仮想化ソフトウェアのセキュア化に向けた脆弱性の調査分析

葛野 弘樹[1)]　**深井 貴明**[2)]
Hiroki Kuzuno　Takaaki Fukai

概要

システムソフトウェアにおいて脆弱性を利用した攻撃が指摘されている．脆弱性を利用した攻撃により，開発者の意図しないソフトウェア動作により，許可されてないメモリ領域への書込みや動作停止が引き起こされる．仮想化ソフトウェアでは，セキュリティを高めた設計を基にした開発も進められている．しかし，実装においては，依然として脆弱性が含まれる可能性がある．本稿では，仮想化ソフトウェアの脆弱性情報の収集と分類を通じ，脆弱性種別ならびに仮想化ソフトウェアにおける修正状況の有無を明らかにし，脆弱性の傾向を把握するための調査分析を進めた．調査分析結果では，Linux KVM において，115 個の脆弱性に対して， 28 種類の脆弱性種別，ならびに 42 個の脆弱性に関する修正結果を把握可能なことを確認した．

1　はじめに

システムソフトウェアにおいて，攻撃に利用可能な脆弱性は意図しないソフトウェア動作を引き起し，メモリの不正改ざんや動作停止に繋がる可能性がある．そのため，攻撃に利用可能な脆弱性への対応としてソースコードの修正と更新は必要になる．最新の仮想化ソフトウェアにおいては，セキュリティを高めた設計に基づいて開発が行われることもあり，今後の進展が期待されている [1]．一方，十分なセキュリティを確保するためには設計を反映した実装を行う必要があり，過去の脆弱性をふまえた実装が求められるが，依然として脆弱性が含まれる可能性が指摘されている [2]．そのため，仮想化ソフトウェアの安全性確保のために次の目的をあげる．

研究の目的：脆弱性分析と修正状況の整理による利活用
仮想化ソフトウェアにて報告された脆弱性種別を整理し，脆弱性の傾向を把握する．仮想化ソフトウェアにおける過去の脆弱性修正状況からセキュリティを高めた設計に基づいた実装を実現するため，今後，仮想化ソフトウェアのセキュリティ向上のための利活用に繋げることを目的とする．

本研究では，仮想化ソフトウェアの脆弱性情報の収集と分類を通じ，脆弱性種別ならびに仮想化ソフトウェアにおける修正状況を集約し，調査分析を行う．調査分析対象として，カーネルの仮想化機能に着目し，仮想化に関する脆弱性一覧と傾向，ならびに脆弱性に対応するソースコードの修正状況に関して傾向の把握を進めた．調査分析の事例として，脆弱性情報は National Vulnerability Database（NVD）より収集分析し， Linux の仮想化機能である KVM のソースコードでの修正状況の把握を行った．

本稿での研究貢献は以下の通りである：

1. 脆弱性情報の取得と分析を通じ，仮想化ソフトウェアにおける脆弱性の傾向分析と修正状況を調査分析を進めた．
2. 調査分析の事例として，Linux の仮想化機能 KVM に着目し，脆弱性種別の集約と修正有無を把握可能か確認し，今後の設計や実装での調査分析結果の利活用を検討した．

1）　神戸大学 大学院工学研究科
2）　国立研究開発法人 産業技術総合研究所

表 1　Vulnerabilities Information

Item	Description
CVE	A list of common identifiers for publicly known cybersecurity vulnerabilities
CPE	A structured naming scheme for systems, software, and packages

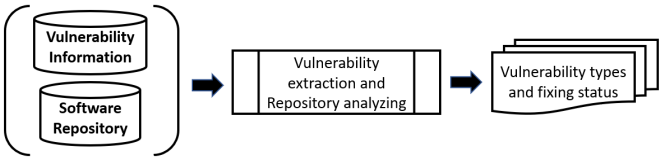

図 1　調査手順の概要

2　脆弱性情報と調査対象

2.1　脆弱性情報

脆弱性は攻撃に利用可能なソフトウェア動作に影響を与える実装不備であり， NVD にて脆弱性情報として提供される．本稿では，脆弱性情報のうち，共通脆弱性識別情報として，脆弱性毎に一意の番号が割当てられる CVE，ならびに共通プラットフォーム一覧として，脆弱性の報告されたハードウェアやソフトウェアを示すために用いられる Common Platform Enumeration（CPE）を用いる（表 1）．

2.2　調査対象

本研究における調査対象ソフトウェアでは，ソースコードは公開されており，ソフトウェアの特定機能に対する脆弱性には一意に識別可能な情報として登録されているとする．また，脆弱性が修正されていた場合，修正状況はソフトウェアを管理するレポジトリのコミットログに記載されていることとする．

3　調査方法と実装

3.1　方針

本研究における仮想化ソフトウェアの脆弱性分析では，次の方針とした．

方針：報告された脆弱性について，脆弱性種別の分析と対象とする仮想化ソフトウェアにおいて修正がなされているか確認可能とする．

3.2　目的

本研究における調査の目的を次のように定めた．

調査の目的：最新の脆弱性にも対応可能とし，かつ対象とする仮想化ソフトウェア更新に追随するため，定期的な脆弱性種別と修正有無の把握を可能とし，分析処理を行う．

3.3　手順

調査手順の概要を図 1 に示す．要件を満たすため，脆弱性データベースから脆弱性情報の一覧，分析対象とする仮想化ソフトウェアの更新状況を保持する．全ての脆

表 2　CVE status of Linux KVM vulnerability

Item	Number of vulnerabilities
Total vulnerability	115
Commit log existing	42
Commit log no existing	73

弱性情報，ならびに更新履歴から脆弱性の種別と修正状況の有無を分析する．

3.4　実装

実現方式において対象とする仮想化ソフトウェアはカーネルにおける仮想化機能を提供する Linux KVM とした．実現方式では，NVD の脆弱性データベースを用いた CVE 情報の取得と分析，ならびに Linux リポジトリにおける CVE 毎の修正状況の調査を行う．

CVE 情報の取得と分析では，CVE に含まれる CPE に記述された Linux に関するソフトウェア識別子を調査し，Linux を対象とする CVE リストを作成する．その後，CVE の詳細情報において KVM に言及している脆弱性を抽出し，KVM の CVE リストを作成する．

CVE 毎の修正状況の調査では，該当する KVM の CVE リストに含まれる CVE 毎に Linux KVM のリポジトリのコミット履歴を探索し，CVE 番号が含まれているか判別し，修正状況の確認を行う．

4　調査結果

4.1　調査項目と目的

特定のカーネル機能に関する脆弱性件数と種別，および脆弱性の分析可能性の把握を目的として調査した．調査項目と内容を以下に示す．

- 仮想化ソフトウェアに対する脆弱性の分析
  カーネルの仮想化機能を対象とし，脆弱性の選別ならびに分類，および修正有無を把握可能か調査分析した．

4.2　調査環境

調査に用いた計算機は Intel(R) Xeon(R) W-2295（3.00GHz，18 コア，メモリ 32GB），OS は Debian 11.3，Linux kernel 6.0 とした．調査対象とする仮想化機能は，Linux kernel の KVM とし，調査分析を行う Python コードは 209 行にて実現した．

4.3　仮想化ソフトウェアに対する脆弱性の分析

調査における Linux KVM の脆弱性の分析として，NVD に登録されている Linux KVM における脆弱性とレポジトリにおけるコミット有無について表 2 に示す．Linux KVM に関する脆弱性は 115 件あり，42 件はコミットログにて修正が記載されているが，73 件はコミットログには記載されていないことが確認された．また，28 種類の CWE が確認された．CWE 毎の脆弱性数を表 3 に示す．表 3 より，10 件以上の CVE が分類された CWE は NVD-CWE-noinfo，CWE-20，CWE-476，CWE-399 であることが確認された．

脆弱性データベースとレポジトリのコミット履歴を用いることで，対象とした Linux KVM の脆弱性報告数と種別，ならびに修正状況を把握可能である．実際の攻撃に利用されている脆弱性か更なる分析を行うことや，他の仮想化ソフトウェアにおける脆弱性傾向との比較に利用可能といえる．

5　考察

表 3　Top CWE of Linux KVM

Type	Content	CVE
NVD-CWE-noinfo	Insufficient Information	16
CWE-20	Improper Input Validation	13
CWE-476	NULL Pointer Dereference	11
CWE-399	Resource Management Errors	10
CWE-264	Permissions, Privileges, and Access Controls	7
CWE-119	Improper Restriction of Operations	6
CWE-362	Race Condition	6
CWE-416	Use After Free	5
CWE-189	Numeric Errors	4
CWE-200	Exposure of Sensitive Information	4
CWE-787	Out-of-bounds Write	4
17 CWEs	Under 3 CVEs	29
Total		115

5.1　調査結果に対する考察

仮想化機能に関する脆弱性の調査にて，脆弱性種別の分類ならびに修正状況の有無を把握可能なことを確認した．調査結果より，115 個の仮想化機能に関して，73 個の脆弱性修正はコミット履歴に含まれておらず，脆弱性情報の更なる分析が必要である．表 3 より，上位の脆弱性種別として，入力処理は 13 件，ポインタ参照は 11 件と，入出力や変数の取扱いに関する脆弱性が多いと推測される．一方，CWE の情報がない脆弱性が 16 件，資源管理関係が 10 件となっており，ソースコード単位での脆弱性箇所の分析が必要といえる．

5.2　調査方法の考察

仮想化ソフトウェアにおいて，セキュリティを考慮した設計に基づいた場合においても，依然として安全な実装は難しく，脆弱性を取り除くことは困難である．調査結果から，他の実装や異なるアーキテクチャでの実装に利活用することを検討しており，脆弱性を含むソースコードの修正前後の比較を行い，他の仮想化ソフトウェアにおける脆弱性調査に活用可能と考えている．

6　まとめと今後の予定

本稿では，仮想化ソフトウェアに関する脆弱性情報の収集と分類を通じ，脆弱性種別と修正状況の有無を明らかにするための調査を行った．カーネルの仮想化機能の脆弱性を調査し，仮想化に関する脆弱性一覧と傾向を分析し，脆弱性に対応するソースコードの修正状況の把握を可能とした．Linux KVM において，115 個の脆弱性を集約し，28 種類の脆弱性種別，ならびに 42 個の脆弱性では修正結果を把握可能なことを確認した．

今後，ソースコードに対する脆弱性調査や，動作中のソフトウェアに対して自動的に脆弱性を発見するファジング手法と組合せ，仮想化ソフトウェアにおける特定機能の開発における効率的なセキュリティ向上を進める予定である．

謝辞

本研究の一部は，JST さきがけ JPMJPR22PB の支援，ならびに 2022 年度国立情報学研究所公募型共同研究（22S0302）の助成を受けたものです．

参考文献

[1] MilvusVisor, available from `https://github.com/RIKEN-RCCS/MilvusVisor/`. (accessed 2022-11-12).

[2] Bae, Y., et al.: RUDRA: Finding Memory Safety Bugs in Rust at the Ecosystem Scale. *Proc. the ACM SIOPS 28th Symposium on Operating Systems Principles*, October, pp. 84–99, (2021).