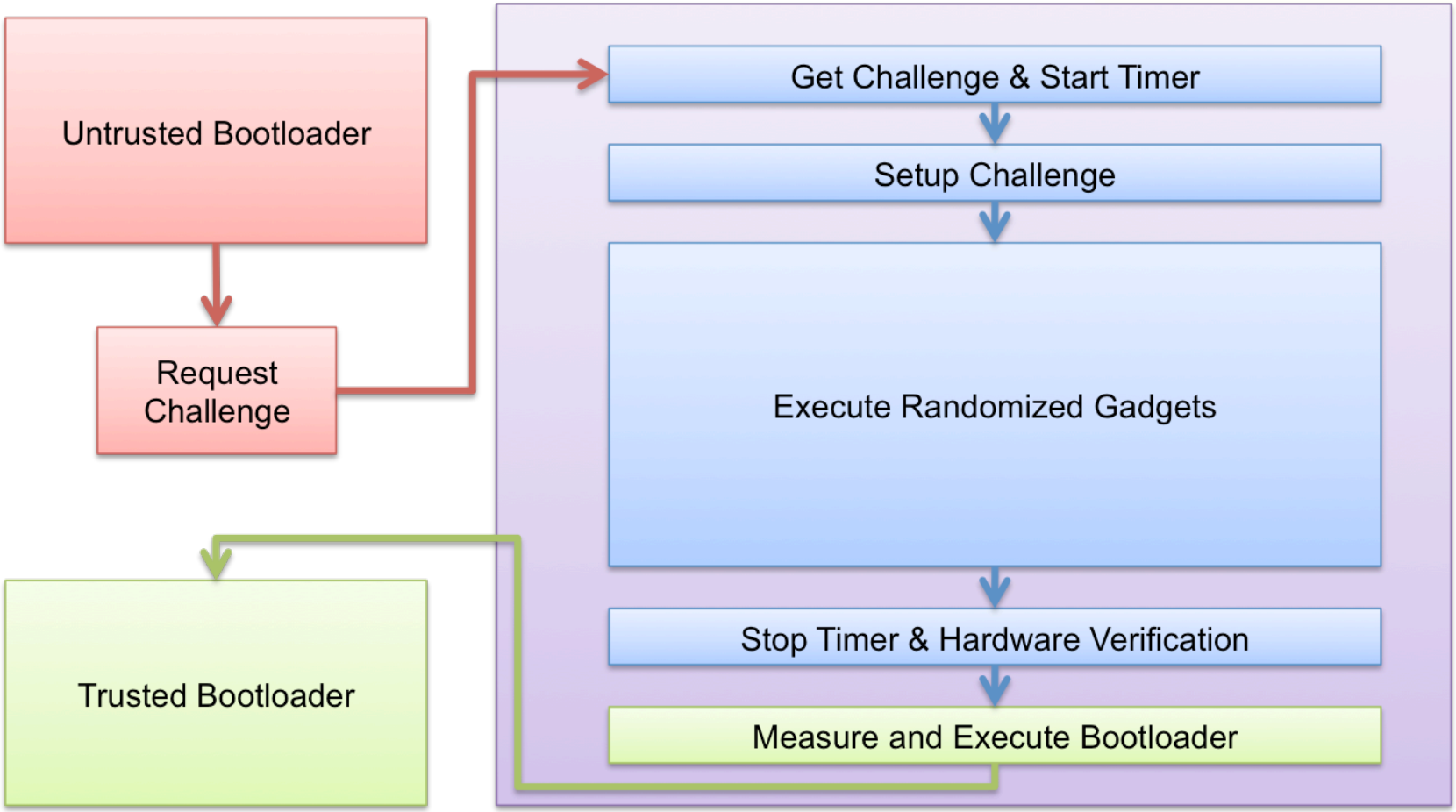
SecureNode I

- Integrated Conqueror system into bootloader to verify it was loaded correctly
- Uses FPGA, USB or remote server as trusted time source
- Would use checksum to “unlock/enable” vTPM
- **Provides S-DRTM (software dynamic root-of-trust measurement) capability**

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

SecureNode II

devastating capability, revolutionary advantage



5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

SecureNode III

- [Demo video of booting Linux without trusting BIOS]

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

SecureNode Caveats

- When performing offensive operations, you do not get the users' benefit of the doubt
  - Many TC primitives require physical presence to provision
  - If you alert the user, you draw attention; in defensive situations you have human relations to back it up

- VM detection no longer is enough, becoming very prevalent
  - Need more granular analysis of "how introspective" the VMM is

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

PUFs

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Background: PUFs I

- Physically-Uncloneable Functions
  - Expose *manufacturing variance* to generate device-specific replies to challenges
- As ICs have shrunk, impossible to build with exactly identical layout, so tolerances are used to prevent errors
- PUFs allow software to access these specific attributes

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Background: PUFs II

- Provides a challenge-response function:
  - Puf(chal) => device-unique response
- These responses should be as non-volatile as possible, and different from one device to the next
- Research showing can be copied with FIB

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Background: PUFs III

- Can be used to attest execution system or generate device-specific data
- Typically require specific techniques for each hardware device, new techniques under active research
- Temperature & hardware age dependent

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Background: Shamir's Secret Sharing

- Method to split a piece of sensitive data (e.g., private key) into *n* pieces in which *m* are needed to recover secret
- Ensures that brute-force search space remains large even with *m-1* pieces
- Uses *m* points on (n+1)-dimensional function

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Background: Error Correcting Codes

- Designed for transmissions in noisy channels
- Increases data size to allow for correction or detection of data corruption
- Tolerance for errors dependent on amount of additional input

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

SRAM PUF Example

- Commonly found on embedded devices and in register banks, SRAM (or IRAM) has natural PUF tendencies:

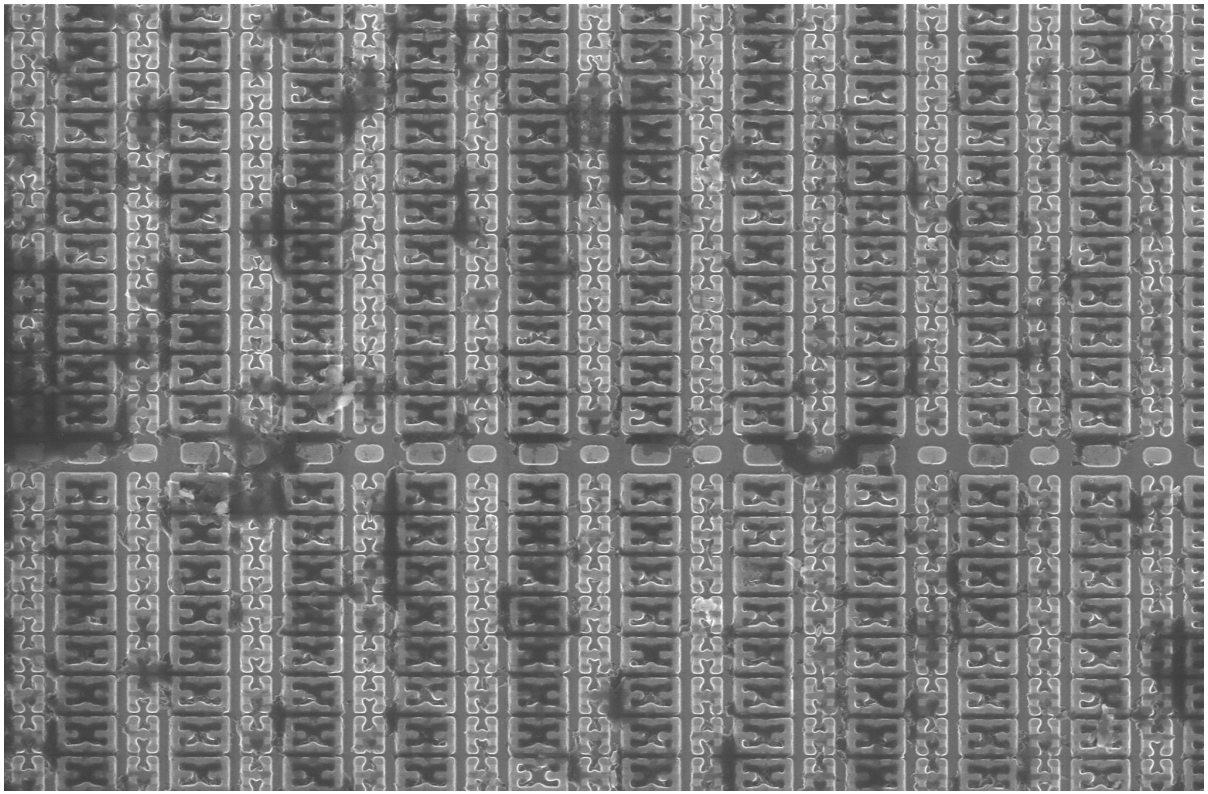


Photo (CC) ZeptoBars

SRAM PUF Example II

- Due to minute differences in the size of each cell, tends to either 1 or 0 on power-on
- The challenge is the physical address range, response is the default bit values
- Each SRAM chip will have different tendencies, but will generally be consistent across power-cycles

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Reliable PUFs

- Due to the nano-scale physics PUFs rely upon, temperature and HW age can change PUF responses
- Need a method to improve reliability whilst maintaining security assurances
- Error correcting codes and Shamir's secret sharing to the rescue!

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Reliable PUFs II

- Categorize PUFs on similar hardware to target to measure expected failure rate, variance, etc…
- Using error-correcting codes to correct small errors between PUF responses
  - Store *key* XOR ECC(SSS_k)
- SSS can be used to add in additional error handling by setting *m of n* ratio, and can be used to have various PUF sources all combined for greater HW usage

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

PUF Sources

- Depending on environment, there may be a mix of:
  - SRAM (power-on initial values)
  - FPGA (inter-gate differences)
  - NAND Flash (cell charge requirements)
  - EEPROM (block erase/write error)
  - DDR RAM (row layout)

- These can be combined to create extremely device-specific "*fingerprint*"
  - Use as method to tie crypto keys to device
  - Prevent outsourced RE

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

FPGA PUFs

- Typically created by creating two ring oscillators that *should* oscillate at the same frequency
- Due to minute gate distance differences, depending on where the logic is programmed onto fabric, the oscillators will beat each other in a unique pattern
- Challenge is where the logic is programmed, response is the resultant bit-stream

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Flash PUFs

- Flash is programmed through applying current to a cell to write/erase for a data-sheet specified time
- Each cell has a unique inflection point where the write “sticks”
- Write for less than the specified time, some cells will switch some will not
- Challenge is flash blocks to change, response is written data

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

EEPROM PUFs

- EEPROM is programmed through applying current for a data-sheet specified time
- Write for less than the specified time, some bits will fail to hold new value
- Challenge is bytes to change, response is written data or errors
- RAM EEPROM is exposed over SMBus for timing information!

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

DDR RAM PUFs

- ***Under active analysis & development***
- Rowhammer showed static builds during RAM row reads, can flip neighboring bits
- Hypothesis: bits that flip are minutely closer to adjacent row, will form device and row-specific pattern of flipping

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

PUF Library

- Will be releasing open-source, modular and cross-platform library for using PUFs on different types of hardware

- Should have initial cleanup and documentation completed in a few months, contact me if you want early access

- Will be: https://github.com/ainfosec/puflib

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Key Usage

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Key Usage

- Keys protected in this manner can then be used for providing:
  - Root-of-trust keys for remote attestation of system (similar to TPM AIK)
  - Data at rest or in transit protections
  - Software enclave key
- PKI layered on top of device specific key(s)
  - Key hierarchies to allow implants to communicate and trust other implants (trusted botnet)

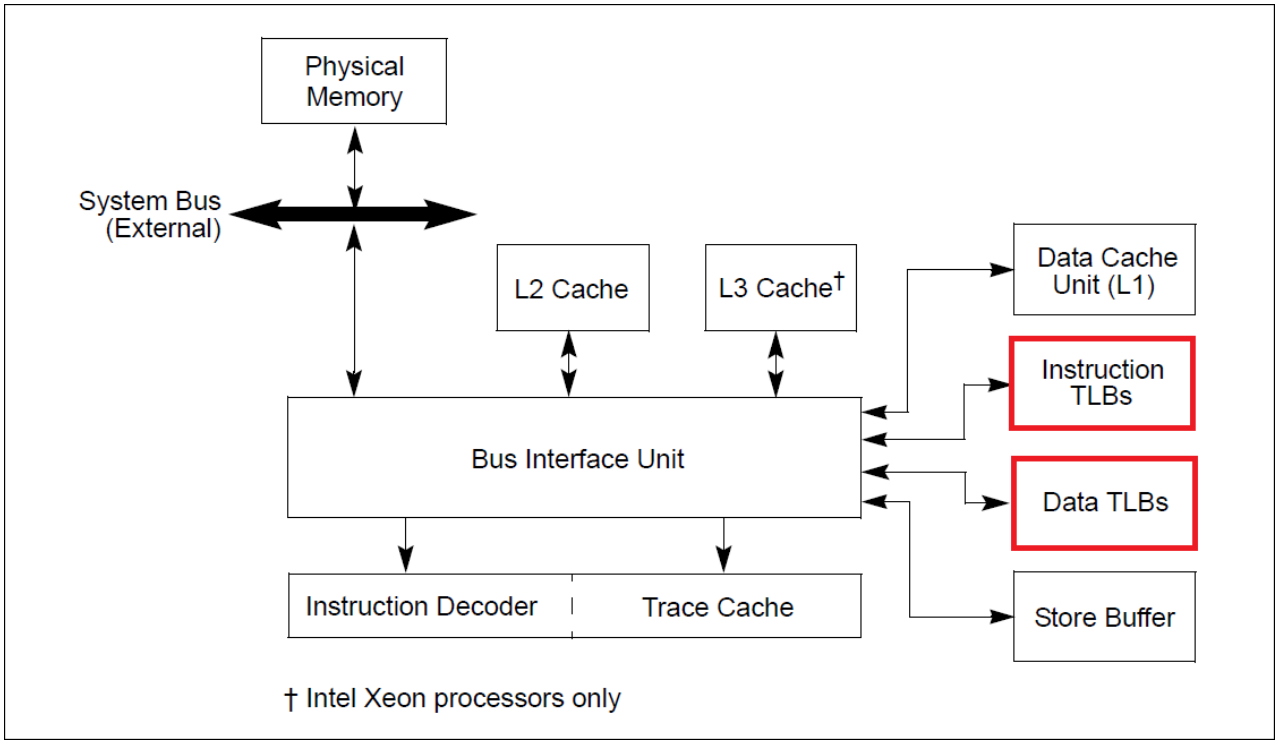
Background: Encrypted Execution

- Originally academic research field looking at adding encryption to instruction set architectures (ISAs)
- Becoming part of Intel ISA in SGX (Skylake & newer)
- Last year's topic of mine was HARES: a way to perform encrypted execution on COTS hardware

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Background: Translation Lookaside Buffer (TLB)

- **TLB is physically two separate entities, one for code, one for data**



Background: AES-NI

- In response to software-based caching attacks on AES, Intel released instruction set to support AES
- Hardware logic is faster, and more protected
- Supports 128-bit and 256-bit AES
- Provides primitives, still requires engineering to make a safe system on top of these

Background: HARES

- **Used split-TLB to provide an AES-NI/TRESOR encrypted capability**
  - AES key stored in CPU debug registers
- **Transparently segregates code and data fetches to different regions of memory**
- **Data fetches are routed to encrypted pages, preventing reverse-engineering**
- **Instruction fetches are routed to decrypted execute-only pages for seamless execution**

Background: Encrypted Execution II

- Possible to significantly increase RE challenge to improve cross-process introspection
- Prevent detection of which PUF challenges are used
- Provide ability to create opaque soft-TPM application

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Encrypted Execution

- SGX or HARES can use a key to bootstrap an encrypted *enclave*
  - Use chain-of-trust to measure and seal enclave to system configuration
- Code running within is opaque even to privileged system software
- RE of enclaves is extremely difficult
  - Store PUF challenges within
  - Prevent loss of TTP/burning capabilities

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

With Our Powers Combined…

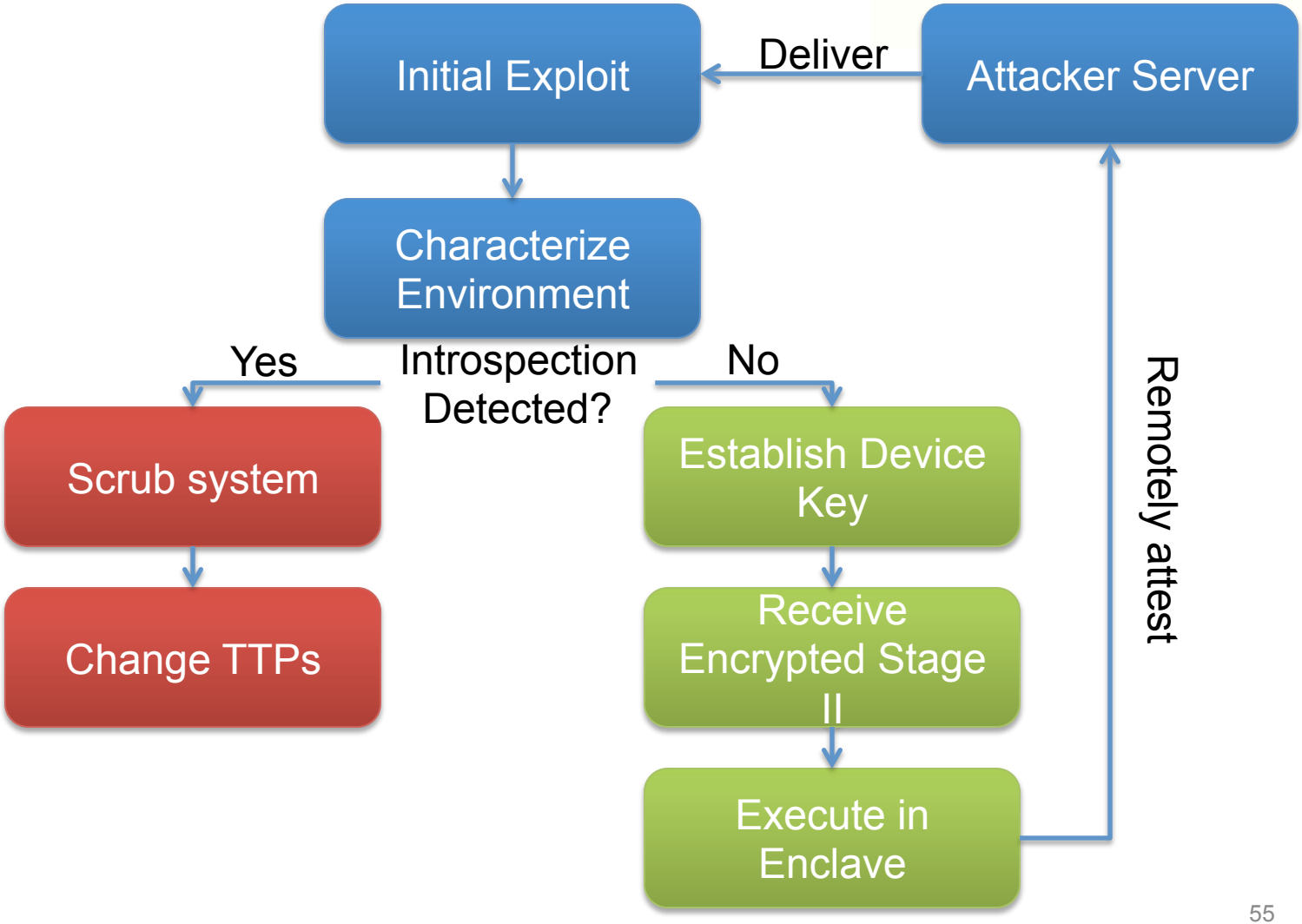
With Our Powers Combined…

devastating capability, revolutionary advantage

- Now we have:
  - Observer Effect: Way to ensure we are executing privately
  - Feng Shui: Gotten cozy in our new home and know the unique features of our device
  - Encrypted execution
- Putting them together allows you to bootstrap trusted implant

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Putting It All Together



Concluding Remarks

- In an advanced red-team exercise, techniques are needed to shield TTPs from discovery
- Features present that can be exposed through low-level hardware access provide a number of building blocks for trust typically needing dedicated ICs (e.g., TPM)
- These can be composed to bootstrap a soft-TPM in order to provide a root-of-trust for code on COTS systems

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Concluding Remarks II

- This technology is just that, technology
- Does not proscribe certain morals or how to use, can be used for both offensive and defensive applications
  - Trusted implants
  - Adding trust to legacy HW appliances that have no trusted computing HW
  - Operating in contested networks through dynamic trust
- Future work into PKI for offensive networks needed

5460 S. Quebec St, Suite 300, Greenwood Village, CO 80111 | +1 315.240.0127 | http://ainfosec.com

Questions

devastating capability, revolutionary advantage

Thank you all for your time!

References

- Cache Teller
  - http://pqdtopen.proquest.com/doc/1086215959.html?FMT=ABS
- PUFs
  - https://spqr.eecs.umich.edu/papers/holcomb_PUFs_date14.pdf
- Shamir's Secret Sharing
  - https://en.wikipedia.org/wiki/Shamir's_Secret_Sharing
- Error-correcting Codes
  - https://en.wikipedia.org/wiki/BCH_code

References II

- Paranoid Fish
  - https://github.com/a0rtega/pafish
- Conqueror
  - http://roberto.greyhats.it/pubs/dimva10.pdf
- HARES
  - http://jacobtorrey.com/HARES-WP.pdf
- Intel SGX
  - https://software.intel.com/en-us/isa-extensions/intel-sgx