RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: PS-T11

Lessons Learned: 50 Years of Mistakes in Cybersecurity

Steve Lipner

Executive Director
SAFECode
@Lipner

Introduction

- Started working on “computer security” in 1970, never stopped
- Joined Microsoft in 1999
  - Created/led Security Development Lifecycle (SDL)
  - Retired in 2015
- Executive director of SAFECode since late 2016
- This presentation is about mistakes – what I wish I’d done differently in the last fifty years
- I’ll wrap up with “lessons learned” and suggested actions

Relevant Career History

- 1970 – 76 The MITRE Corporation 

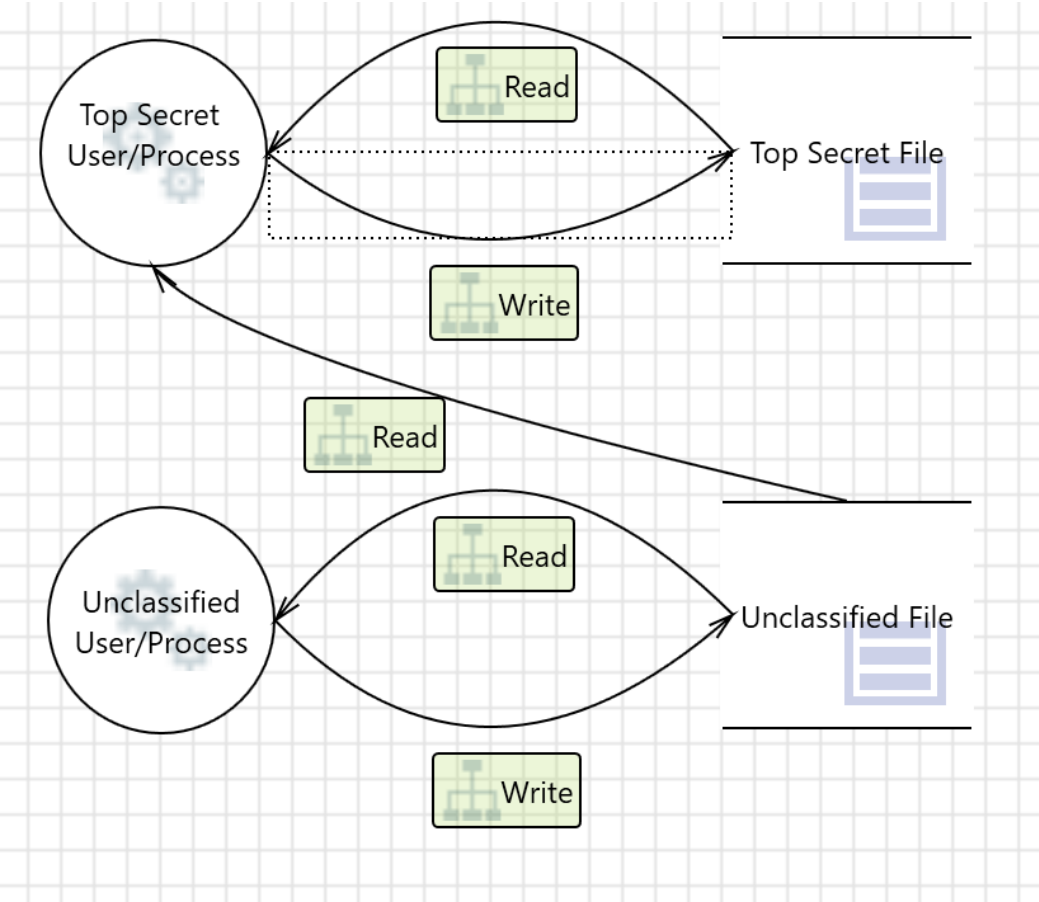
- 1981 – 92 Digital Equipment Corporation 

- 1994 – 97 Trusted Information Systems 

- 1999 – 2015 Microsoft 

Mistake 1 – Bell-LaPadula

- Objective was “Multilevel Security” (MLS)
- Driven by DoD model of information security
- Model was a major breakthrough

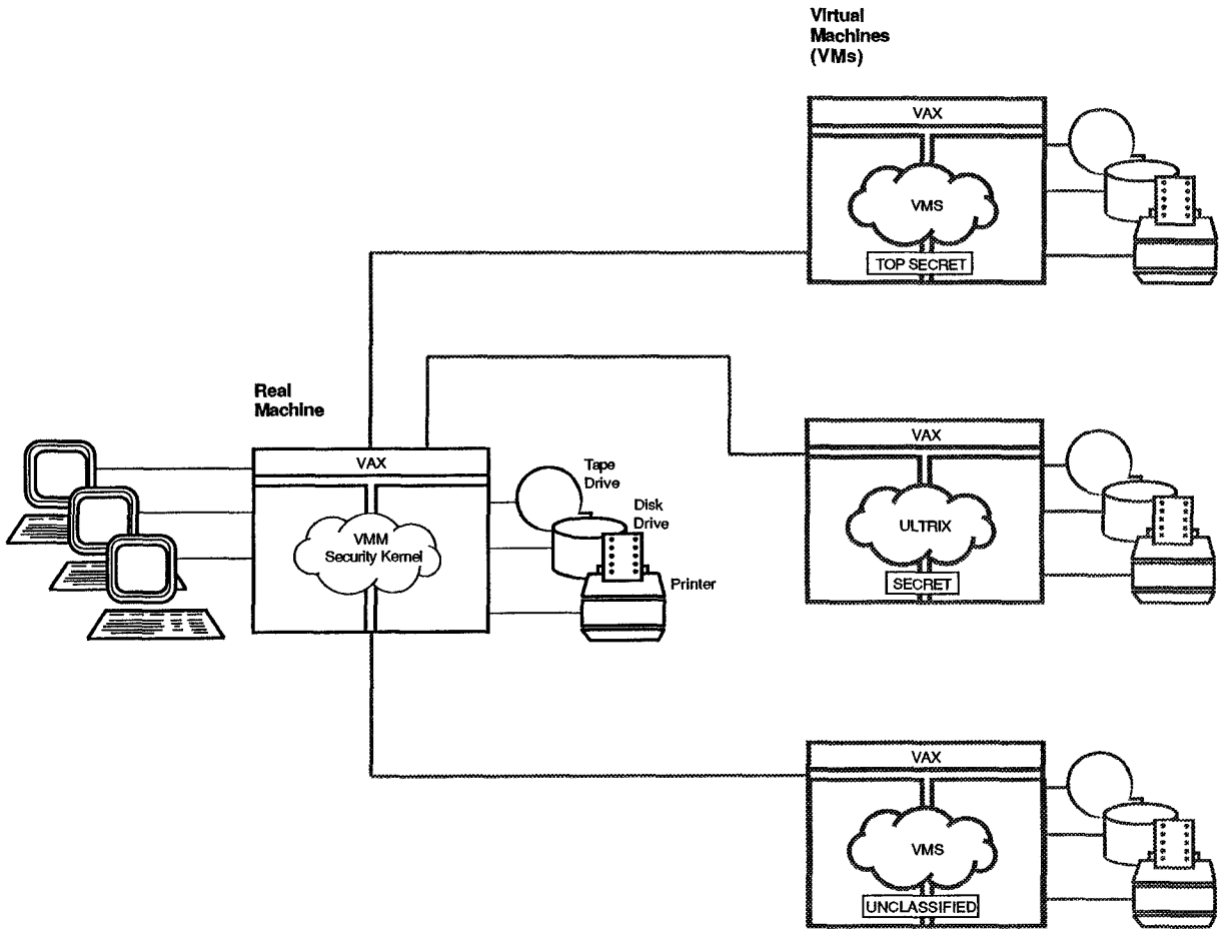


Mistake 1 – Bell LaPadula

- Model worked for one time-shared Multics system
- I believed it was a general solution for MLS
- In practice too many reasons to violate the model
- We invented pop-up fatigue: do you want to downgrade this message?
  - "Allow or deny?"…
- Consequences discussed with next mistake…

Mistake 2: VAX SVS

- DoD evaluation criteria specified Bell-LaPadula, high assurance
- I sold DEC on building an A1 system
- Fully functional VMM based VMS and Unix time-sharing

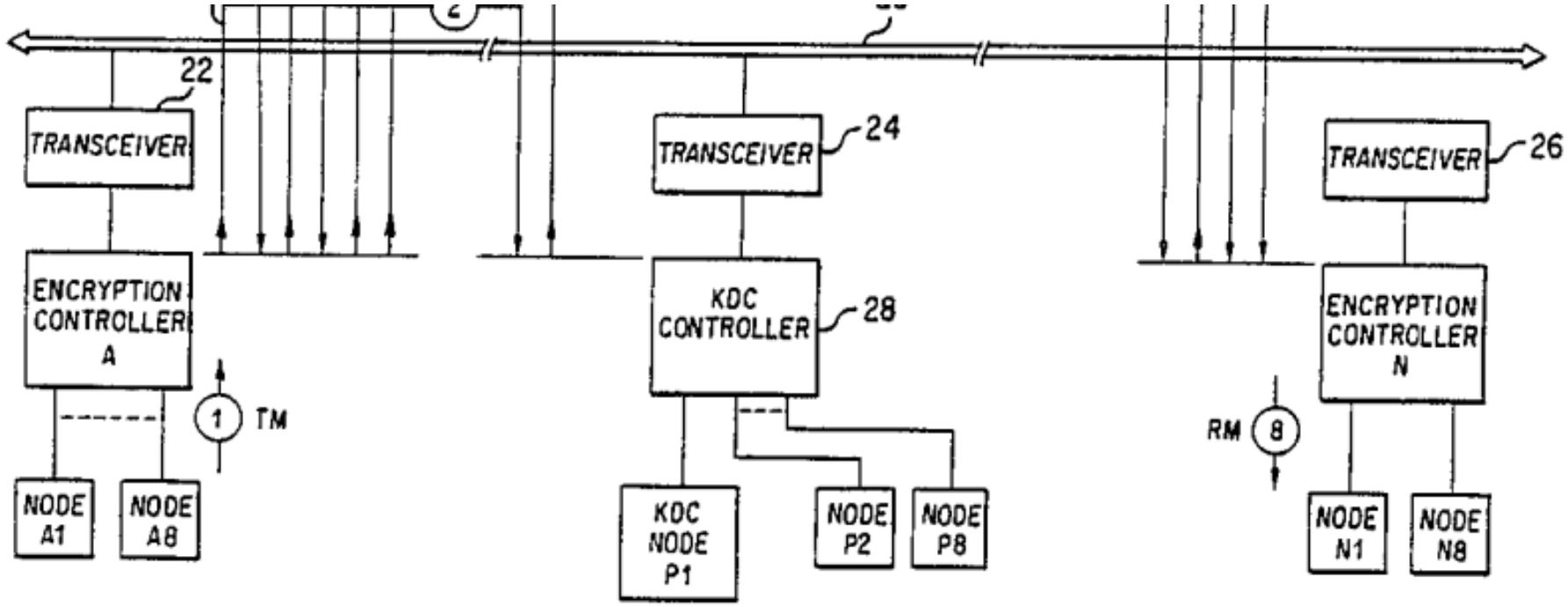


Mistake 2 – VAX SVS

- Development was challenging (code minimization, performance, adversarial evaluation process), ran well behind schedule
- By the time the system was “done” and in Beta test, users had moved to PCs, networks, and GUIs
- “Nobody wanted a system that secure”
- I cancelled the product in 1990
- SVS development cost DEC about $20M

Mistake 3: DESNC

- DEC adopted Ethernet in the mid-1980s – great connectivity, no security
- DEC product family used multiple transport protocols – DECNet, LAT…



Mistake 3: DESNC

- DESNC approach had both technological and fundamental problems
  - Hardware was too costly and performance too limited
  - DECnet, terminal concentrators nearing end of life; IP became dominant
- “Right answer” would have been software implementation at the IP layer – if DEC had been committed to IP
- Development cost $M – product was sold but in very small volume

Mistake 4: Gauntlet Firewall

- Gauntlet was an early application proxy firewall product
  - Security based on complete mediation and minimal trusted code
- I made two big mistakes
  - Didn't invest soon or deeply enough in a management GUI (see "minimal code")
  - Ported to NT, but network transparency would have required a complex hack (see "trusted code")
- Gauntlet was moderately successful but Checkpoint Firewall 1– with GUI and NT support – ate our lunch

Mistake 5: Inventing a Key Escrow System

- TIS was pro-crypto export, opposed the Clipper Chip
- CEO asked to invent a way to invalidate USG claims
  - Need for non-public algorithm
  - Need for hardware implementation
- I did
- TIS then decided to commercialize key escrow
  - Invested in building and selling it to a market dictated by USG mandate

Mistake 5: Inventing a Key Escrow System

- Investment distracted from Gauntlet business
- Government abandoned key escrow mandate

Mistake 6: "Think Like a Hacker"

- Windows Security Push of 2002 was a major Microsoft commitment and major success
- Planning period about eight weeks– inventing on the fly
- Some of our training tried to teach product engineers to "think like hackers" – find vulnerabilities, invent attacks
  - That part mostly didn't work…
  - Distracted some engineers from reviewing code and tool outputs
- But the security push showed us things that would work and became parts of the SDL

Lesson Learned – Apply or Cry

- The customer is right – even if he/she is wrong
- Usability is important
- Security isn't everything
- Think carefully before you decide government will create a market
- Faster to market is better than perfect
- You may have to evade organization norms to succeed

RSAConference2020

Questions?

Lipner@safecode.org