Redemption: Real-time Protection Against Ransomware at End-Hosts

Written By Amin Kharraz and Engin Kirda

RAJSHAKHAR PAUL

Outlines

- Introduction
- Existing works
- Contribution
- Threat Model
- Design Overview
- Evaluation
- Limitations

Outlines

- Introduction
- Existing works
- Contribution
- Threat Model
- Design Overview
- Evaluation
- Limitations

Introduction

Ransomware

- What is Ransomware?

- A type of malware that prevents users from accessing their data by encrypting those and demands ransom payment in order to regain access.

- The earliest versions of ransomware were developed in the late 1980s

- Attackers generally order the payment via cryptocurrency

Ransomware

- One of the biggest security threats of current era
- Hospitals and healthcare industries are mainly affected

Data Retrieval

How can I get back my data?

Data Retrieval

How can I get back my data?

- Easiest solution: creating back up of important data
- If system is compromised by ransomware, retrieve data from back up

Data Retrieval

I don’t have any back up of my data ☹

How can I retrieve??

Data Retrieval

- Law enforcement agencies and security firms have launched program to assist ransomware victim in retrieving their data without paying ransom
- Used reverse analysis of the cryptosystems used by malware to extract secret keys
- Tried to find design flaws of encryption system
- Work for weak cryptography

But the attackers are smart and use strong cryptography

Prevention

How can I prevent this?

Prevention

How can I prevent this?

- The authors introduce Redemption
- An endpoint approach to defend against unknown ransomware attack and recover lost data
- Two main approaches:
  - An abstract characterization of the behavior of the ransomware attacks
  - Employs a high-performance mechanism to protect and restore all attacked files

Outlines

- Introduction
- Existing works
- Contribution
- Threat Model
- Design Overview
- Evaluation
- Limitations

Existing Works

UNVEIL

- Proposed by Kharraz et al. at 2016
- A dynamic analysis system
- Specifically designed to assist reverse engineers to analyze the intrinsic behavior of an arbitrary ransomware sample
- Not an end-point solution
- No real end-user interaction was involved in their test

Existing Works

CryptoDrop

- Proposed by Scaife et al. at 2016
- The approach is able to detect a ransomware attack after a median of ten file losses
- Main limitation: the tool does not provide any recovery or minimal data loss guarantees

Existing Works

ShieldFS

- Proposed by Continella et al. at 2016
- Similar goal to Redemption
- The authors look into file system layer to find typical ransomware activity
- Rely on cryptographic primitive identification
- Limitation: not resistant to unknown cryptographic function
- Relying on cryptographic primitive identification can result false positive.

Existing Works

PayBreak

- Proposed by Kolodenker et al. at 2017
- Securely stores cryptographic encryption keys in a key vault that is used to decrypt affected files after a ransomware attack
- Intercepts calls to functions that provide cryptographic operations, encrypts symmetric encryption keys, and stores the results in the key vault
- After a ransomware attack, the user can decrypt the key vault with his private key and decrypt the files without making any payment
- Pros: imposes negligible overhead
- Cons: like ShieldFS, it depends on identifying functions that implement cryptographic primitives

Outlines

- Introduction
- Existing works
- Contribution
- Threat Model
- Design Overview
- Evaluation
- Limitations

Contribution

- Presents a general approach to defend unknown ransomware attacks in a transperant manner.
- Shows that efficient ransomware protection with zero data loss is possible
- Presents a prototype implementation for Windows, and evaluate it with real users to show that the system is able to protect user files during an unknown ransomware attack imposing no observable overhead

Outlines

- Introduction
- Existing works
- Contribution
- Threat Model
- Design Overview
- Evaluation
- Limitations

Threat Model

Assumptions:

- Ransomware can employ any standard, popular techniques to attack machines like other types of malware.
- The malicious process can employ any techniques to generate the encryption key, use arbitrary encryption key lengths, or utilize any customized or standard cryptosystems to lock the files
- A user can install and run programs from arbitrary untrusted sources, and therefore, that malicious code can execute with the privileges of the user
- Trusted components: Display module, OS kernel, and underlying software and hardware

Outlines

- Introduction
- Existing works
- Contribution
- Threat Model
- Design Overview
- Evaluation
- Limitations

Design Overview

Redemption has two main components

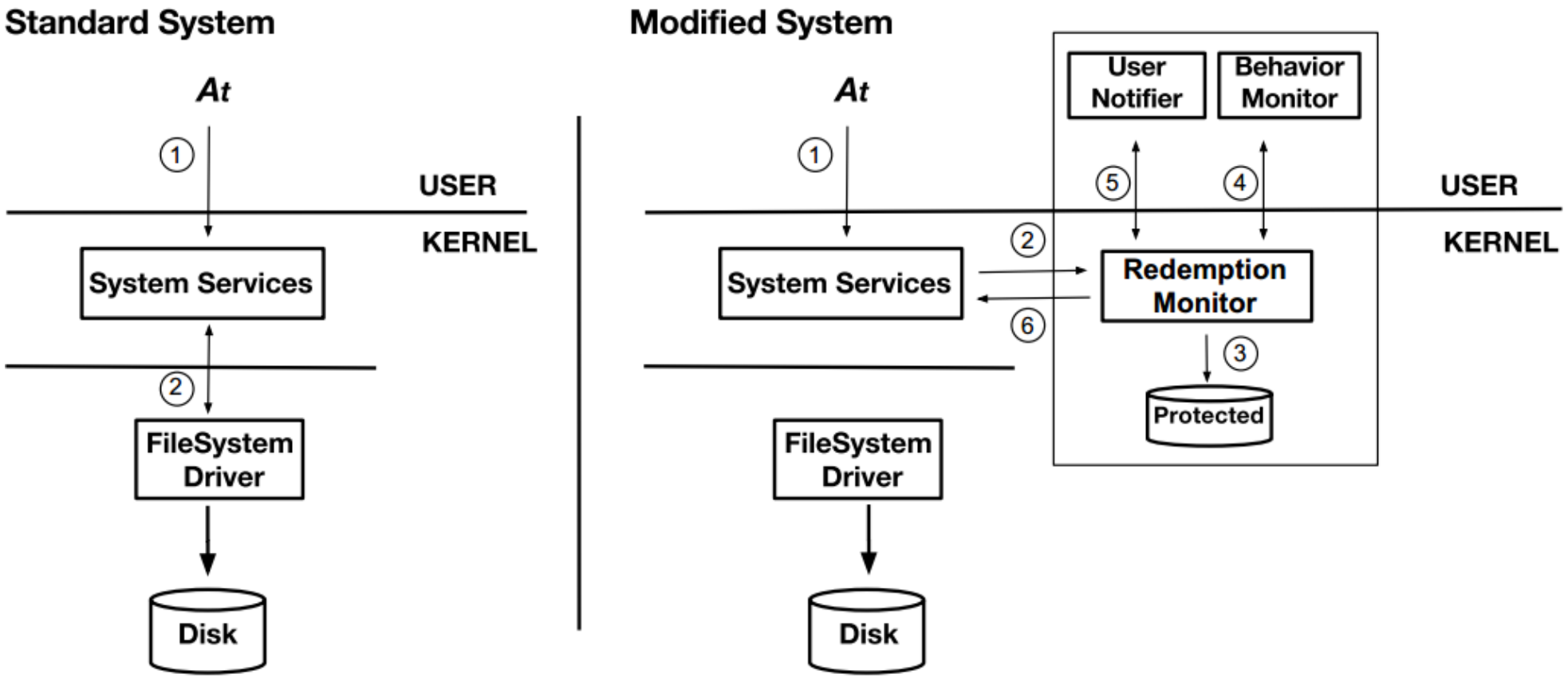
1. A lightweight kernel module

- intercepts process interactions and stores the event
- manages the changes in a protected area

2. Behavioral monitor and notification module

- assigns a malice score to a process
- notify the user about the potential malicious behavior of a process

Design Overview

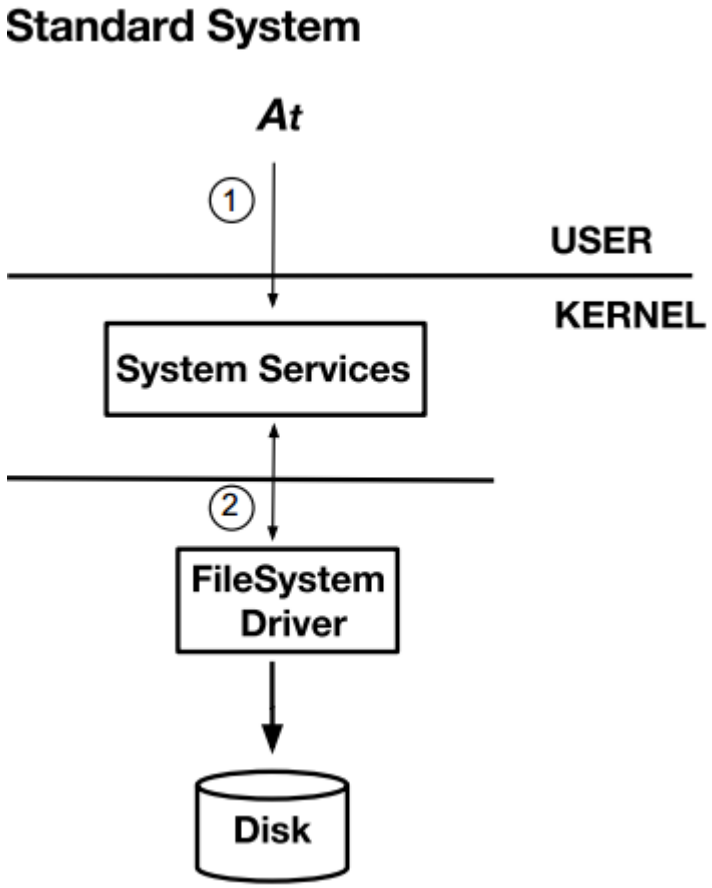


Design Overview

- In standard system,

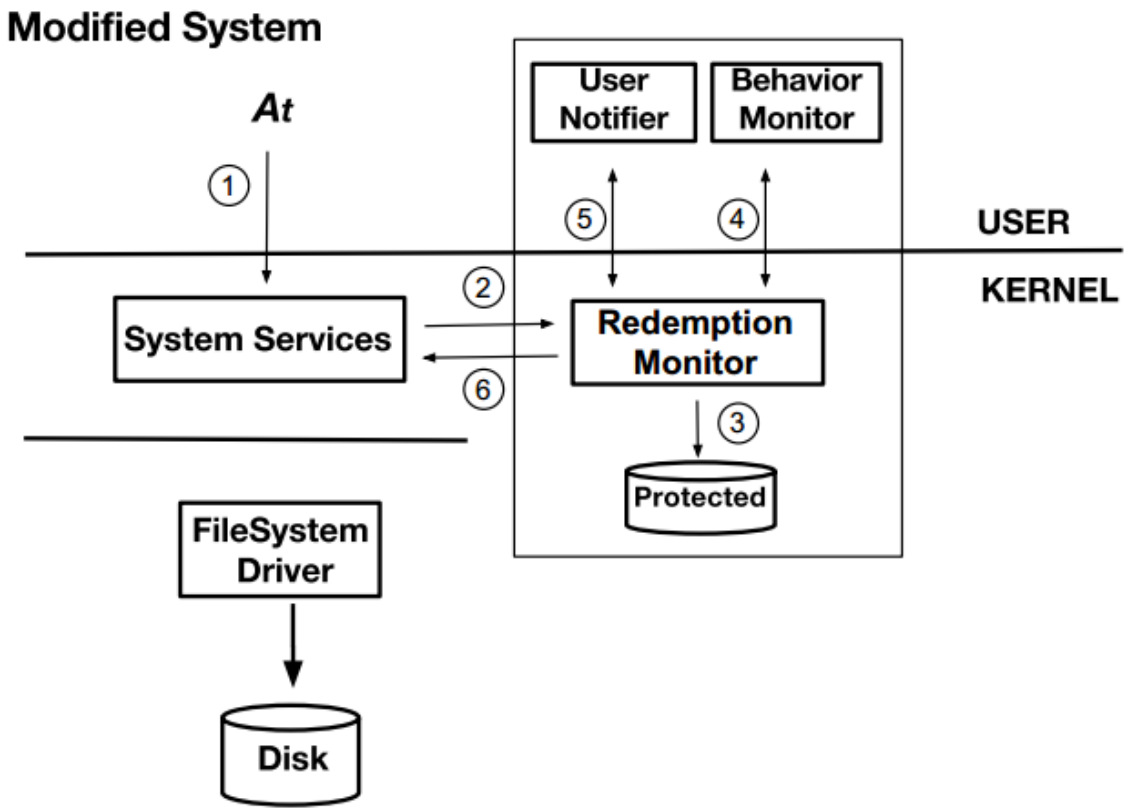
  the request would succeed if the corresponding file exists, and as long as the process holds the permission

- Redemption introduces some changes



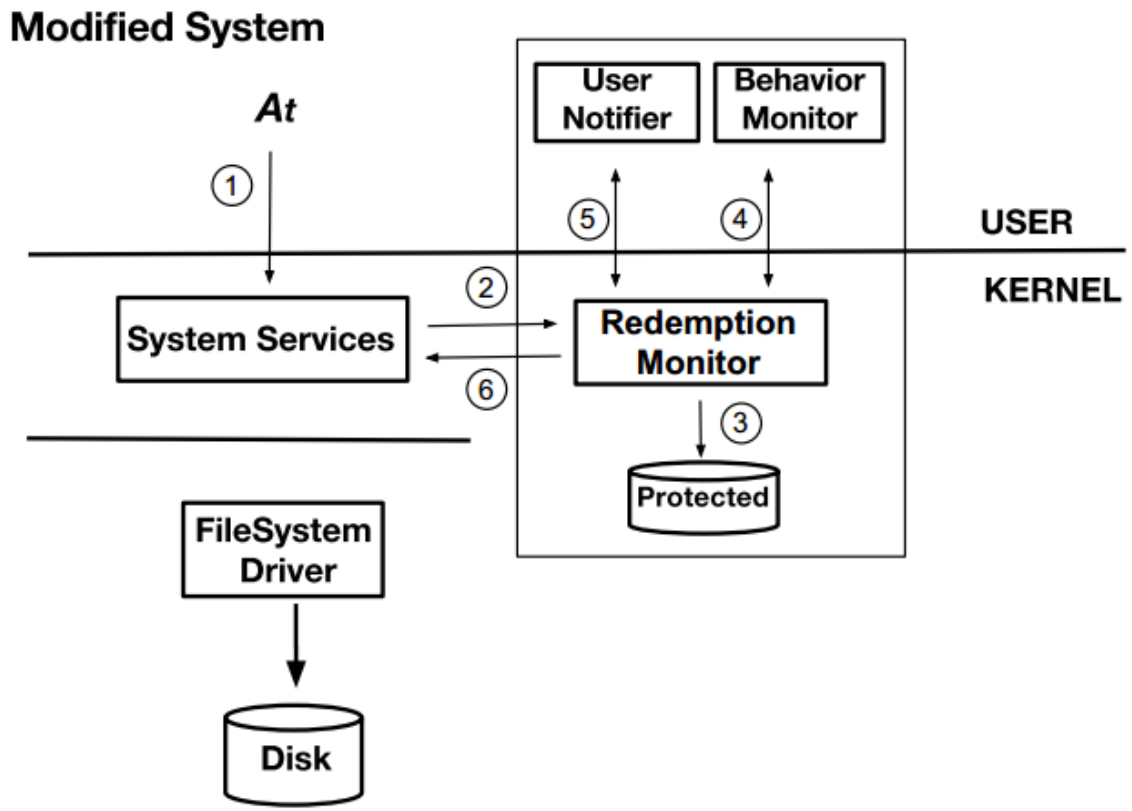
Design Overview

1. Redemption receives the request A from the application X to access the file F at the time *t*



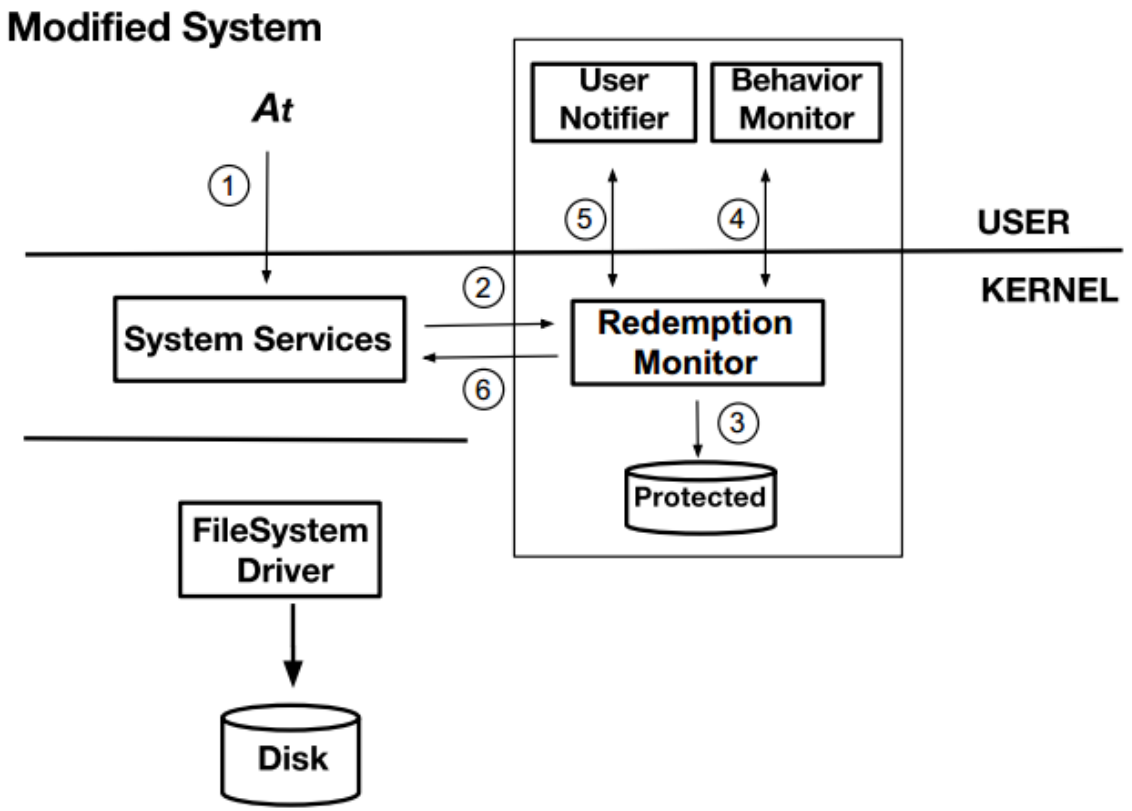
Design Overview

2. If the requests access with write or delete privilege to the file *F* , and the file *F* resides in a user defined path, the Redemption's monitor is called



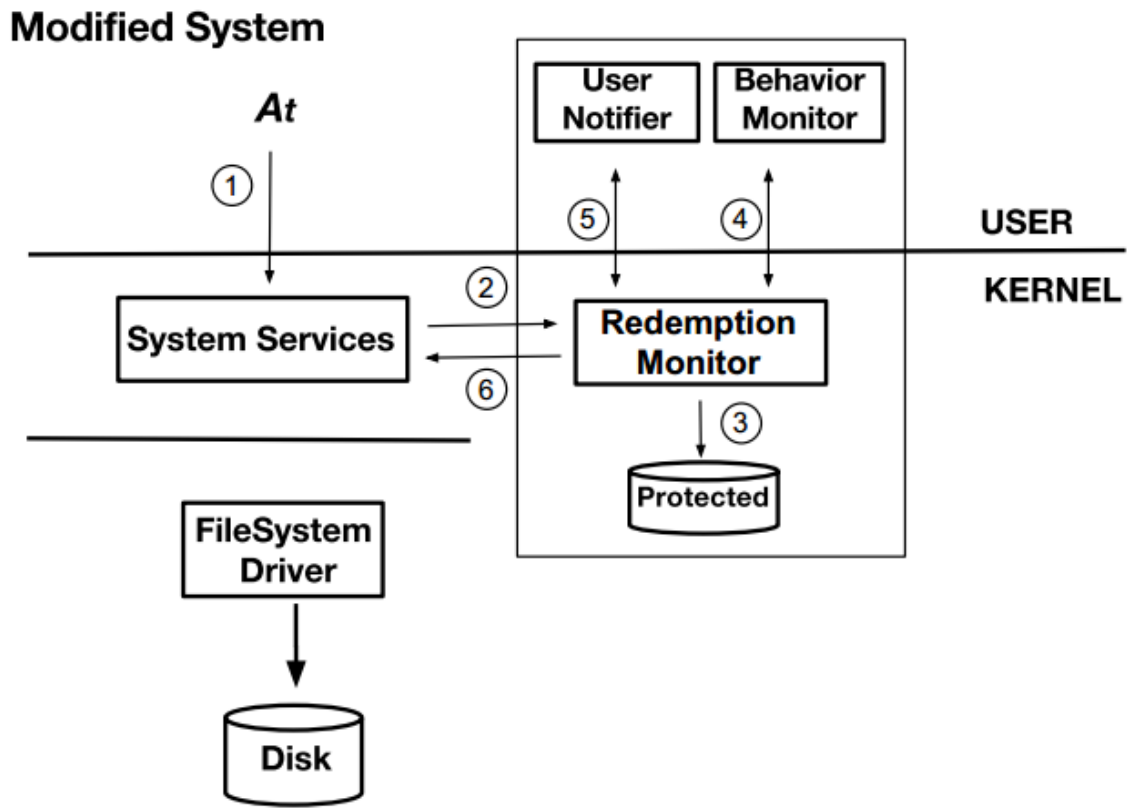
Design Overview

3. Redemption creates a corresponding file in the protected area, called *reflected* file, and handles the write requests. These changes are periodically flushed to the storage to ensure that they are physically available on the disk



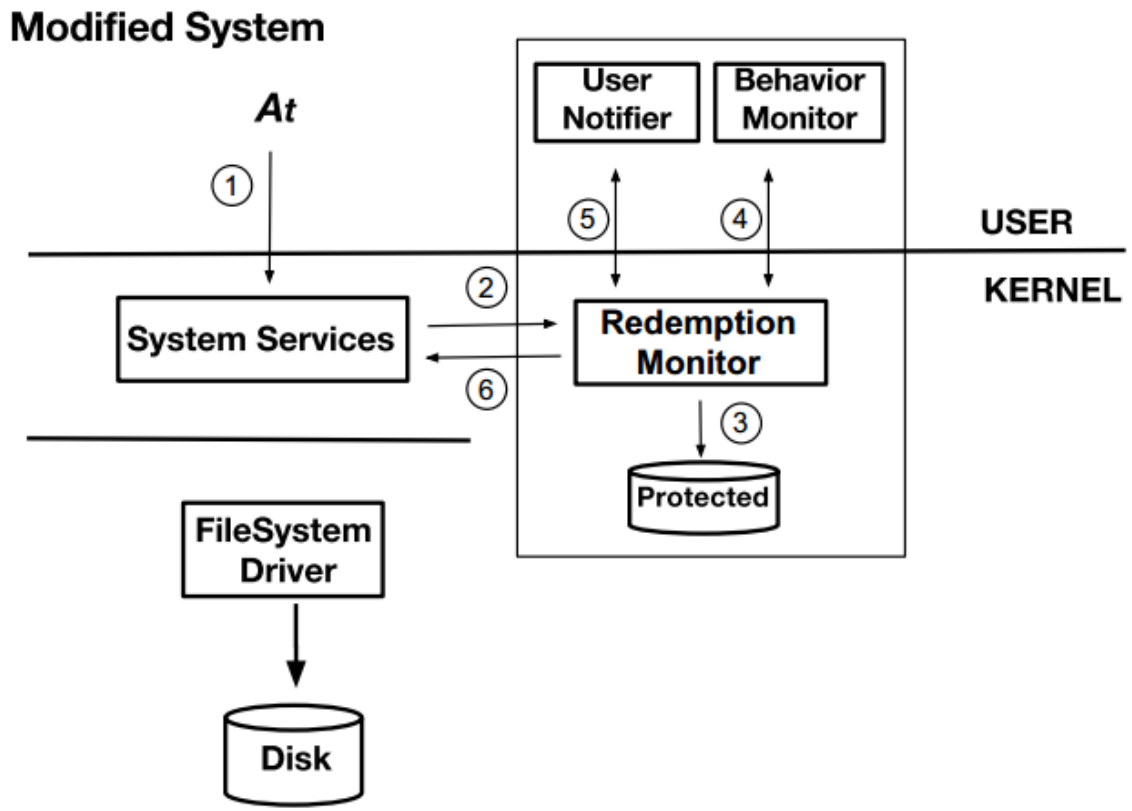
Design Overview

4. The malice score of the process is updated, and is compared to a pre-configured threshold α



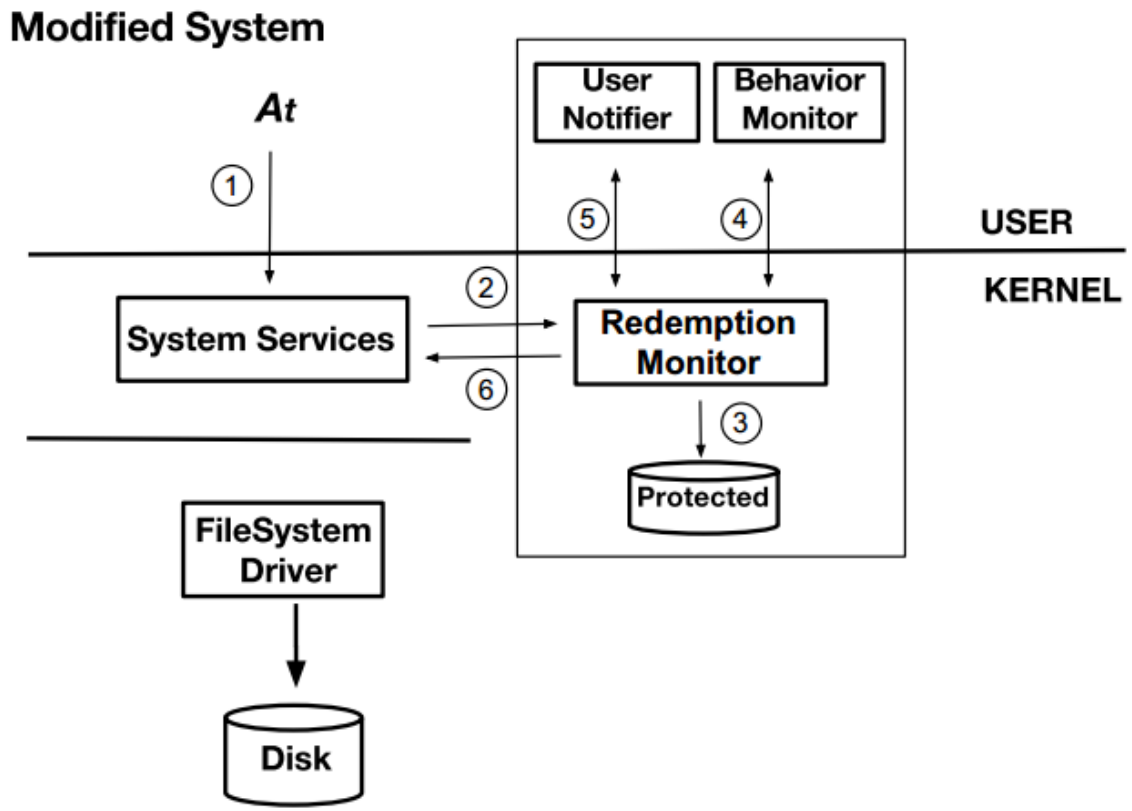
Design Overview

5. The Redemption monitor sends a notification to the display monitor to alert the user depending on the calculated malice score



Design Overview

6. A success/failure notification is generated, and is
sent to the system service manager



Detection Approach

Malice Score

- The malice score of a process represents the risk that the process exhibits ransomware behavior
- It determines whether the Redemption monitor should allow the process to access the file, or notify the user

Malice Score Calculation

Two features to be considered

1. Content-based features
 - i.e., changes in the content of each file

2. Behavior-based features
 - i.e., cross-file behavior of a process

Content-based Features

Entropy Ratio of Data Blocks

- For every read and write request to a file, Redemption computes the entropy of the corresponding data buffer.
- Comparing the entropy of read and write request serves an excellent indicator of ransomware behavior because of the popular strategy of reading in the original file data, encrypting it, and writing the encrypted version

Content-based Features

File Content Overwrite

- Malicious process overwrites the content of the user files with random data
- The system increases the malice score of a process if the process requests write access to different parts of a file
- A process is assigned a higher malice score if it overwrites all the content of the files

Content-based Features

Delete Operation

- Generally ransomware generate an encrypted version of the file, and delete the original file
- If a process requests to delete a file that belongs to the enduser, it receives a higher malice score

Behavior-based Features

Directory Traversal

- During an attack, the malicious process often arbitrarily lists user files, and starts encrypting the files with an encryption key
- A process receives a higher malice score if it is iterating over files in a given directory

Behavior-based Features

Converting to a Specific File Type

- A process receives a higher malice score if it converts files of differing types and extensions to a single known or unknown file type

Behavior-based Features

Access Frequency

- If a process frequently generates write requests to user files, the process would be given a higher malice score

Malice Score Calculation

- Recursive Feature Elimination (RFE) approach to determine the significance of each feature
- In each step, a feature with the minimum weight was removed
- The FP and TP rates were calculated by performing 10 fold cross-validation to quantify the contribution of each feature
- The assigned weights are then used in the formula

$$MSC(r) = \frac{\sum_{i=1}^{k} w_i \times r_i}{\sum_{i=1}^{k} w_i}$$

Implementation

- The authors implemented the system for the Windows environment as Windows OS is the main target of current ransomware attacks

Outlines

- Introduction
- Existing works
- Contribution
- Threat Model
- Design Overview
- Evaluation
- Limitations

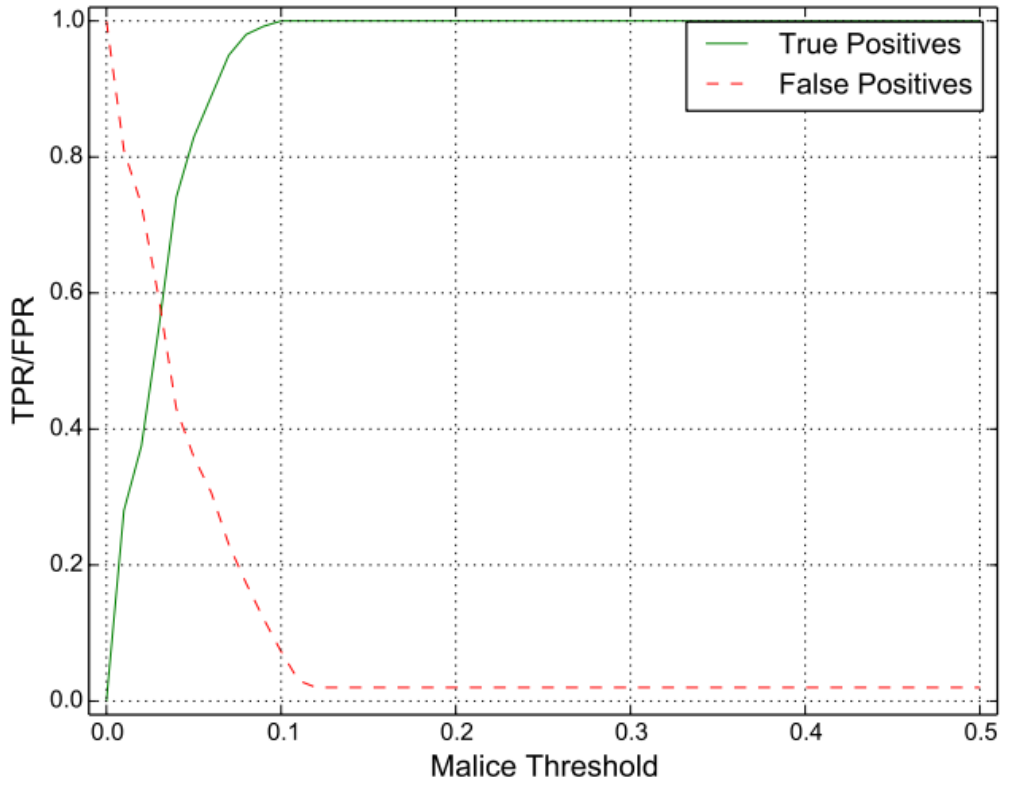
Evaluation

Data Collection

- Collect 9432 ransomware samples from public repository
- Collect benign applications from normal activities on Windows 7 machine

Results

The threshold value α = 0.12 gives the best detection and false positive rates (FP = 0.5%)



Results

Family	Redemption Samples/FA	CryptoDrop [31] Samples/FA	ShieldFS [15] Samples	PayBreak [23] Samples
Almalocker	-	-	-	1
Androm	-	-	-	2
Cerber	30/6	-	-	1
Chimera	-	-	-	1
CoinVault	19/5	-	-	-
Critroni	16/6	-	17	-
Crowti	22/8	-	-	-
CryptoDefense	42/7	18/6.5	6	-
CryptoLocker(copycat)	-	2/20	-	-
Cryptolocker	29/4	31/10	20	33
CryptoFortess	12/7	2/14	-	2
CryptoWall	29/5	8/10	8	7
CrypWall	-	-	-	4
CrypVault	26/3	-	-	-
CryptXXX	45/3	-	-	-
CryptMIC	7/3	-	-	-
CTB-Locker	33/6	122/29	-	-
DirtyDecrypt	8/3	-	3	-
DXXD	-	-	-	2
Filecoder	34/5	72/10	-	-
GpCode	45/3	13/22	-	2
HDDCryptor	13/5	-	-	-
Jigsaw	12/4	-	-	-
Locky	21/2	-	154	7
MarsJokes	-	-	-	1
MBL Advisory	12/4	1/9	-	-
Petya	32/5	-	-	-
PayCrypt	-	-	3	-
PokemonGo	-	-	-	1
PoshCoder	17/4	1/10	-	-
TeslaCrypt	39/6	149/10	73	4
Thor Locky	-	-	-	1
TorrentLocker	21/6	1/3	12	-
Tox	15/7	-	-	9
Troldesh	-	-	-	5
Virlock	29/7	20/8	-	4
Razy	-	-	-	3
SamSam	-	-	-	4
SilentCrypt	43/8	-	-	-
Xorist	14/7	51/3	-	-
Ransom-FUE	-	1/19	-	-
WannaCry	7/5	-	-	-
ZeroLocker	5/8	-	1	-
Total Samples (Families)	**677(29)**	**492(15)**	**305(11)**	**107(20)**
File Attacked/Recovered(FA/FR) Median	**5/5**	**10/0**	**-**	**-**

System Overhead

- The overhead of protecting a system from ransomware was under 6% in every test case
- On average, running applications took only 2.6% longer time to complete their tasks

Outlines

- Introduction
- Existing works
- Contribution
- Threat Model
- Design Overview
- Evaluation
- Limitations

Limitations

Attacking Redemption's Monitor
- Using social engineering techniques to frustrate users by creating fake alert message

Attacking the Malice Score Calculation Function
- keeping malice score lower than threshold by
  - selective content overwrite
  - using low entropy payload for content overwrite
  - launching periodic file destruction

Thank you