Software Engineering Security

05 NOV 2019

Gigamon®

Introductions

William Peteroy

@wepIV

Used to hack stuff—now build things and security research

Security

Our Definition of Security

- Engineer and build systems to prevent attackers from doing things we don't want them to do
- When we're building software
  - Build software in a way that's usable but not vulnerable to an attacker

Why should anyone care?

The weapon's target was Ukraine. But its blast radius was the entire world. "It was the equivalent of using a nuclear bomb to achieve a small tactical victory," Bossert says.

The release of NotPetya was an act of cyberwar by almost any definition—one that was likely more explosive than even its creators intended. Within hours of its first appearance, the worm raced beyond Ukraine and out to countless machines around the world, from hospitals in Pennsylvania to a chocolate factory in Tasmania. It – crippled multinational companies including Maersk, pharmaceutical construction company Saint- manufacturer Reckitt Bencki costs. It even spread back to Rosneft.

"The result was more than $10 billion in total damages…

Business

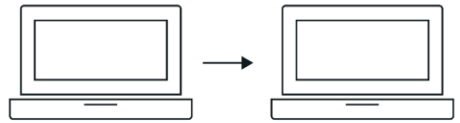
Google uncovers 2-year iPhone hack that was 'sustained' and 'indiscriminate'

The attack may be one of the largest ever on iPhone users, exposing messages, address books, GPS data and more.

Gigamon®

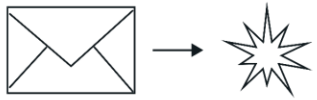
Attacker Lifecycle

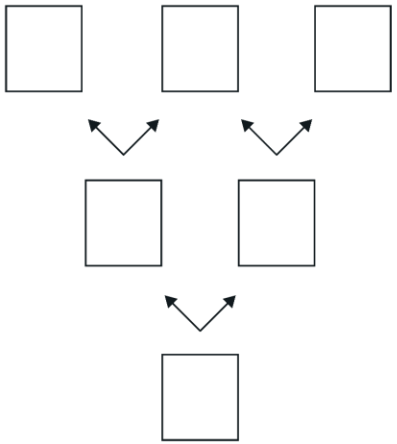
Attacker Lifecycle



Persistence

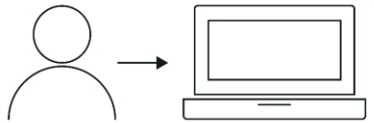
Initial Access
- Delivery
- Exploitation
- Code Execution



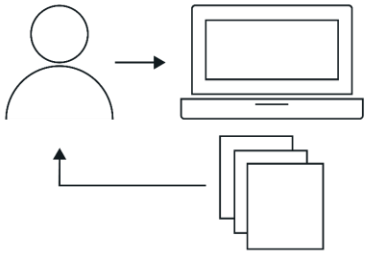


Lateral Movement
- Exploitation
- Credential Theft and Reuse

Access to Data or Systems
- Application Access
- IT Asset Access
- Malware
- Command and Control



Actions on Target
- Local Enumeration
- Privilege Escalation
- Environment Enumeration
- Data Theft



What does this mean for Software Engineering?

- Knowledge of the attacker lifecycle influences design
  - Security mitigations (DEP / ASLR / etc)
  - Understand our code will be attacked
  - Understand our users will be attacked
- Threat model extensively

Software Security

Core Software Security Goals

1. Build robust, secure code
2. Use secure defaults
3. Keep it simple
   - Don’t roll your own crypto

- “CIA Triad” + Extended Properties
  - Confidentiality
  - Integrity
  - Availability
  - Authentication
  - Authorization
  - Nonrepudiation

Challenges in Software Security

Core Challenges in Software Security

- Performance
- Usability
- Complexity
- Threat Modelling
- Politics

Performance

How bad design decisions created the least secure driver on Windows

Thomas Garnier Follow

Aug 21, 2016 · 5 min read

This driver is called win32k, it manages the user interface of Windows. This post will discuss the multiple bad ideas that are part of this driver.

How bad is it?

It is hard to get a bug count estimate. Each page unique and it can be hard to infer affected modu patch and look at the files, when the links still w

Designed with trust in mind

In Windows NT 3.5, the UI was managed by a user-mode module in the CSRSS system process. The original design was overhauled in Windows NT 4 due to bad performance. The win32k driver was created almost as an extension of its user-mode counterpart with an obvious trust between the two of them. This new design was much faster and flexible though hard to make reliable and secure.

Usability

- We can make things secure by default…
- … up to a point

What is Protected View?

Excel for Office 365, Word for Office 365, PowerPoint for Office 365, Excel 2019, More...

Files from the Internet and from other potentially unsafe locations can contain viruses, worms, or other kinds of malware that can harm your computer. To help protect your computer, files from these potentially unsafe locations are opened as read only or in Protected View. By using Protected View, you can read a file and see its contents and enable editing while reducing the risks.

NEWS

Email attacks exploit unpatched Microsoft Word vulnerability

Attackers have been exploiting a zero-day vulnerability in Microsoft Word since January to infect computers with malware

Complexity

- In general security is good
- Some security can add un-necessary complexity

Project Zero

News and updates from the Project Zero team at Google

Home

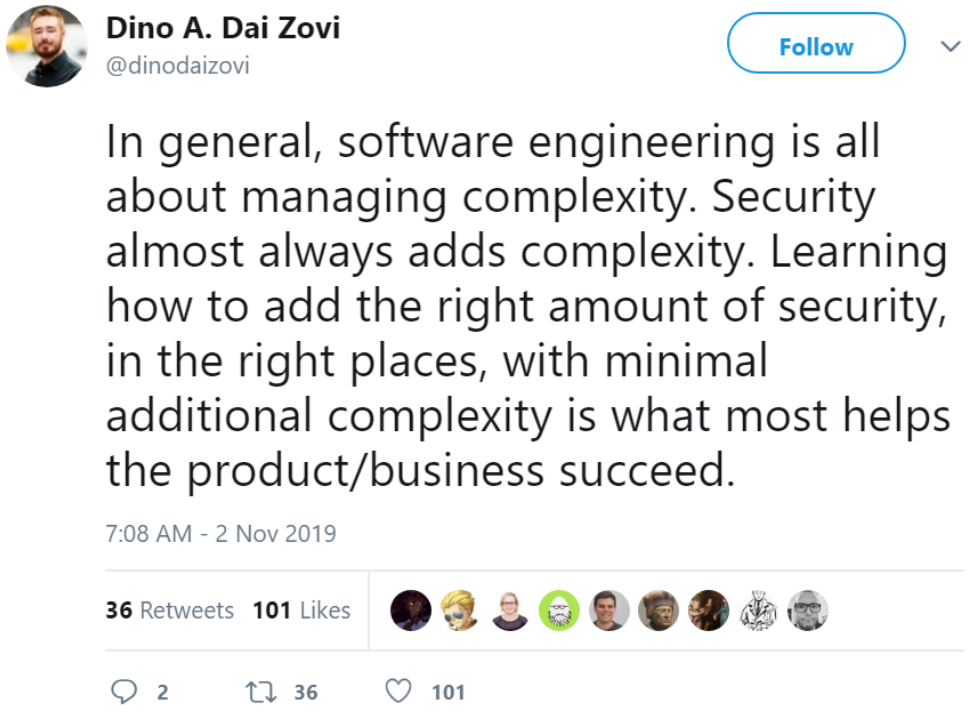
Tuesday, June 28, 2016

How to Compromise the Enterprise Endpoint

Posted by Tavis Ormandy.

Symantec is a popular vendor in the enterprise security market, their flagship product is Symantec Endpoint Protection. They sell various products using the same core engine in several markets, including a consumer version under the Norton brand.

Today we're publishing details of multiple critical vulnerabilities that we discovered, including many wormable remote code execution flaws.



Dino A. Dai Zovi
@dinodaizovi

Follow

In general, software engineering is all about managing complexity. Security almost always adds complexity. Learning how to add the right amount of security, in the right places, with minimal additional complexity is what most helps the product/business succeed.

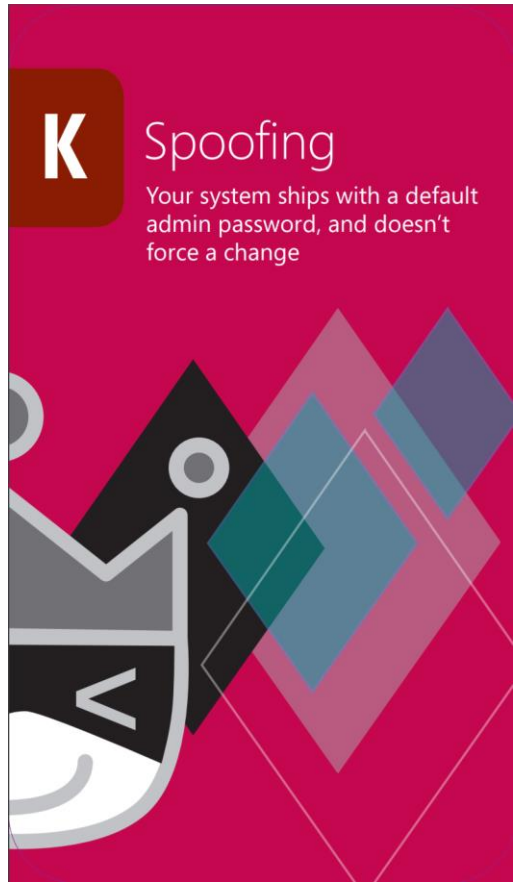
7:08 AM - 2 Nov 2019

36 Retweets 101 Likes

2 36 101

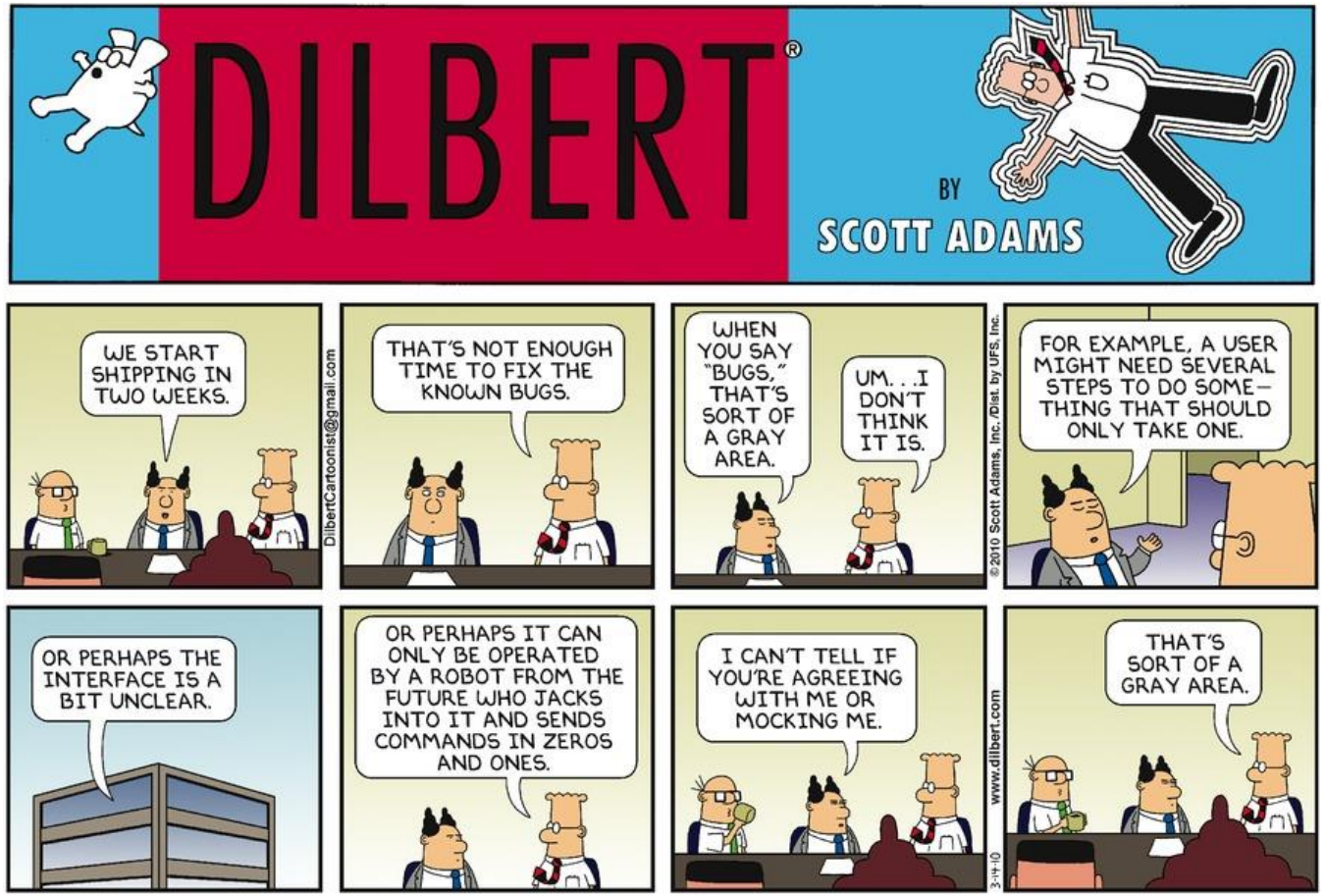
Threat Modelling

- Identify threats before we ship a product
- Evaluate design from an attacker perspective
- STRIDE
  - **S**poofing
  - **T**ampering
  - **R**epudiation
  - **I**nfo Disclosure
  - **D**enial of Service
  - **E**levation of Privilege



Politics

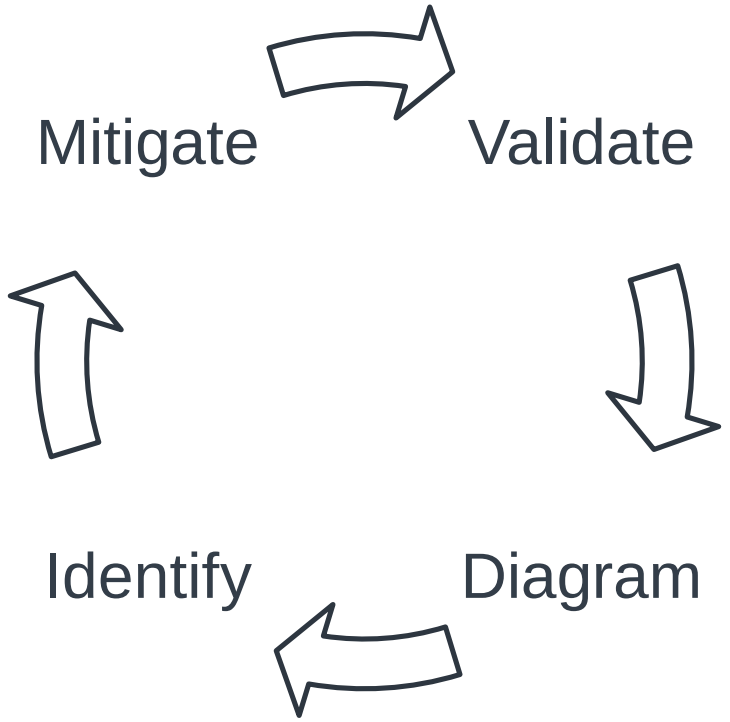
- Software Engineering VP's get paid to deliver product features on a timeline
- Features typically impact a business line / the bottom line
- When that timeline is threatened, their bonuses are threatened
- This is a big reason we have SDLC



Threat Modelling

Threat Modelling

- Identify security objectives
- Identify relevant threats
- Identify relevant vulnerabilities and countermeasures



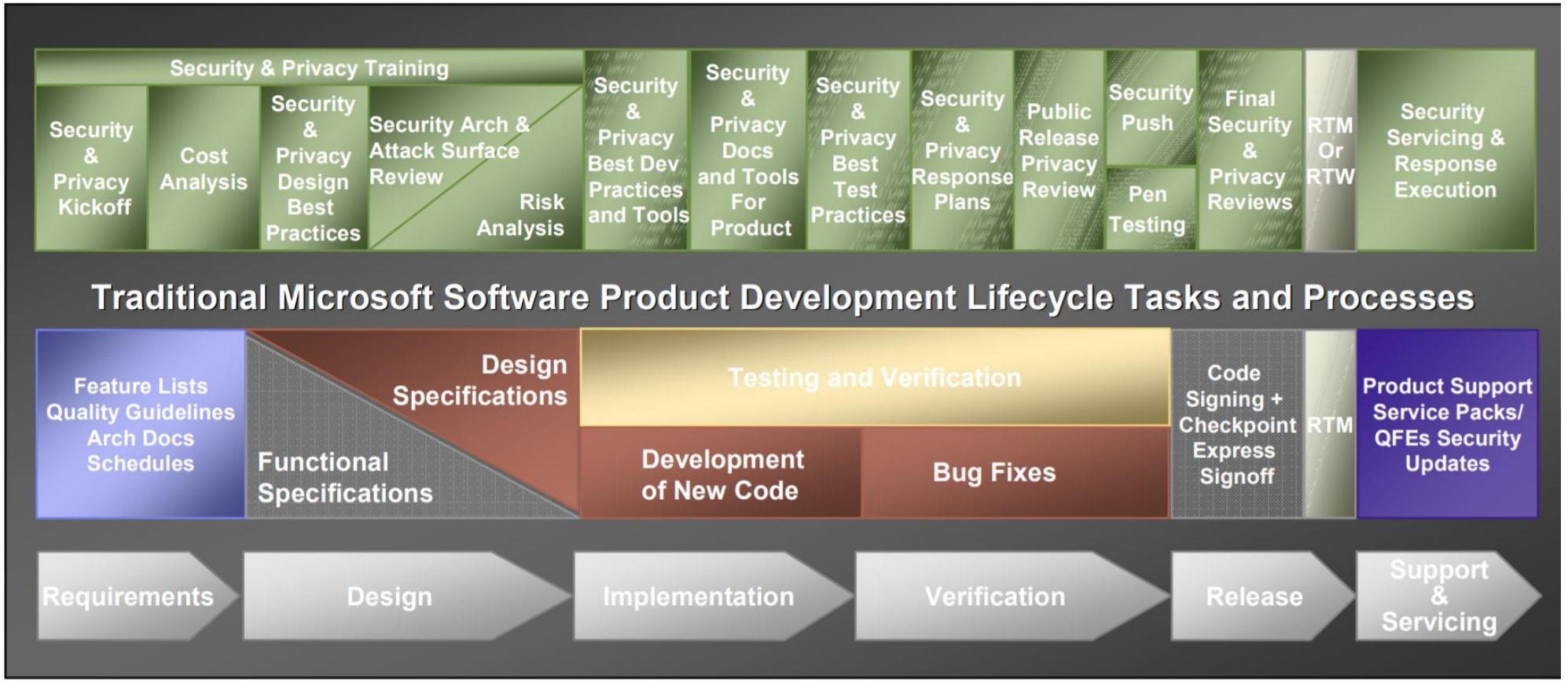
Threat Modelling Challenges

- Timeliness
  - Formal SDLC?
  - Waterfall? Agile?
- Threat coverage
  - Which threats? Why?
- Threat / Vulnerability Relevance
  - Understand context from attacker lifecycle

SDLC Primer

Secure Development Lifecycle

- ISO 27207
- Largely defined by Microsoft



SDL Bug Bar

- Criteria that establish minimum quality levels
- Define and document org. standards
- Align between security and engineering
- Reference to talk about bugs in context of other bugs

Server Please refer to the Denial of Service Matrix below for a complete matrix of server DoS scenarios.	
SEV 1	• Elevation of Privilege – The ability to either execute arbitrary code OR obtain more privilege than intended ○ Remote Anonymous User

SEV 2	• Denial of Service
	  ○ Must be “easy to exploit” by sending small amount of data or be otherwise quickly induced
	  ○ Anonymous
	• Elevation of Privilege – The ability to either execute arbitrary code OR obtain more privilege than intended
	  ○ Remote Authenticated User
	  ○ Local Authenticated User (Terminal Server)
	• Information Disclosure (Targeted)
	  ○ Cases where the attacker can locate and read information *from anywhere* on the system, including system information, that was not intended/designed to be exposed
	• Spoofing
	  ○ Computer connecting to server is able to masquerade as a different user or computer of his/her choice *using a protocol* that is designed and marketed to provide strong authentication
	• Tampering
	  ○ Permanent modification of any user data or data used to make trust decisions *in a common or default scenario* that persists after restarting the OS/application.

SEV 3	• Denial of Service
	  ○ Anonymous
	  ▪ Temporary DoS without amplification in a default/common install
	  ○ Authenticated
	  ▪ Permanent DoS
	  ▪ Temporary DoS with amplification in a default/common install
	• Information Disclosure (Targeted)

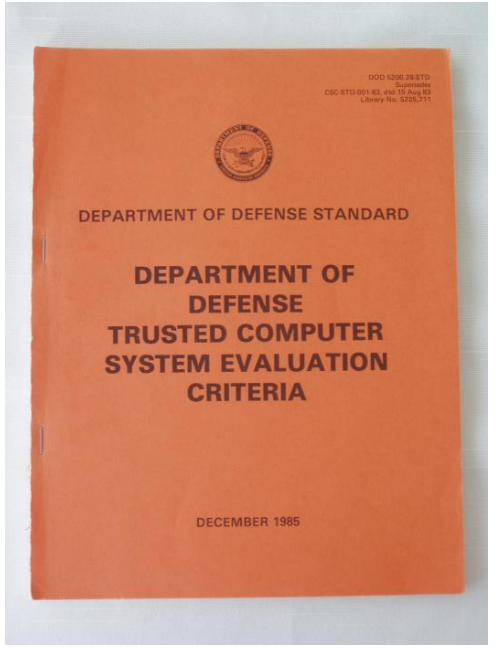
Gigamon®

Case Study 1

Side Channel Attacks

Side Channel Attacks

- We've known that they're a potential issue since TCSEC aka the "Orange Book" in 1983
  - There shall be no covert storage channels with a capacity exceeding 100 bits/second;
  - All covert storage channels with capacities exceeding 10 bits/second shall be auditable;
  - All covert storage channels with capacities exceeding 1 bit/second shall be described in the product's covert channel analysis.



Back to the Future

1983…2019 ?

- Fun with shared hardware…aka… “the cloud”
- “Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds” (Ristenpart, Tromer, et. Al)
  - What’s the core issue / root cause
  - Do y’all think this is a big deal?

1983…2019 - Discussion

- Fun with shared hardware…aka… “the cloud”
- “Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds” (Ristenpart, Tromer, et. Al)
  - What’s the core issue / root cause
    1. Security through obscurity
    2. Assumption that people would behave w/ a shared state (side channels) – probably thought noise would make it impossible
  - Do y’all think this is a big deal?
    - .06- .2 bps of side channel
    - In graphics cards we see side-channel bandwidth measured in GBPS and reliable at 6kbps (*Rendered Insecure* – Naghibijouybari, Qian, et. Al)

Q and A

Where to learn / do more with security

- Reverse Engineering
  - CrackMes
- Finding Vulnerabilities
  - Bug Bounties
- Competing / having fun with CTFs
  - “Intro to CTFs”
- My contact info
  - @wepIV

Thank You