SRA: Smart Recovery Advisor for Cyber Attacks

Ka-Ho Chow[+], Umesh Deshpande[*], Sangeetha Seshadri[*], Ling Liu[+]
[+] Georgia Institute of Technology, Atlanta, Georgia, USA
[*] IBM Research - Almaden, San Jose, California, USA

ABSTRACT

Continuous Data Protection (CDP) is becoming instrumental in recovering applications from crypto-ransomware attacks. It enables fine-grained recovery through journaling, allowing the applications (its volumes) to recover to any previous state. While zero data loss can be achieved during recovery with CDP, the timestamp of the desired restore point, i.e., the one just prior to the attack, needs to be provided to reconstruct the volume. Such information is often unavailable in practice, and system administrators can only adopt a trial-and-error strategy to narrow down the time range of desired restore points by making multiple time-consuming recovery attempts. The recovery systems offer little guidance in pointing to the restore points containing a valid application state and reducing data loss. To address this problem, we equip the CDP-based recovery with machine intelligence. This demonstration showcases Smart Recovery Advisor (SRA), which offers interpretable, data-driven, and feedback-aware restore point recommendations that reduce the number of recovery attempts while minimizing data loss.

CCS CONCEPTS

• **Information systems** → **Storage recovery strategies**.

ACM Reference Format:
Ka-Ho Chow, Umesh Deshpande, Sangeetha Seshadri, and Ling Liu. 2021. SRA: Smart Recovery Advisor for Cyber Attacks. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21), June 18–27, 2021, Virtual Event, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3448016.3452766

1 INTRODUCTION

Over the past few years, cyber threats have been on the rise. Ransomware attacks are being automated [5], and they are causing ever-increasing downtime in production systems [4]. Scheduled backups have traditionally offered data protection by snapshotting the system state periodically. However, depending on the backup frequency, they can incur unacceptable data loss. Figure 1(a) shows an example of a crypto-ransomware attack from 14:00 to 16:30 on a system protected by creating a snapshot every six hours. In this case, the recovery to the nearest Restore Point (RP) results in two hours of data loss.

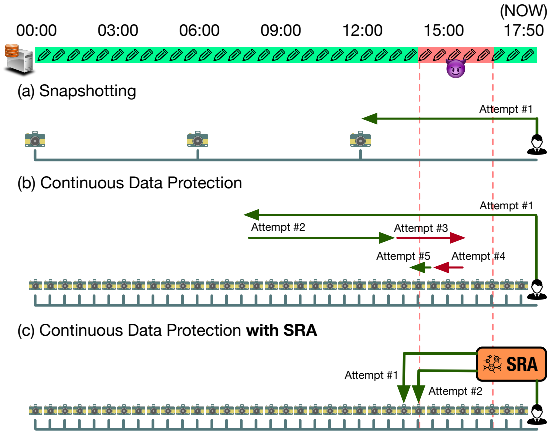


Figure 1: Recovery attempts at the recovery stage of (a) snapshotting technologies and continuous data protection (b) without and (c) with recommendations from SRA.

In the cybersecurity landscape, Continuous Data Protection (CDP) is regaining its popularity as a recovery mechanism. It offers near-zero data loss in recovery by journaling every modification to the filesystem or storage [9]. Therefore, in case of a cyber attack, the application state can be recovered just prior to the attack, attracting various database and disaster recovery solutions to employ CDP to achieve business continuity with near-zero data loss [13, 18]. Unfortunately, even though CDP is effective in minimizing data loss, the administrators need to spend significant recovery time to identify the desired version of data out of thousands of possible RPs. The increased recovery time could incur other damages such as the reputation of businesses or even life (e.g., a Californian hospital under ransomware attacks in 2016 took five days to restore and become fully operational [10]).

Existing research on CDP focuses on optimizing storage overhead [15] or volume reconstruction efficiency to speed up recovery time [16]. They assume the timestamp of the desired RP is known in advance, and thus the recovery can be accomplished in a single recovery attempt to the timestamp just before the attack. However, in practice, the desired restore point is not known in advance, and administrators need to guess it by trial-and-error. Taking the crypto-ransomware attack in Figure 1(b) as an example, the administrator first guesses the RP to be attempted (e.g., 08:00) and reconstructs the corresponding volume. A verification function, such as file hash comparison or virus scanning, is then conducted on the recovered volume to examine whether the valid (pre-attack) version is found. This reconstruction-verification process is repeated with different RP guesses to either find a valid version or further reduce data loss. Even though techniques speeding up volume reconstruction in a recovery attempt have been proposed, the verification function can still be time-consuming. Depending on the volume size and the verification type, an attempt may take hours to complete. Since the overall recovery time grows linearly with the number of recovery attempts, limited trials can be done in practice. Therefore, it

SIGMOD '21, June 18–27, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8343-1/21/06…$15.00
https://doi.org/10.1145/3448016.3452766

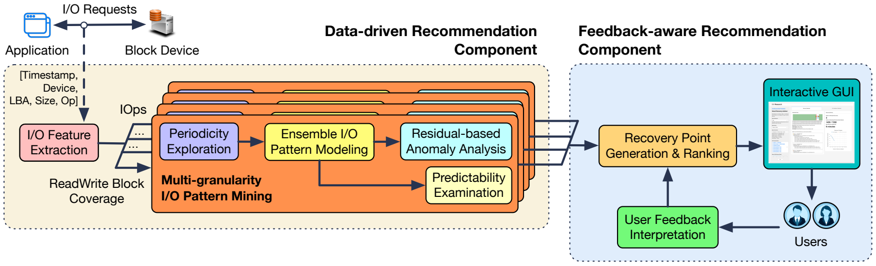

I/O Requests
Application
Block Device
Data-driven Recommendation Component
Feedback-aware Recommendation Component
[Timestamp, Device, LBA, Size, Op]
IOps
I/O Feature Extraction
ReadWrite Block Coverage
Periodicity Exploration
Ensemble I/O Pattern Modeling
Residual-based Anomaly Analysis
Predictability Examination
Multi-granularity I/O Pattern Mining
Recovery Point Generation & Ranking
Interactive GUI
User Feedback Interpretation
Users

Figure 2: System architecture of SRA with the data-driven component mining block-level I/O requests and the feedback-aware component enabling human-in-the-loop.

I/O Characteristics	Description
IOps	Number of I/O requests
Write Ratio	Ratio of writes to all I/O requests
Write Coverage	Coverage of blocks being written
ReadWrite Coverage	Coverage of blocks being overwritten
Write Age	Age of blocks being written

Table 1: Five example I/O characteristics (i.e., aggregation functions $\mathcal{F}$) employed by SRA.

is crucial to only conduct recovery attempts on high-quality RPs that minimize data loss. CDP would be more practical in recovering from crypto-ransomware attacks if the administrators are advised to pinpoint promising RPs to be attempted (Figure 1(c)), rather than guessing various RPs on the timeline without any basis.

Motivated by the above need, we propose the first machine learning-powered **S**mart **R**ecovery **A**dvisor, coined as SRA, towards usable CDP with the following contributions.

- SRA generates RP recommendations in a **data-driven** manner using a novel unsupervised machine learning algorithm to learn regular block-level I/O patterns and identify unusual activities to construct RP candidates.
- SRA offers **interpretable** recommendations where each RP is associated with a summary of the recommendation rationale, helping administrators to incorporate their domain knowledge in the decision-making process.
- SRA is **feedback-aware** and fine-tunes the recommendation strategy based on the feedback from administrators, allowing session-tailored optimization.

The existing crypto-ransomware detection systems can detect attacks by using upper-layer information, such as process IDs and file names [3, 7]. However, with evolving ransomware behaviors, it is common for the attacks to remain undetected for some time [1, 11, 12], and in spite of detection, the attack could be well underway, and the exact time of the attack may still be unknown. In such cases, SRA can assist administrators to quickly retrieve the version of data just prior to the point of the attack. Moreover, SRA offers a more general-purpose solution even for non-filesystem based attacks, e.g., database-level attack [17], which may not be detectable by filesystem-level analysis.

In this paper, we demonstrate the applicability of SRA in recovering from crypto-ransomware attacks. However, we believe that SRA can also be useful in detecting and recovering from other unusual activities, e.g., accidental deletion of a large number of files.

2 SRA SYSTEM OVERVIEW

SRA features interpretable RP recommendations through unsupervised machine learning with human-in-the-loop. Figure 2 illustrates the system architecture with two ingredients: (1) data-driven component and (2) feedback-aware component.

The crypto-ransomware behavior typically consists of reading victim files or databases, encrypting their content, and overwriting them (or creating a new copy) [8]. Such behavior results in an unusual I/O pattern at the block-level, which is rarely observed in uncompromised applications, e.g., untimely access, updates to an unusually large fraction of data, or updates to old or rarely modified blocks. SRA intercepts the block-level read/write requests between the application and its block storage device. Each request comprises five attributes: request timestamp, target device, logical block address (LBA), request size, and operation type (Op). The data-driven component performs little computation in real-time, while most of the work happens during recovery. In real-time, it only extracts salient features from the block-level I/O request stream and stores them for later analysis. Upon initiating a recovery session on the user interface, it models normal I/O behavior of the workload and compiles the suspicious events in a ranked list of RPs, where each RP is associated with a recommendation rationale to assist administrators in selecting the RP for the next recovery attempt. SRA also collects the user feedback for analysis to refine the strategy of generating the next batch of RP recommendations with recovery semantic inference and session-tailored optimization.

2.1 Data-driven Recommendation

SRA mines block-level I/O requests to recommend RPs. However, such raw input is not directly useful in detecting interesting I/O patterns and hence requires feature engineering for extracting the high-level aggregated statistics. Therefore, SRA first employs I/O feature extraction module to the block-level input stream and transforms it into a collection of carefully defined I/O characteristics. Each characteristic is constructed by consolidating the block-level I/O over an aggregation time window using different aggregation functions. Table 1 provides a few example I/O characteristics employed by SRA, and Figure 3 visualizes the "IOps" and "Write Ratio" with an aggregation window size of 15 minutes. Those time series of I/O characteristics describe the unique behavior of the workload and allow SRA to conduct time series modeling to detect unusual patterns and construct RP candidates.

Detecting I/O pattern anomalies is challenging, especially in workloads having complex and regularly repeated behaviors, as unusual activities may blend into normal ones and become difficult to identify. For instance, a crypto-ransomware attack was simulated between the time interval marked in red color in Figure 3. Since the regular behavior of the example workload contains periodic spikes, the I/O overhead incurred by the attack seems to resemble one of the normal spikes. Hence, it is difficult to spot even by human experts. To tackle such challenges, we design a novel I/O pattern mining algorithm employing unsupervised machine learning techniques.

We exploit the feature extraction module to generate a collection of I/O characteristics using aggregation functions $\mathcal{F}$ over multiple

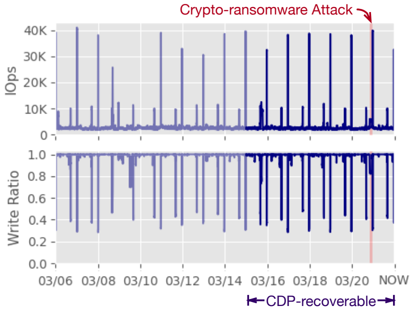


Figure 3: The challenging scenarios where malicious I/O patterns blended into regular I/O behaviors of the system.

aggregation windows $\mathcal{W}$. This design allows SRA to learn I/O characteristics from various temporal granularities and uncover anomalies spanning across both short and long durations. Then, we detect statistically anomalous activities on each of the $|\mathcal{F}| \times |\mathcal{W}|$ time series and combine them to finalize the overall decision. We introduce the concept of safe zones for SRA to securely learn normal patterns without human supervision. While CDP allows data recovery to any previous point in time, the oldest RP is often restricted due to limited storage (e.g., the administrator can only recover the system back to at most 7 days from now). We leverage the historical I/O patterns outside the CDP-recoverable window (i.e., the safe zone) to mine normal I/O behaviors, which are exploited to detect suspicious activities within the CDP-recoverable window (see Figure 3). Note that I/O characteristics can be compactly stored once they were extracted. Hence, even though historical block-level I/O requests would be aged out, their I/O characteristics can still be preserved at a low cost.

We design an ensemble learning-based algorithm to identify unusual activities on a given time series. In particular, we strategically divide the input time series into multiple sub-time series, conquer each sub-time series, and aggregate their decisions. This does not only allow SRA to be highly parallelizable but also more accurate, thanks to the error reduction property in ensemble learning [6]. Let $\boldsymbol{y}_{1:N} = \{y_1, y_2, ..., y_N\}$ be a sub-time series to be conquered and $\boldsymbol{y}_{1:s}$ be the segment inside the safe zone. SRA first leverages autocorrelation to examine the periodicity of the I/O characteristics, which is essential for the downstream analysis. In particular, we use the Auto-Correlation Function (ACF) at lag k:

$$\mathrm{ACF}(k) = \frac{\sum_{t=k+1}^{s}(y_t - \bar{\boldsymbol{y}}_{1:s})(y_{t-k} - \bar{\boldsymbol{y}}_{1:s})}{\sum_{t=1}^{s}(y_t - \bar{\boldsymbol{y}}_{1:s})^2}, \tag{1}$$

where $\bar{\boldsymbol{y}}_{1:s} = \frac{1}{s}\sum_{t=1}^{s} y_t$, to capture the correlation of a time series to a delayed copy of itself. By examining the local maxima across all k, one can obtain the periods of repeated patterns in $\boldsymbol{y}_{1:s}$.

Upon determining the periodicity of $\boldsymbol{y}_{1:s}$, SRA exploits a season-trend decomposition technique based on STL [2] to decompose $\boldsymbol{y}_{1:N}$ into three components:

$$\boldsymbol{y}_{1:N} = \mathrm{Season}(\boldsymbol{y}_{1:N}) + \mathrm{Trend}(\boldsymbol{y}_{1:N}) + \mathrm{Residual}(\boldsymbol{y}_{1:N}), \tag{2}$$

where the season and trend components constitute the normal behavior of $\boldsymbol{y}_{1:N}$ while the residual component captures the uncertainty of measurements following a normal distribution. We extract the statistics of residual from the safe zone (i.e., $\mathrm{Residual}(\boldsymbol{y}_{1:s})$), which are precious for SRA in two aspects. First, we can leverage the well-known 3-σ rule to detect anomalies within the CDP-recoverable window. Second, the statistics also serve as an indicator of how predictable is the I/O pattern. SRA possesses a weighting mechanism that automatically emphasizes highly predictable I/O characteristics as the violation of predictable patterns is much more alarming than unpredictable ones. Eventually, we consolidate results across different I/O characteristics, temporal granularity, and sub-time series to finalize the recommended RPs.

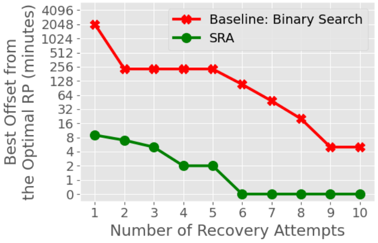


Figure 4: The best offset from the optimal RP following recommendations from the baseline and SRA. SRA achieves data loss of only 9 minutes in the 1st recovery attempt, but the baseline takes nine attempts to accomplish the same quality.

2.2 Feedback-aware Recommendation

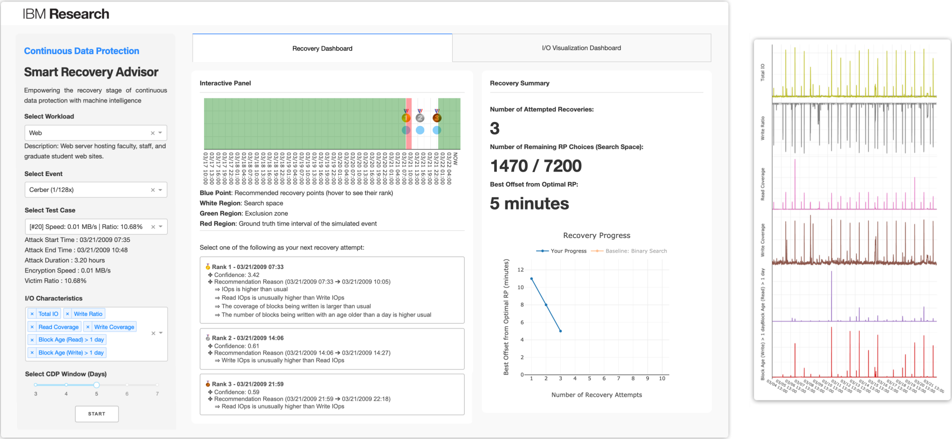
Given that multiple uncommon activities can happen (e.g., a scheduled system update followed by a crypto-ransomware attack after six hours), incorporating user feedback allows SRA to adapt to domain knowledge and recovery goal, and accordingly re-prioritize the RP recommendations. To assist the administrators in providing feedback, each RP is associated with a summary of rationale (e.g., an unusually high IOps is detected). We categorize feedback into *explicit* and *implicit*, and design dedicated modules to interpret feedback based on each recovery session.

Explicit feedback is obtained directly from the administrator interacting with the SRA user interface, e.g., the satisfaction of the recovery attempt, which can be answered by running verification functions on the recovered volume to determine whether the recovery goal is met. It allows SRA to infer the new temporal search space, refine the multi-granularity I/O pattern mining algorithm (e.g., the safe zone for statistics extraction), and regenerate RP candidates for the next batch of recommendations.

Implicit feedback is collected indirectly by monitoring the behavior of the administrator throughout the recovery session. Every recovery attempt begins with selecting an RP from the ranked list generated by SRA. We build a session-specific recovery profile by learning patterns of selected RPs (e.g., always avoiding RPs with low IOps implies the event to be resolved tends to generate high IOps). Then, SRA uses this profile to reorder the RP candidates. Additionally, SRA keeps track of blocks accessed by the administrator during the verification process. Such information is valuable in pinpointing high-quality RPs, e.g., by learning about the part of data that is interesting to the administrator.

2.3 Evaluation

To delineate the advantages at the recovery stage of CDP using SRA, Figure 4 compares it with a baseline strategy using binary search, a common option adopted by the administrators, iteratively



IBM Research

Continuous Data Protection

Smart Recovery Advisor

Empowering the recovery stage of continuous data protection with machine intelligence

Select Workload: Web

Description: Web server hosting faculty, staff, and graduate student web sites.

Select Event: Cerber (1/128x)

Select Test Case: [#20] Speed: 0.01 MB/s | Ratio: 10.68%

Attack Start Time : 03/21/2009 07:35

Attack End Time : 03/21/2009 10:48

Attack Duration : 3.20 hours

Encryption Speed : 0.01 MB/s

Victim Ratio : 10.68%

I/O Characteristics: × Total IO × Write Ratio × Read Coverage × Write Coverage × Block Age (Read) > 1 day × Block Age (Write) > 1 day

Select CDP Window (Days): 5

Interactive Panel

Blue Point: Recommended recovery points (hover to see their rank)

White Region: Search space

Green Region: Exclusion zone

Red Region: Ground truth time interval of the simulated event

Select one of the following as your next recovery attempt:

Rank 1 - 03/21/2009 07:33
- Confidence: 3.42
- Recommendation Reason (03/21/2009 07:33 → 03/21/2009 10:05)
  - ⇒ IOps is higher than usual
  - ⇒ Read IOps is unusually higher than Write IOps
  - ⇒ The coverage of blocks being written is larger than usual
  - ⇒ The number of blocks being written with an age older than a day is higher usual

Rank 2 - 03/21/2009 14:06
- Confidence: 0.61
- Recommendation Reason (03/21/2009 14:06 → 03/21/2009 14:27)
  - ⇒ Write IOps is unusually higher than Read IOps

Rank 3 - 03/21/2009 21:59
- Confidence: 0.59
- Recommendation Reason (03/21/2009 21:59 → 03/21/2009 22:18)
  - ⇒ Read IOps is unusually higher than Write IOps

Recovery Summary

Number of Attempted Recoveries: 3

Number of Remaining RP Choices (Search Space): 1470 / 7200

Best Offset from Optimal RP: 5 minutes

Figure 5: Screenshots of SRA's user interface.

reducing the temporal search space by half. We create two recovery sessions to recover the data of the same application undergone a simulated crypto-ransomware attack. Each of them follows the recommendations from their respective algorithm and attempts recovery ten times. We measure the best offset from the optimal RP reached at each recovery attempt. After the 1st attempt, the recovery session using SRA can already reach an RP 9 minutes before the start time of the attack, which was kept secret from SRA. The same recovery quality in the session using binary search can not be reached until the 9th attempt. Interestingly, SRA does not stop progressing and continues to reach better RPs when the administrator keeps following the refined recommendations and eventually achieves the optimal RP at the 6th trial.

3 DEMONSTRATION OVERVIEW

Our demonstrations leverage SRA to resolve crypto-ransomware attacks. We provide test cases overlaying malicious I/O requests simulating three infamous crypto-ransomware families (i.e., Cerber, CBT-Locker, and CryptVault) on real-world block-level traces [14]: a home directory, a web server, and an online learning platform. The victim size and encryption speed of the attack are sampled from distributions collected from real crypto-ransomware samples [3]. To mimic advanced adversaries who attempt to bypass defense mechanisms by blending malicious I/O into regular patterns of the workload, we provide additional test cases for Cerber with various slow-down factors (e.g., 1/128×). By repeating each configuration with randomly selected attack start time, over two thousand test cases spanning across various workloads, attacks, and aggressiveness are available for demonstration participants to explore SRA.

Once participants configure the simulation and initiate the recovery session by clicking the "START" button in the SRA user interface (Figure 5), the set of recommended RPs will be offered in the recovery dashboard presented as both a timeline (top) and a ranked list (bottom) where the detailed rationales are also available. For example, one reason the top-ranked RP is recommended is the number of read requests was unusually higher than writes within the specified time period (i.e., 07:33 to 10:05). Upon selecting the RP, the demonstration platform simulates the recovery attempt and updates the summary, including the remaining search space, the current best offset from the optimal RP, and a comparison between SRA and binary search recovery progress. Note that such evaluation is unavailable in practical deployments where the ground truth attack time is inaccessible. Participants are also allowed to visualize the I/O patterns extracted by SRA for data-driven recommendations through the visualization dashboard.

The demonstration begins with understanding the importance of SRA's machine learning approach to detect unusual activities for data-driven recommendations. We compare the I/O patterns extracted from different simulation settings to show that identifying malicious I/O blended into normal behaviors of the workload is challenging even for a human expert. It further strengthens the need for a sophisticated technique to mine the I/O patterns of the system and detect activities violating typical behaviors.

The next set of demonstrations include recovery sessions resolving crypto-ransomware attacks with minimal data loss and recovery attempts. We demonstrate two extreme cases. We first consider the scenario where the administrator always trusts SRA and selects the top-ranked RP to be the next recovery attempt. SRA can offer high-quality recommendations and lead to markedly less data loss than the binary search solution using the same number of recovery attempts. Next, we demonstrate the case where the administrator always selects the least-likely RP returned by SRA. The goal is to reveal SRA's ability to progressively correct users' suboptimal decisions, which is inevitable in practice even if the recovery is operated by an experienced administrator.

In summary, the demonstration provides participants opportunities to experience the intelligent recovery offered by SRA. With thousands of test cases covering a wide range of scenarios, SRA unleashes the full potential of CDP by offering interpretable, data-driven, and feedback-aware RP recommendations.

ACKNOWLEDGMENTS

The first author thanks the opportunity of the 9-week internship at IBM Research - Almaden in summer 2020, especially thanks Dr. Wil Plouffe, Manager of the Storage Systems Research group, and Larry Chiu, Head of Global Storage Research, for their support and valuable feedback. The authors from the Georgia Institute of Technology are partially supported by the National Science Foundation under Grants NSF 1564097, NSF 2026945, and NSF 2038029.

REFERENCES

[1] Bander Ali Saleh Al-rimy, Mohd Aizaini Maarof, and Syed Zainudeen Mohd Shaid. 2018. Ransomware Threat Success Factors, Taxonomy, and Countermeasures: A Survey and Research Directions. *Computers & Security* 74 (2018), 144–166.

[2] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. 1990. STL: A Seasonal-trend Decomposition. *Journal of Official Statistics* 6, 1 (1990), 3–73.

[3] Andrea Continella, Alessandro Guagnelli, Giovanni Zingaro, Giulio De Pasquale, Alessandro Barenghi, Stefano Zanero, and Federico Maggi. 2016. ShieldFS: A Self-healing, Ransomware-aware Filesystem. In *Annual Conference on Computer Security Applications (ACSAC)*.

[4] Coveware. 2020. Q4 Ransomware Marketplace Report. https://www.coveware.com/blog/2020/1/22/ransomware-costs-double-in-q4-as-ryuk-sodinokibi-proliferate.

[5] EY. 2019. Surviving Extreme Digital Disruption. https://www.ey.com/en_us/consulting/how-to-better-prepare-for-cyberattacks.

[6] Jiawei Han, Jian Pei, and Micheline Kamber. 2011. *Data Mining: Concepts and Techniques*. Elsevier.

[7] Amin Kharaz, Sajjad Arshad, Collin Mulliner, William Robertson, and Engin Kirda. 2016. UNVEIL: A Large-scale, Automated Approach to Detecting Ransomware. In *USENIX Security Symposium*.

[8] Amin Kharraz, William Robertson, Davide Balzarotti, Leyla Bilge, and Engin Kirda. 2015. Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*.

[9] Guy Laden, Paula Ta-Shma, Eitan Yaffe, Michael Factor, and Shachar Fienblit. 2007. Architectures for Controller Based CDP. In *USENIX Conference on File and Storage Technologies (FAST)*.

[10] Steve Mansfield-Devine. 2016. Ransomware: Taking Businesses Hostage. *Network Security* 10 (2016), 8–17.

[11] NewsCred. 2018. 11 Ways Ransomware is Evolving. https://insights.samsung.com/2018/04/11/11-ways-ransomware-is-evolving/.

[12] Nyotron. 2019. RIPlace Evasion Technique. https://www.zerto.com/protect/business-running-with-no-downtime/.

[13] Oracle. 2020. Database Cyber-Attack Protection with Zero Data Loss Recovery Appliance. https://blogs.oracle.com/maa/db-cyber-attack-protection-with-zdlra.

[14] Akshat Verma, Ricardo Koller, Luis Useche, and Raju Rangaswami. 2010. SRCMap: Energy Proportional Storage Using Dynamic Consolidation. In *USENIX Conference on File and Storage Technologies (FAST)*.

[15] Weijun Xiao, Jin Ren, and Qing Yang. 2008. A Case for Continuous Data Protection at Block Level in Disk Array Storages. *IEEE Transactions on Parallel and Distributed Systems* 20, 6 (2008), 898–911.

[16] Jing Yang, Qiang Cao, Xu Li, Changsheng Xie, and Qing Yang. 2011. ST-CDP: Snapshots in TRAP for Continuous Data Protection. *IEEE Trans. Comput.* 61, 6 (2011), 753–766.

[17] ZDNet. 2020. Hacker Ransoms 23k MongoDB Databases and Threatens to Contact GDPR Authorities. https://www.zdnet.com/article/hacker-ransoms-23k-mongodb-databases-and-threatens-to-contact-gdpr-authorities.

[18] Zerto. [n.d.]. Mitigate Downtime and Data Loss. https://www.zerto.com/protect/business-running-with-no-downtime/.