On Deception-Based Protection Against Cryptographic Ransomware

Ziya Alper Genç, Gabriele Lenzini, and Daniele Sgandurra

June 20, 2019
Ransomware Threat

Oops, your files have been encrypted!

What Happened to My Computer?
Your important files are encrypted.
Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?
Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don’t pay in 7 days, you won’t be able to recover your files forever.
We will have free events for users who are so poor that they couldn’t pay in 6 months.

How Do I Pay?
Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am GMT from Monday to Friday.

Send $300 worth of bitcoin to this address:
115p7UMMngoj1pMvkpHiljcRdfjNXj6LrLn

About bitcoin
How to buy bitcoins?
Ransomware Threat

The Untold Story of NotPetya,

**The Cost of NotPetya**

In 2017, the malware NotPetya spread from the servers of an unsuspecting Ukrainian software firm to some of the largest businesses worldwide, paralyzing their operations. Here's a list of the approximate damages reported by some of the worst victims:

- **$188,000,000**
  Snack company Mondelez (parent company of Nabisco and Cadbury)

- **$129,000,000**
  British manufacturer Reckitt Benckiser (owner of Lysol and Durex condoms)

- **$10 BILLION**
  Total damages from NotPetya, as estimated by the White House

By noon, ISSP’s founder, entrepreneur Derezianko, had spent the vacation too. Derezianko, had spent the day driving north to the early afternoon, he was warning ever
In the context of ransomware, 
**Deception = Decoy**

Decoys are 
- fictitious files 
- placed among user files 
- supposed to be not written.

**Any** write event on decoy files indicates ransomware activity.
Overview of Decoy File-Based Defense

Advantages:
- Accurate Detection
- Real-Time Protection
- Low Overhead

Requirements:
- Security: Mimicking the user.
- Usability: Not interfering the user.
Overview of Decoy File-Based Defense

Advantages:

Accurate Detection
Real-Time Protection
Low Overhead

Requirements:

Security: Mimicking the user.
Usability: Not interfering the user.
Overview of Decoy File-Based Defense

Advantages:
- Accurate Detection
- Real-Time Protection
- Low Overhead

Requirements:
- Security: Mimicking the user.
- Usability: **Not** interfering the user.
Overview of Decoy File-Based Defense

Advantages:
- Accurate Detection
- Real-Time Protection
- Low Overhead

Requirements:
- Security: Mimicking the user.
- Usability: **Not** interfering the user.
Overview of Decoy File-Based Defense

Advantages:
- Accurate Detection
- Real-Time Protection
- Low Overhead

Requirements:
- Security: Mimicking the user.
- Usability: **Not** interfering the user.

Question: How can we assess the efficiency of a decoy-file strategy?
Deception-Based Anti-Ransomware Systems

**CRYPTOSTOPPER**
Triggered on Write

**RWGUARD**
Behavioral Analysis
API Hooking
Decoy Files

**R-LOCKER**
Triggered on Read
Deception-Based Anti-Ransomware Systems

**CRYPTOSTOPPER**
triggered on write

**RWGUARD**
behavioral analysis
api hooking
decoy files

**R-LOCKER**
triggered on read
Deception-Based Anti-Ransomware Systems

**CRYPTOSTOPPER**
- Triggered on Write

**RWGUARD**
- Behavioral Analysis
- API Hooking
- Decoy Files

**R-LOCKER**
- Triggered on Read
Algorithm 1 \texttt{ANTISTATIC}: Collect files that are not hidden or filled with zero value.

1: function \texttt{COLLECT}(\textit{path}) \Comment*[h]{Directory of files to scan.}
2: \hspace{1em} \textit{FileList} ← \texttt{EnumerateFiles}(\textit{path})
3: \hspace{1em} \textit{GenuineList} ← \emptyset
4: \hspace{1em} for all \textit{f} ∈ \textit{FileList} do
5: \hspace{2em} if \textit{IsHidden}(\textit{f}) then
6: \hspace{3em} \textit{allNull} ← \texttt{True}
7: \hspace{2em} \hspace{1em} \textbf{while} not \texttt{EOF} do
8: \hspace{3em} \hspace{2em} \textit{b} ← \textit{f}.\texttt{.ReadByte}()
9: \hspace{3em} \hspace{2em} if \textit{b} \neq 0 then
10: \hspace{3em} \hspace{3em} \textit{allNull} ← \texttt{False}
11: \hspace{3em} \hspace{3em} \textbf{break} \Comment*[h]{\textit{f} might not be decoy, try next file.}
12: \hspace{2em} \hspace{1em} if \textit{allNull} = \texttt{True} then
13: \hspace{3em} \hspace{2em} \textit{GenuineList} ← \textit{GenuineList} \cup \{\textit{f}\}
14: \hspace{2em} \hspace{1em} \hspace{1em} else
15: \hspace{3em} \hspace{2em} \textit{GenuineList} ← \textit{GenuineList} \cup \{\textit{f}\}
16: \hspace{1em} \textbf{return} \textit{GenuineList}
CRYPTOSTOPPER vs ANTISTATIC
Let be $A$ a ransomware, $D$ the set of files generated by a decoy strategy $g$, $F = D \cup \neg D$, $S \subseteq F$ a set of files, and $n$ a natural number.

$$\Pr[X_A^g(S) = n]$$

(1)

It is the probability that $A$ encrypts $n$ other files before encrypting one in $S$.

For example, $\Pr[X_A^g(D) = 0] = 1$ indicates a good decoy file strategy, i.e., $g$ fools $A$ immediately.

For CRYPTOSTOPPER, $\Pr[X_{Alg_1}^{CS}(\neg D) > 0] = 0$. 
Confoundedness

Let be $U$ a user and $D$ the set of decoy file generated according to a strategy $g$, $F = D \cup \neg D$, and $S \subseteq F$ a set of files.

$$\Pr[\mathcal{Y}^g_U(S) = 1]$$

(2)

It is the probability that $U$ accesses a file in $S$ within a working session.

For example, $\Pr[\mathcal{Y}^g_U(D) = 1] = 0$ means that $U$ never gets confused.
Table 1: Selected attributes for NTFS files.

| Attribute       | Attribute Type Name                      | Description                                                                                                                                 |
|-----------------|------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Standard        | $STANDARD_INFORMATION                    | File attributes such as read-only, archive, and so on; time stamps, including when the file was created or last modified.                   |
| Information     |                                          |                                                                                                                                             |
| File Name       | $FILE_NAME                               | The file’s name.                                                                                                                              |
| Data            | $DATA                                    | The contents of the file.                                                                                                                    |
| Attribute List  | $ATTRIBUTE_LIST                          | List of the attributes that make up the file and the file record number.                                                                     |
Rowe\(^1\) collected file attributes, including (i) file name; (ii) file size; (iii) file type; and (iv) last modification time.

For a file system \(i\), let \(\mu_{ik}\) and \(\sigma_{ik}\) denote the mean and standard error of metric \(k\). Difference between systems \(i\) and \(j\):

\[
S_{ij} = \left(\frac{1}{72}\right) \sum_{k=0}^{35} \left[ \frac{|\mu_{ik} - \mu_{jk}|}{\sqrt{\sigma_{ik}^2 + \sigma_{jk}^2}} + \frac{|\sigma_{ik} - \sigma_{jk}|}{2\sigma_k} \right]
\]

\(^1\)Rowe, “Measuring the Effectiveness of Honeypot Counter-Counterdeception”.
Algorithm 2 Monitor User.

1: function MONITOR
2: \[Exp \leftarrow \text{FindProcess}(\text{Explorer})\]
3: \[\text{InjectProcess}(Exp, \text{SpyModule})\]
4: \[\text{GenList} \leftarrow \emptyset\]
5: while true do
6: \[f \leftarrow \text{Listen}(\text{SpyModule})\]
7: \[\text{GenList} \leftarrow \text{GenList} \cup \{f\}\]
8: return GenList

Algorithm 3 Replace WriteFile.

1: function REPLACE
2: \[\text{PList} \leftarrow \text{EnumAllProcesses}()\]
3: for all \(p \in \text{PList}\) do
4: \[\text{InjectProcess}(p, \text{InterceptMod})\]
5: \[\text{wf} \leftarrow \text{GetFuncAddr}(\text{WriteFile})\]
6: if \(\text{wf} \neq \text{NULL}\) then
7: \[\text{Replace}(\text{wf, encFile})\]
8: return Success
Consider an adversary $A$, which can monitor the user activity.

Let $[F]_U$ be files that $U$ accesses and cares, i.e., would pay the ransom for $[F]_U$.

If $g$ is perfectly usable, then its confoundedness is null, i.e., $\Pr[Y^g_U(D) = 1] = 0$.

If $A$ observed $[F]_U$, then $A$ could simply choose among the files in $[F]_U$.

If $\Pr[Y^g_U(D) = 1] = p > 0$, which means that $[F]_U \cap D \neq \emptyset$,
Assume that $A$ picks a target file in $[F]_U$ at random.

$A$ has $\frac{|[F]_U \cap \neg D|}{|[F]_U|} \cdot p + (1 - p)$ chance to pick up a good file.

If $p$ is negligible, $A$ has still a good chance.

If $p$ is significant, then $U$ accesses decoy files which goes against usability.

What if $A$ employs a better strategy?
Decoy-based anti-ransomware is a promising defense strategy.

Experimental results show that more research is needed. Especially, on generating decoy files.

Non-interference of decoy files should be examined.

Research challenge: find the right balance between security & usability. Being effective while not confusing the user.
On Deception-Based Protection Against Cryptographic Ransomware

ziya.genc@uni.lu