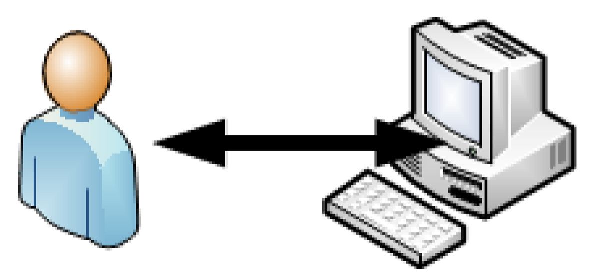
Detection of malicious keyloggers in virtual desktop environments

Zachary Estrada, Phuong Cao, Zbigniew Kalbarczyk, Ravishankar Iyer

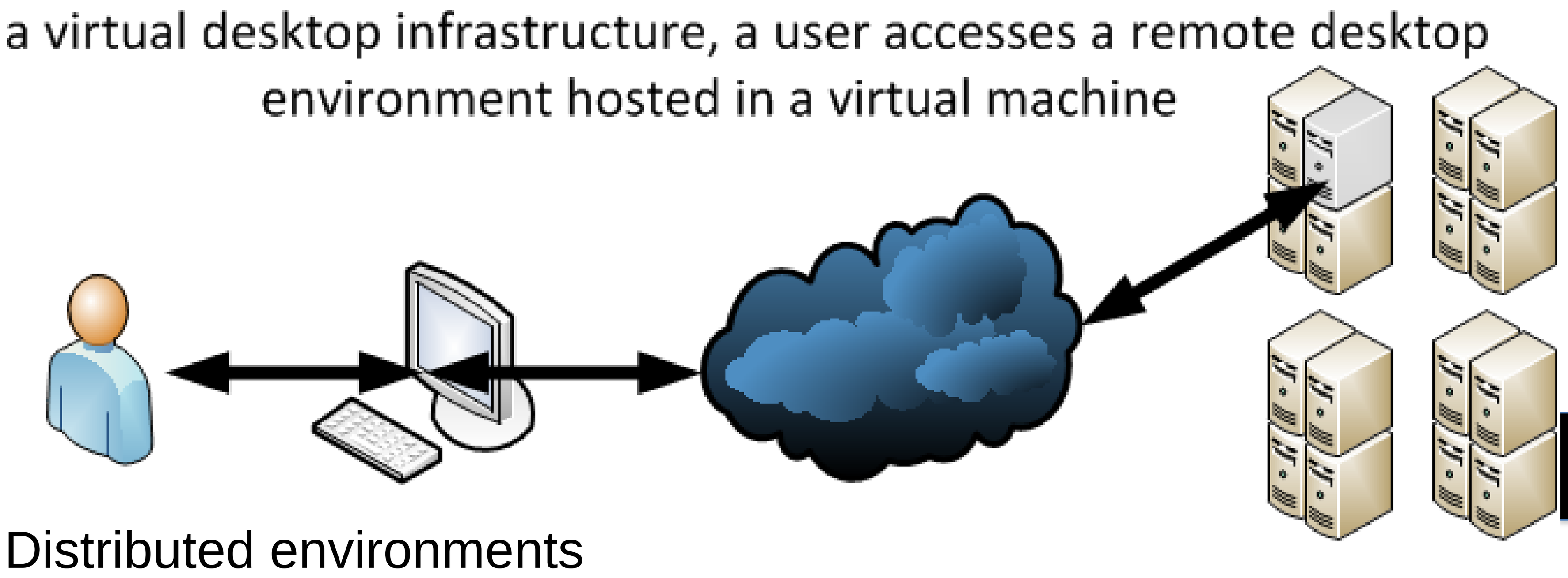
Enterprise environments use Virtual Desktop Integration (VDI) to provide workstations for employees. In VDI, each user's desktop environment is hosted on a remote Virtual Machine (VM) inside a datacenter or cloud. While VDI provides security benefits due to the isolation offered by virtualization, such environments are vulnerable to many software-based attacks as traditional desktop environments. One such attack is a software *keylogger* that records keystrokes inside the guest OS. This project leverages the VDI hypervisor to detect software-based keyloggers running in VMs.

Virtual Desktop Infrastructure

In a traditional desktop infrastructure, a user has direct physical access to the desktop environment



In a virtual desktop infrastructure, a user accesses a remote desktop environment hosted in a virtual machine



+ Distributed environments
+ More robust against local attacks
- Still vulnerable to software-based attacks

Process-based Keyloggers

Process based keyloggers run as processes inside the victim OS. These *keyloggers* represent a significant threat as they are widely available and easy to install. Because of the sensitive information that can be collected using process-based *keyloggers*, they represent an important component of spyware applications.

Detecting Keyloggers with Process Traces

Method

- VM monitoring allows us to detect when a keystroke is pressed. We identify process changes by observing virtual address space changes whenever the CR3 (a control register in Intel microprocessor) register's value changes.

Observation

- After a keystroke is passed into the guest OS, a keylogger process responds to consume that keystroke.

Detection (intuition)

- The more processes there that respond to a keystroke, the more likely a keylogger is present.

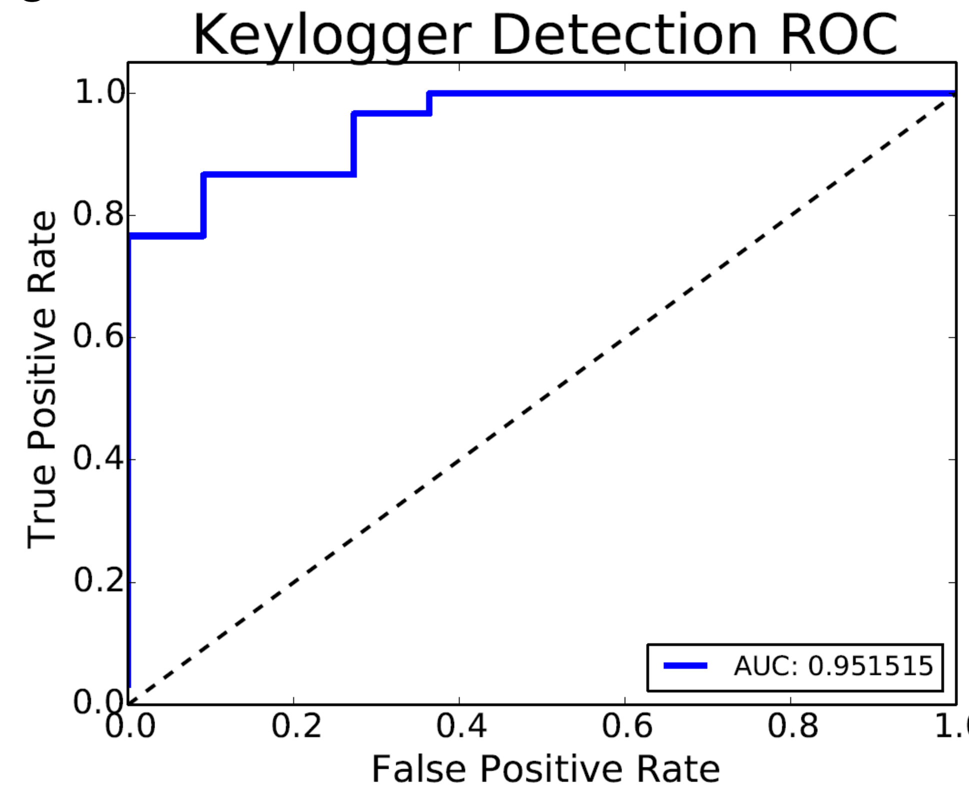
Timestamp	Event
Jun 22 15:14:16 [605468.649985]	Keypress! CR3: 0x185000
Jun 22 15:14:16 [605468.650971]	CR3: 0x21310000
Jun 22 15:14:16 [605468.652154]	CR3: 0xa688000
Jun 22 15:14:16 [605468.653126]	CR3: 0x21310000
Jun 22 15:14:16 [605468.653925]	CR3: 0xa688000
Jun 22 15:14:16 [605468.654745]	CR3: 0x21310000
Jun 22 15:14:16 [605468.655560]	CR3: 0xa688000
Jun 22 15:14:16 [605468.656386]	CR3: 0x215ed000

Approach 1: Responsiveness Score

- Based on those process changes, we calculate a responsiveness score for each process:

$$R_k(\mathrm{CR3}) = e^{-\ln(2)\frac{t_{CR3}-t_k}{t_{1/2}}}$$

- The responsiveness score uses the difference between the time of the CR3 change (t_{CR3}) and the last keystroke (t_k) and is correlated to the number of processes responding to a keystroke, which is expected to be higher in the presence of a keylogger.
- We measured the mean responsiveness for various workloads using four keyloggers and obtained the following Reciever Operating Characteristics:



Approach 2: Bayesian Detection

Assumptions

- A process is a keylogger with probability θ; we define the first time a process appears after a keystroke as its position *k* in the scheduling list of *n* processes.
- The Prior probability for θ follows a Beta distribution:

$$P(\theta;\alpha,\beta) = Beta(\theta;\alpha,\beta) = \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

- The likelihood of the observed position of the process follows the Binomial distribution:

$$L(n,k,\theta) = Binom(n,k,\theta) = \binom{n}{k}\theta^k(1-\theta^k)$$

Approach

- At the beginning, the pdf of θ is peaked at 0.5 ($\alpha = \beta = 1$), since without any information, we assume that the process is equally likely to be a keylogger or a benign process
- After each key press, the parameters are updated to reflect new belief about whether each process is a keylogger:

$$P(\hat{\theta}) = L(n,k,\theta)P(\theta,\alpha,\beta)$$

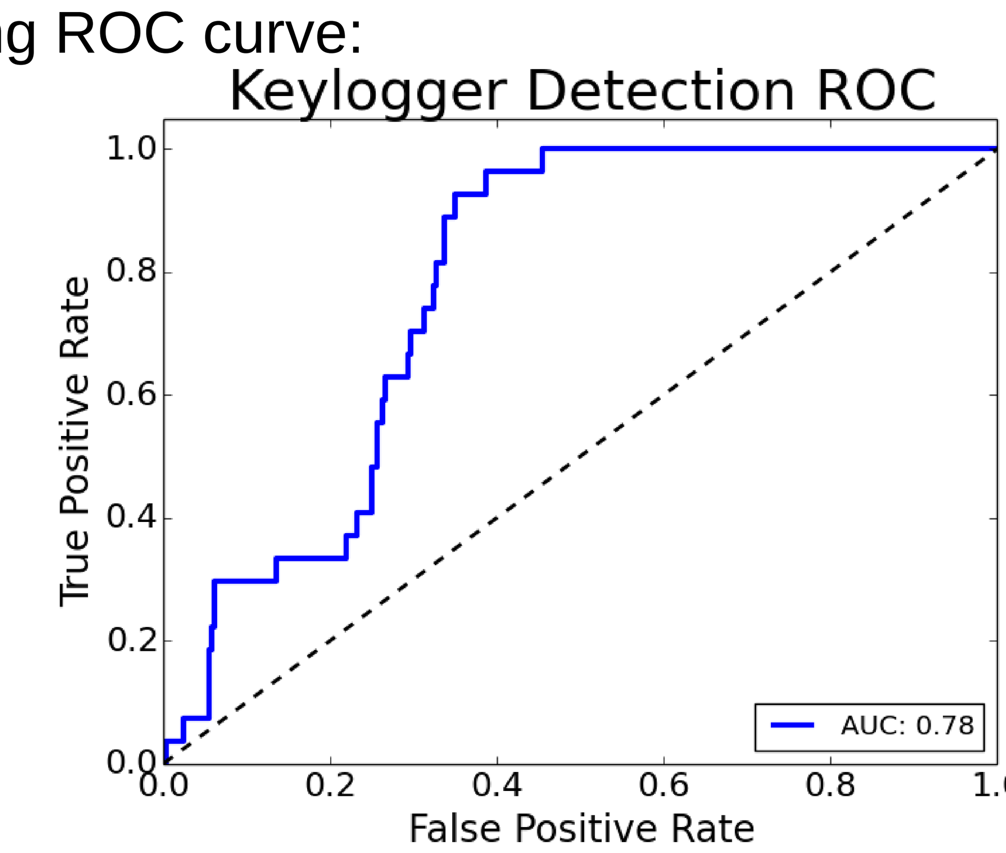
where *n* is the total number of processes and *k* is the position of the process

- Since the Beta distribution is the conjugate prior of the binomial distribution, the posterior probability is given by:

$$P(\hat{\theta}) = Beta(\sum_{k_i} + \alpha, \sum_{n-k_i} + \beta)$$

where k_i is the position of the process after key press *i*.

- We used the same data as in Approach 1 and obtained the following ROC curve:



SCIENCE OF SECURITY
VIRTUAL ORGANIZATION
Funded by the National Security Agency.

INFORMATION TRUST INSTITUTE