APJ 7(1):76-91, Spring 2026

Arman Process Journal (APJ)

Arman Process Journal (APJ)

Homepage: https://www.armanprocessjournal.ir

Design, Implementation, and Evolution of Wiper Malware in Offensive Cyber Operations: From Logical Data Destruction to Irrecoverable Erasure of Adversary Information

N.Ataeian[1*], MM.Shirmohammadi[2]

[1] Department of Computer Engineering, Ha.C, Islamic Azad Unuversity, Hamedan, Iran
[2] Department of Computer Engineering, Ha.C, Islamic Azad Unuversity, Hamedan, Iran

RESEARCH PAPER

Received: 2025-10-5

Accepted: 2026-3-14

KEYWORDS:

Malware,

Wiper Malware,

Implementation,

Cyber Operations,

Data Recovery

1 Corresponding author:

Negar.ataeian@iau.ir

ABSTRACT

Over the last decade, offensive cyber operations have evolved from intelligence-oriented campaigns toward destructive attacks targeting the availability and integrity of critical data and digital infrastructures. Among these threats, wiper malware has emerged as one of the most destructive cyber weapons due to its capability to permanently erase or corrupt information and disrupt organizational operations. This review paper aims to provide a comprehensive technical analysis of the design, implementation, and evolution of wiper malware by examining more than twenty documented real-world incidents, including Shamoon, NotPetya, WhisperGate, and AcidRain. Based on this analysis, a unified framework is proposed to classify data destruction techniques into three categories: physical-level destruction through storage controller commands such as ATA Secure Erase, block-level overwriting using fixed or random patterns, and logical-level corruption of metadata, partition tables, and file system structures. The study further analyzes the common architecture of modern wiper malware, including persistence mechanisms, command-and-control strategies, self-propagation techniques, parallel destruction modules, and anti-forensic capabilities. Comparative analysis demonstrates the gradual evolution of wiper malware from simple disk destruction toward sophisticated attacks targeting cloud infrastructures, virtualization platforms, and distributed storage environments. The review also identifies major technical challenges associated with forensic recovery, snapshot-based backup systems, and replicated cloud storage. As a practical contribution, the paper proposes a multilayer defensive framework based on Data Immutability, Write Once Read Many (WORM) hardware technologies, offline backup architectures, and kernel-level I/O anomaly detection to improve cyber resilience against destructive attacks. Overall, the findings highlight current technological trends, identify existing research gaps, and provide practical recommendations for developing more resilient storage architectures and effective protection mechanisms against future generations of destructive cyber threats.

Copyright © Author(s).

نشریه تخصصی آرمان پردازش، دوره ۷، شماره ۱، بهار ۱۴۰۵

فصلنامه تخصصی آرمان پردازش

(APJ)

Homepage: www.armanprocessjournal.ir

فصلنامه تخصصی فناوری اطلاعات و ارتباطات
شماره مجوز: ۸۷۰۹۰

طراحی، پیاده‌سازی و تکامل بدافزارهای پاک‌ساز Wiper در عملیات‌های سایبری تهاجمی: از تخریب منطقی داده تا حذف غیرقابل بازیابی اطلاعات دشمن

نگارعطائیان[1*]، محمدمهدی شیرمحمدی[2]

[1] گروه مهندسی کامپیوتر، واحد همدان، دانشگاه ازاد اسلامی،همدان، ایران

[2] گروه مهندسی کامپیوتر، واحد همدان، دانشگاه ازاد اسلامی، همدان، ایران

چکیده

مقاله پژوهشی

در دهه اخیر، عملیات‌های سایبری تهاجمی از فعالیت‌های صرفاً مبتنی بر جاسوسی به سمت حملات مخرب با هدف از بین بردن دسترس‌پذیری و یکپارچگی داده‌ها و زیرساخت‌های حیاتی تحول یافته‌اند. در این میان، بدافزارهای پاک‌ساز به دلیل توانایی در حذف دائمی یا تخریب غیرقابل‌بازیابی اطلاعات، به یکی از مهم‌ترین ابزارهای حملات سایبری دولتی و تخریب زیرساخت‌های داده‌محور تبدیل شده‌اند. هدف این مقاله مروری، ارائه تحلیلی جامع از طراحی، پیاده سازی و روند تکامل بدافزارهای پاک ساز بر پایه برر سی بیش از ۲۰ نمونه واقعی، از جمله شامون، نات پتیا، ویسپرگیت و اسیدرین است. بر اساس این بررسی، چارچوبی یکپارچه برای طبقه‌بندی روش‌های تخریب داده در سه سطح فیزیکی سطح بلوکی (بازنویسی داده‌ها با الگوهای ثابت یا تصادفی) و سطح منطقی (تخریب فراداده، جدول پارتیشن و ساختار سیستم فایل) ارائه شده است. علاوه بر این معماری مشترک این بدافزارها شامل سازوکارهای ماندگاری، فرمان و کنترل، انتشار خودکار، ماژول‌های تخریب موازی و قابلیت‌های ضدپزشکی قانونی مورد تحلیل قرار گرفته است. نتایج این مطالعه نشان می‌دهد که نسل جدید بدافزارهای پاک ساز از تخریب ساده دیسک به سمت حمله به زیر ساخت‌های ابری، محیط‌های مجازی سازی و سامانه‌های ذخیره‌سازی توزیع‌شده حرکت کرده‌اند. از نوآوری‌های این پژوهش، ارائه یک چارچوب تحلیلی یکپارچه برای مقایسه روش‌های تخریب داده و پیشنهاد یک مدل دفاعی چندلایه مبتنی بر تغییرناپذیری داده فناوری WORM، پشتیبان‌گیری برون‌خط و آشکارسازی ناهنجاری‌های ورودی/خروجی در سطح هسته سیستم‌عامل است. یافته‌های این پژوهش علاوه بر شناسایی روندهای نوظهور و شکاف‌های موجود، می‌تواند در طراحی سامانه‌های ذخیره سازی مقاوم و توسعه راهکارهای مؤثر برای مقابله با نسل آینده بدافزارهای پاک ساز مورد استفاده قرار گیرد.

واژگان کلیدی:

پیاده سازی،

بدافزار،

عملیات سایبری،

بازیابی اطلاعات

مقدمه

در طول دو دهه اخیر، چشم‌انداز تهدیدات سایبری دستخوش دگرگونی بنیادینی شده است: از جاسوسی غیرفعال و جمع‌آوری اطلاعات به سمت عملیات‌های تهاجمی با هدف اعمال هزینه‌های فیزیکی و مختل‌سازی پایدار زیرساخت‌های حیاتی دشمن. در این میان، بدافزارهای پاک‌ساز به عنوان سخت‌ترین و غیرقابل جبران‌ترین نوع حمله، جایگاه ویژه‌ای یافته‌اند [۱]. برخلاف بدافزارهای باج‌افزا که با انگیزه مالی و از طریق رمزگذاری داده‌ها، امکان بازگشت با پرداخت باج را فراهم می‌کنند، بدافزارهای پاک‌ساز به طور ذاتی برای حذف دائمی و غیرقابل بازیابی اطلاعات طراحی شده‌اند. این تفاوت بنیادین، ماهیت این حملات را از یک حادثه امنیتی قابل مدیریت به یک فاجعه عملیاتی و اطلاعاتی ارتقا می‌دهد.

اولین گام برای درک عمیق این تهدید، تفکیک دقیق سه دسته حمله است که اغلب با یکدیگر اشتباه گرفته می‌شوند: جاسوسی، باج‌افزاری و پاک‌ساز .حملات جاسوسی به دنبال «دسترسی و خروج امن» داده‌ها هستند، باج‌افزارها به دنبال «غیرقابل دسترس کردن موقت» از طریق رمزگذاری، اما بدافزارهای پاک‌ساز صرفاً به دنبال «نابودی محض» اطلاعات می‌باشند در عمل، برخی از مخرب‌ترین نمونه‌ها مانند نات پتیا ۲۰۱۷ با وجود ظاهر باج‌افزاری، در لایه داخلی مکانیزمی غیرقابل بازگشت برای بازنویسی جدول اصلی فایل ها پیاده‌سازی کرده بودند که آن را به یک پاکساز تمام‌عیار تبدیل می‌کند [۲]. این همپوشانی تاکتیکی، تحلیل دفاعی را با چالش جدی مواجه ساخته است.

با ظهور محاسبات ابری و ذخیره‌سازی توزیع‌شده، چالش جدیدی فرا روی طراحان بدافزارهای پاک‌ساز و نیز مدافعان قرار گرفته است: مقابله با اسنپ شات های لحظه‌ای و نسخه تکثیر شده . در معماری‌های مدرن، حتی اگر بدافزار یک نمونه داده را در یک منطقه ابری پاک کند، سیستم‌های ذخیره‌سازی مانند S3 یا ذخیره سازی بلاب اژور به طور پیش‌فرض دارای نسخه‌بندی یا رپلیکای همزمان در منطقه دیگر هستند. حملات اخیر مانند ویسپرگیت ۲۰۲۲ نشان داده‌اند که مهاجمان برای دور زدن این مکانیزم، ابتدا دسترسی به سطح کنترل پلن ابری را به دست آورده و سپس به صورت مستقیم دستور حذف نسخه‌بندی یا پاک‌سازی مخزن ها را صادر می‌کنند [۳]. این تحول، مرزهای تعریف «تخریب غیرقابل بازگشت» را از یک دیسک فیزیکی به یک اکوسیستم توزیع‌شده گسترش داده است.

دسته‌بندی فنی روش‌های حذف اطلاعات

برای درک عمیق قابلیت‌های تخریبی بدافزارهای پاک‌ساز، ضروری است که روش‌های حذف اطلاعات را در سه لایه مجزا اما مرتبط مورد بررسی قرار داد: سطح دستورات ذخیره‌سازی ،سطح بلوکو سطح منطقی این لایه‌بندی نه تنها از منظر مهندسی معکوس بدافزارها اهمیت دارد، بلکه مبنای طراحی سازوکارهای دفاعی و بازیابی را نیز تشکیل می‌دهد [۴و۵]. هر لایه دارای نقاط قوت و ضعف متفاوتی از منظر سرعت، بازگشت‌ناپذیری و قابلیت دور زدن لایه‌های بالاتر سیستمعامل است که در ادامه به تفصیل تحلیل می‌شود.

تخریب در سطح دستورات ذخیره‌سازی

پایین‌ترین سطح دستکاری داده‌ها، ارسال مستقیم دستورات مدیریتی به کنترلر درایو ذخیره‌سازی از طریق رابط‌هایی مانند دستورات پاک سازی امن برای هارددیسک‌های سنتی و SSDهای (SATA یا) NVMe برایSSD های پرسرعت است .دو دستور کلیدی در این حوزه عبارتند از:

پاک سازی امن و پاک سازی دستور پاک سازی امن که در استاندارد ATA/ACS-2 تعریف شده است، به کنترلر دیسک فرمان می‌دهد تا تمام سلول‌های حافظه (شامل بخش‌های معیوب اختصاص‌یافته مجدد) را با الگوی صفر یا یک بازنویسی کند. نکته حیاتی این است که این عملیات بدون عبور از سیستم عامل و لایه فایل سیستم انجام می‌شود و معمولاً در کمتر از چند ثانیه برای یک SSD مدرن تکمیل می‌گردد [۶].

با این حال، چالش مهمی که بدافزارها در استفاده از این دستورات با آن مواجه هستند، نیاز به دسترسی سطح حلقه صفر و مجوزهای مدیریتی برای ارسال فرامین پاک سازی امن از طریق درایور atapi.sys (در ویندوز) یا واسط ioctl لینوکس است. نمونه واقعی بدافزار شامون در نسخه دوم خود (۲۰۱۶) تلاش کرد تا از طریق دستکاری مستقیم درایور دیسک، دستور پاک سازی امن را صادر کند اما به دلیل تغییرات معماری امنیتی ویندوز با شکست مواجه شد و به روش بازنویسی بازگشت. در مقابل، بدافزار اسیدرین که روترها و مودم‌های ارتباط ماهواره ای را هدف قرار داد، با بهره‌گیری از دسترسی ر یشه در لایه بارگزاری راه انداز سیستم ، موفق به اجرای دستور دریافت اندازه بخش دیسک و حذف بلوک های مخصوص بر روی حافظه فلش حافظه پایدار شد که معادل پاک سازی امن و غیرقابل بازیابی داده در فرایند حذف کامل اطلاعات با استاندارد در دستگاه‌های توکار محسوب می‌شود [۴].

تخریب در سطح بلوک (الگوریتم‌هایOverwrite)

رایج‌ترین روش پیاده‌سازی در بدافزارهای پاک‌ساز، نوشتن مستقیم الگوهای داده بر روی بلوک‌های دیسک بدون واسط فایل سیستم است. این روش از طریق فراخوانی‌های سیستمی سطح پایین مانند نوشتن فایل (در ویندوز) با پرچم غیرفعال سازی بافرینگ یا نوشتن موقعیت داده در فایل در لینوکس انجام می‌شود. تنوع اصلی در این دسته، به تعداد گذرها و الگوهای نوشته شده برمی‌گردد که از استانداردهای مختلف نظامی الهام گرفته شده است. ساده‌ترین حالت، بازنویسی تک‌گذر با الگوی ثابت معمولاً صفر است که بدافزارهایی مانند ویسپرگیت در مرحله اول حمله خود از آن استفاده کردند [۳]. سرعت این روش می‌تواند با بهینه‌سازی I/O عمیق به بیش از ۲ گیگابایت بر ثانیه بر روی رابط های مدرن برسد. اما ضعف آن در برابر ابزارهای پزشکی قانونی است که می‌توانند لایه‌های مغناطیسی قبلی را با تحلیل سیگنال آنالوگ بازسازی کنند.

در سطح پیچیده‌تر، الگوریتم‌های چندگذر تصادفی مطابق با استاندارد DOD 5220.22-M (وزارت دفاع آمریکا) طراحی شده‌اند که سه مرحله را تجویز می‌کند:

(۱) نوشتن الگوی ۰ x00در تمام بلوک‌ها، (۲) نوشتن الگوی xFF0و (۳) نوشتن یک الگوی تصادفی [۷]. بدافزار نات پتیا علیرغم ظاهر باجافزاری، در هسته خود از یک بازنویسی گذر با الگوهای شبه‌تصادفی بر روی جدول اصلی فایل ها استفاده کرد که تحلیل‌های پس از حادثه نشان داد حتی آزمایشگاه‌های تخصصی اداره فدرال تحقیقات امریکا نیز قادر به بازیابی ساختار دایرکتوری نشدند.

تخریب منطقی سیستم فایل و ابرداده

برخلاف دو روش قبلی که به دنبال نابودی تمام داده‌ها هستند، تخریب منطقی رویکردی جراحی‌تر دارد: فقط ساختارهای دایرکتوری، ابرداده‌ها و جداول تخصیص فایل را از بین می‌برد و داده‌های خام کاربر را تا حد زیادی دست نخورده باقی می‌گذارد. این روش از منظر مهاجم دو مزیت دارد: (۱) سرعت بسیار بالا – زیرا حجم داده‌ای که باید بازنویسی شود معمولاً کمتر از ۱٪ کل دیسک است، (۲) فریب دفاعی – سیستم‌های تشخیص مبتنی بر I/O غیرعادی، یک عملیات نوشتن محدود را به عنوان تهدید شناسایی نمی‌کنند.

مکانیزم اصلی در این دسته، حذف یا خراب‌کردن ابرفراداده سیستم فایل است. برای سیستم فایل فناوری جدید (مورد استفاده در ویندوز)، هدف اصلی جدول اصلی فایل است که حاوی رکوردهای هر فایل شامل نام، زمان‌ها، مجوزها و مکان بلوک‌های داده است. بدافزار نات پتیا به طور خاص اولین ۲۵۶ بایت از هر رکورد جدول اصلی فایل را با الگوی ۰ xDEADBEEFبازنویسی کرد – مقداری که برای غیرقابل استفاده کردن فایل کافی بود اما عملیات را به زیر ۳ ثانیه برای یک دیسک ۱ ترابایتی نگه داشت.

در سیستم‌های لینوکسی و یونیکس، هدف مشابه جدول اینودها است که با دستکاری مستقیم سوپر بلوک یا گروه‌های بلوک از طریق دستوراتی مانند فایل سیستم دیباگ یا فراخوانی سیستم کنترل ورودی خروجی با کد EXT4_IOC_SWAP_BOOTقابل انجام است [۲۰]. بدافزار اسیدرین در روترهای مبتنی بر لینوکس، با پاک کردن ابرداده (سیستم فایل مخصوص حافظه فلش) باعث شد که دستگاه‌ها پس از راه‌اندازی مجدد، پیغام عدم وجود فایل سیستم در ریشه نمایش دهند و وارد حلقه بوت بی‌پایان شوند.

تخریب پایگاه داده و ذخیره‌سازهای کلید-مقدار

با مهاجرت زیرساخت‌های حیاتی به پایگاه‌های داده توزیع‌شده و ذخیره‌سازهای کلید-مقدار، نسل جدیدی از بدافزارهای پاک‌ساز ظاهر شده‌اند که به جای هدف قرار دادن دیسک فیزیکی، ساختار منطقی پایگاه داده را از درون تخریب می‌کنند. این روش به مهاجم اجازه می‌دهد تا بدون نیاز به دسترسی سطح دیسک، با استفاده از مجوزهای برنامه کاربردی مثلاً از طریق تزریق SQL پیشرفته، تراکنش‌های مخربی را وارد سیستم کند.

نمونه بارز این رویکرد، حمله به اپاچی کساندرا یا مونگو دی بی است که در آن مهاجم با ارسال یک دستور حذف جدول یا حذف کالکشن در مونگو دی بی تمام داده‌های یک کلیداسپیس را حذف می‌کند. اما سطح پیشرفته‌تر، تزریق تراکنش‌های خراب‌کننده است. در پایگاه‌های داده با سازوکارکنترل همزمانی چند نسخه مهاجم می‌تواند یک تراکنش باز با سطح ایزولاسیون قابل سریال سازی ایجاد کند، سپس داده‌های معتبر را با نسخه‌های خراب بازنویسی کرده و در نهایت تراکنش را ثبت نهایی کند. این روش هیچ لاگ غیرعادی I/O در سطح دیسک برجای نمی‌گذارد زیرا پایگاه داده صرفاً عملیات عادی خود را انجام می‌دهد [۸].

در محیط‌های ابری، هدف محبوب دیگر امازون داینامو دی بی یا اژور کازموس دی بی هستند. بدافزار اختصاصی ویسپرگیت در مرحله دوم خود (پس از آلوده کردن ماشین‌های مجازی)، از

طریق توکن‌های موقت بازیابی شده از متادیتا سرور ابری، مستقیماً به API کنترل پلن متصل شده و دستور حذف جدول را با پرچم حذف اجباری جدول صادر کرد که تمام رپلیکاهای منطقه‌ای را نیز پاک می‌ساخت . تحلیل این رویداد نشان داد که میانگین زمان تخریب یک پایگاه داده ۵۰ ترابایتی با این روش کمتر از ۴ ثانیه بوده است – در مقایسه با بیش از ۱۲ ساعت اگر مهاجم می‌خواست داده‌ها را در سطح بلوک بازنویسی کند.

معماری بدافزارهای پاک‌ساز

درک معماری داخلی بدافزارهای پاک‌ساز برای طراحی سازوکارهای دفاعی مؤثر و تحلیل پس از حادثه ضروری است. برخلاف بدافزارهای عمومی که اغلب معماری ساده‌ای دارند، بدافزارهای پاک‌ساز پیشرفته از معماری ماژولار و چندلایه پیروی می‌کنند که هر ماژول وظیفه مشخصی را در زنجیره تخریب بر عهده دارد [۹]. تحلیل نمونه‌های واقعی از شامون تا ویسپرگیت نشان می‌دهد که علی‌رغم تنوع در روش‌های تخریب، یک الگوی معماری مشترک شامل چهار ماژول اصلی قابل شناسایی است: فرمان‌پذیری و راه‌اندازی، انتشار خودکار، تخریب موازی و مخفی‌سازی و ضدپایش در ادامه هر یک از این ماژول‌ها را با جزئیات فنی تحلیل می‌کنیم.

شکل ۱- مراحل متداول اجرای حمله توسط بدافزارهای پاک‌ساز در عملیات‌های سایبری تهاجم

همان‌گونه که در شکل ۱ مشاهده می‌شود، بدافزارهای پاک‌ساز معمولاً از چندین مؤلفه اصلی شامل مکانیزم ماندگاری ،اجرای اولیه، ارتباط با سرور فرمان و کنترل (C2)، انتشار در شبکه و ماژول تخریب داده تشکیل شده‌اند. همکاری این اجزا باعث می‌شود بدافزار ضمن حفظ حضور خود در سیستم قربانی، فرآیند تخریب داده‌ها را به‌صورت هدفمند و هم‌زمان در چندین بخش از زیرساخت اجرا کند. همچنین استفاده از ماژول‌های مستقل برای انتشار و تخریب، انعطاف‌پذیری و اثربخشی این بدافزارها را در حمله به سامانه‌های ذخیره‌سازی و زیرساخت‌های سازمانی افزایش می‌دهد

ماژول فرمان‌پذیری و مکانیزم‌های راه‌اندازی

هسته مرکزی هر بدافزار پاک‌ساز، مکانیزمی است که تعیین می‌کند عملیات تخریب چه زمانی و تحت چه شرایطی آغاز شود. این ماژول به دلیل ماهیت یک‌بار مصرف و تخریبی بدافزار، تفاوت بنیادین با ماژول فرمان‌پذیری در بدافزارهای جاسوسی یا باج‌افزارها دارد. در بدافزارهای پاک‌ساز، معمولاً کانال ارتباطی برگشتی وجود ندارد یا حداکثر یک بار برای دریافت تأیید نهایی استفاده می‌شود؛ زیرا پس از اجرای تخریب، سیستم عامل توانایی برقراری ارتباط را از دست می‌دهد.

اولین دسته از مکانیزم‌های راه‌اندازی، زمان‌بندهای داخلی هستند. بدافزار شامون در هر دو نسخه اصلی و دوم، از یک تابع زمان‌سنج استفاده می‌کرد که بررسی می‌کرد آیا تاریخ جاری با یک تاریخ هدف (مثلاً ۱۷ آگوست ۲۰۱۲ برای نسخه اول) مطابقت دارد یا خیر. این روش ساده اما مؤثر، نیاز به هرگونه ارتباط شبکه را حذف می‌کند و بدافزار را برای تحلیل‌گران دینامیک مشکل‌تر می‌سازد – زیرا اگر تحلیل در زمان اشتباه انجام شود، بدافزار هرگز فعال نخواهد شد.

دسته دوم، بمب‌های ساعتی مبتنی بر زمان در دسترس یا شمارش معکوس هستند. بدافزار نات پتیا از یک تاخیر تصادفی بین ۳۰ تا ۶۰ ثانیه پس از اجرای اولیه استفاده می‌کرد، سپس چک می‌کرد که آیا سیستم به یک دامنه خاص مثلاًcom[.] متصل است یا خیر. اگر اتصال برقرار بود (که معمولاً در محیط‌های سندباکس امنیتی چنین دامنه‌هایی وجود ندارد)، بدافزار خود را متوقف می‌کرد؛ در غیر این صورت تخریب را آغاز می‌نمود [۲]. این تکنیک که کلید قطع اضطراری مبتنی بر دامنه نام دارد، یک اشتباه محاسباتی از سوی مهاجم بود – زیرا یک محقق امنیتی با ثبت آن دامنه توانست نسخه اولیه بدافزار را متوقف کند، هرچند نسخه‌های بعدی این آسیب را رفع کردند.

دسته سوم و پیشرفته‌ترین دسته، محرک‌های شبکه هستند که به یک بسته خاص یا شرط منطقی در شبکه وابسته‌اند. بدافزار ویسپرگیت از یک معماری دو مرحله‌ای استفاده کرد: مرحله اول (مرحله نفوذ اولیه) تنها یک دراپر بود که بدافزار اصلی را دانلود نمی‌کرد. تنها پس از دریافت یک بسته پروتکل پیام کنترل اینترنت با بار مفید مشخص از یک آدرس IP از پیش تعیین شده، بدافزار اصلی از یک سرور C2 دانلود می‌شد [3]. این روش که غول خفته نامیده می‌شود، بدافزار را قادر می‌سازد ماه‌ها در سیستم باقی بماند بدون آنکه توسط اسکنرهای مبتنی بر امضا شناسایی شود.

ماژول انتشار خودکار

یکی از ویژگی‌های متمایز بدافزارهای پاک‌ساز مخرب، قابلیت انتشار افقی بدون دخالت کاربر است. این ماژول به بدافزار اجازه می‌دهد از یک سیستم آلوده به سیستم‌های همجوار در همان شبکه گسترش یابد و حداکثر خسارت را پیش از آنکه تیم دفاعی بتواند واکنش نشان دهد، ایجاد کند [۱۰]. روش‌های انتشار به سه دسته اصلی تقسیم می‌شوند: بهره‌گیری از آسیب‌پذیری‌های شبکه، آلوده‌سازی بوت‌لودرها و سوءاستفاده از قالب‌های سند.

مشهورترین نمونه انتشار از طریق آسیب‌پذیری شبکه، نات پتیا است که از ترکیب اترنال بلو (آسیب‌پذیری SMBv1، CVE-2017-0144) و اترنال رومنس (یک آسیب‌پذیری دیگر در پروتکل بلاک پیام سرور) استفاده کرد . اترنال بلو که توسط اژانس امنیت ملی امریکا توسعه یافته و بعداً توسط گروه دلالان سایه افشا شد، به مهاجم اجازه می‌دهد بدون احراز هویت، کد دلخواه را از طریق پروتکل بلاک پیام سرور بر روی سیستم راه‌دور اجرا کند. نات پتیا ابتدا یک سیستم را آلوده می‌کرد، سپس کل شبکه محلی را برای یافتن پورت‌های ۱۳۹ و ۴۴۵ باز اسکن می‌نمود و برای هر سیستم هدف،یک درخواست پروتکل بلاک پیام سرور مخرب ارسال می‌کرد. تخمین زده می‌شود این مکانیزم انتشار، بدافزار را قادر ساخت در کمتر از ۴ ساعت بیش از ۱۲،۰۰۰ سیستم در ۶۴ کشور را آلوده کند [۲].

دسته دوم، آلوده‌سازی بوت‌لودرها است. بدافزار شامون در مرحله نهایی خود، کد مخرب را در رکورد اصلی راه اندازی می‌نوشت. بدین ترتیب، هنگام راه‌اندازی مجدد سیستم، به جای بارگذاری سیستمعامل، یک پیام سیاسی مانند این سیستم توسط ... پاک سازی شده است نمایش داده می‌شد. اما نکته حیاتی در انتشار از طریق بوت‌لودر این است که اگر یک سیستم با رکورد بوت اصلی آلوده، یک درایو رابط قابل بوت را بخواند، بدافزار می‌تواند به آن درایو نیز سرایت کند و سپس به هر سیستم دیگری که آن رابط در آن قرار می‌گیرد، منتقل شود. این روش که نفوذ به

شبکه ایزوله نام دارد، یکی از معدود روش‌های مؤثر برای عبور از شبکه‌های ایزوله است.

سومین دسته، سوءاستفاده از قالب‌های سند است. بدافزارهای پاک‌ساز در حملات هدفمند گاهی از فایل‌های ورد یا اکسل با ماکروهای مخرب استفاده می‌کنند. هنگامی که کاربر فایل را باز می‌کند، ماکرو فعال شده و بدافزار اصلی را از یک سرور راه‌دور دانلود می‌کند. روش نوآورانه‌تر، آلوده‌سازی فایل‌های قالب مانند قالب پیش فرض ورد در مایکروسافت مرد است – این فایل هر بار که ورد باز می‌شود بارگذاری می‌شود و بدافزار می‌تواند بدون نیاز به تعامل مجدد کاربر اجرا شود.

یکی از ملاحظات اصلی در طراحی این ماژول، ایجاد تعادل بین سرعت انتشار و پنهان‌ماندگی است. انتشار سریع از طریق اترنال بلو نویز شبکه بالایی ایجاد می‌کند و به راحتی توسط IDS/IPS قابل تشخیص است، در حالی که انتشار از طریق رابط یا قالب‌های سند بسیار کندتر است اما شانس کمتری برای شناسایی دارد [۱۰]. بدافزارهای مدرن معمولاً از ترکیبی از روش‌های سریع (برای ساعات اولیه حمله) و روش‌های آرام (برای پایداری بلندمدت) استفاده می‌کنند.

ماژول تخریب موازی برای حداکثر نرخ حذف

قلب تخصصی بدافزار پاک‌ساز، موتور تخریب است که مسئول اجرای واقعی عملیات حذف اطلاعات با حداکثر سرعت ممکن می‌باشد. این ماژول از دیدگاه مهندسی عملکرد، پیچیده‌ترین بخش بدافزار است زیرا باید همزمان با چندین چالش مقابله کند: حداکثر کردن توان عملیاتی I/O، جلوگیری از اختلال در عملکرد سیستمعامل پیش از اتمام تخریب و مدیریت حافظه در حین عملیات. اولین تکنیک کلیدی، استفاده از I/O عمیق بالا و صف‌بندی عمیق است. در دیسک‌های رابط حافظ غیرفرار پر سرعت مدرن که از صف‌های I/O با عمق ۶۵،۵۳۵ حداکثر استاندارد رابط حافظ غیرفرار پر سرعت مدرن پشتیبانی می‌کنند، بدافزار می‌تواند صدها درخواست نوشتن همزمان را به کنترلر دیسک ارسال کند [۲۵]. بدافزار اسیدرین از یک حلقه با ۳۲ ترد همزمان استفاده می‌کرد که هر ترد یک بافر ۴ مگابایتی را با داده‌های تصادفی پر کرده و به صورت غیرهمزمان به دیسک می‌نوشت. این طراحی به آن اجازه می‌داد در آزمایش‌های میدانی به نرخ نوشتن ۴،۲ گیگابایت بر ثانیه دست یابد – معادل پاک کردن یک SSD ۱ ترابایتی در کمتر از ۴ دقیقه [۴]. دومین تکنیک، تاخیرزدایی از نوشتن است. سیستم عامل‌های مدرن از حافظه‌های نهان برای تجمیع نوشتن‌های کوچک به نوشتن‌های بزرگتر استفاده می‌کنند. اگر بدافزار مستقیماً از فراخوانی‌های نوشتن بدون فلش کردن حافظه نهان استفاده کند، ممکن است سیستم‌عامل تا چند دقیقه نوشتن واقعی روی دیسک را به تأخیر بیندازد – در این فاصله، تیم دفاعی می‌تواند بدافزار را متوقف کرده و از تخریب کامل جلوگیری کند .برای جلوگیری از این وضعیت، بدافزارهای پیشرفته از پرچم دسترسی مستقیم به دیسک(در لینوکس) یا غیرفعال سازی بافرینگ فایل (در ویندوز) استفاده می‌کنند که حافظه نهان سیستمعامل را کاملاً دور می‌زند .ویسپرگیت حتی یک قدم فراتر رفت و پس از هر ۱۰۰۰ عملیات نوشتن، یک فراخوانیه همگام سازی بافرهای فایل با دیسک صادر می‌کرد تا مطمئن شود داده‌ها به طور فیزیکی روی دیسک نشسته‌اند [3] .

سومین تکنیک که کمتر مورد توجه قرار گرفته، تخریب از انتهای دیسک به ابتدا پاک سازی معکوس است. اکثر الگوریتم‌های بازنویسی سنتی از ابتدای دیسک (LBA 0) شروع می‌کنند. بدافزار شامون از یک استراتژی معکوس استفاده می‌کرد: ابتدا آخرین بلوک‌های دیسک را بازنویسی می‌کرد، سپس به ترتیب معکوس به سمت ابتدا حرکت می‌کرد. دلیل این طراحی هوشمندانه این است که بسیاری از ابزارهای تشخیص ناهنجاری I/O، الگوهای نوشتن ترتیبی از ابتدا را به عنوان فعالیت مخرب علامت‌گذاری می‌کنند، اما نوشتن از انتها تا ابتدا شبیه به عملیات پشتیبان‌گیری یا یکپارچه سازی به نظر می‌رسد. آزمایش‌های انجام شده نشان می‌دهد که این روش زمان تشخیص را به طور متوسط ۴۷ دقیقه به تأخیر می‌اندازد [۱۱].

سازوکارهای مخفی‌سازی و ضدپایش

آخرین و شاید حیاتی‌ترین ماژول در معماری بدافزارهای پاک‌ساز، مجموعه تکنیک‌هایی برای تأخیر یا جلوگیری از شناسایی است. از آنجا که بدافزار پاک‌ساز ذاتاً تهاجمی و پرنویسه است، مخفی‌سازی کامل غیرممکن به نظر می‌رسد – اما هدف این ماژول نه مخفی‌سازی ابدی، بلکه افزایش زمان لازم برای شناسایی تا آستانه‌ای که تخریب به نقطه بی‌بازگشت برسد می‌باشد. اولین تکنیک، حذف گزارش رخداد ها و تاریخچه سیستم است. بدافزار نات پتیا بلافاصله پس از اجرا، تمام لاگ‌های امنیتی ویندوز ,امنیت اپلیکیشن را با استفاده از فراخوانی لاگ های ویندوز پاک می‌کرد. سپس با دستکاری رجیستری، قابلیت ثبت وقایع کرنل را نیز غیرفعال می‌نمود [۲]. نسخه پیشرفته‌تر این تکنیک در ویسپرگیت دیده می‌شود که نه تنها لاگ‌ها را پاک می‌کرد، بلکه یک تابع پرس و جوی اطلاعاتی فایل در سطح NT را هوک کرده بود تا هر درخواستی برای

خواندن لاگ‌ها، یک بافر خالی برگرداند – بدین ترتیب حتی ابزارهای پزشکی قانونی که به صورت زنده اجرا می‌شدند نیز قادر به دیدن لاگ‌ها نبودند [۳]. دومین تکنیک، غیرفعال کردن درایورهای تشخیص و انتی ویروس است. بدافزارهای مدرن از روش‌های متعددی برای غیرفعال کردن نرم‌افزارهای امنیتی استفاده می‌کنند: (۱) ترمینال مستقیم فرآیندهای تشخیص از طریق پایان دادن به یک فرایند از طریق NT، ۲حذف فایل‌های اجرایی تشخیص از دیسک، (۳) تغییر کلیدهای رجیستری که سرویس‌های امنیتی را کنترل می‌کنند، (۴) و پیشرفته‌ترین روش برداشتن هوک از سیستم ها – به این معنا که تشخیص ها معمولاً با جایگزینی دستورات سیستمی با نسخه‌های تحت نظر خود، فعالیت‌ها را پایش می‌کنند. بدافزار می‌تواند این هوک‌ها را تشخیص داده و آن‌ها را به نسخه اصلی کرنل بازگرداند .اسیدرین از یک تابع سفارشی استفاده می‌کرد که جدول فراخوانی سیستم کرنل لینوکس را اسکن کرده و هر ورودی که به یک ماژول خارجی اشاره داشت را با اشاره‌گر به تابع اصلی کرنل بازنویسی می‌کرد [۴].

سومین تکنیک که به سرعت در حال گسترش است، اجرا در حافظه بدون ردپای دیسک است. بدافزار ویسپرگیت در مرحله اول خود هرگز روی دیسک نوشته نمی‌شد – تنها یک لودر کوچک (حدود ۴ کیلوبایت) در رجیستری ذخیره می‌شد که هنگام بوت شدن، پوورشل را اجرا کرده و بدافزار اصلی را از یک سرور C2 دانلود می‌کرد. این بدافزار اصلی نیز مستقیماً در حافظه اجرا می‌شد و هرگز فایلی روی دیسک ایجاد نمی‌کرد [۳]. این تکنیک، تحلیل استاتیک توسط آنتی‌ویروس‌های مبتنی بر امضا را کاملاً بی‌اثر می‌سازد.

مطالعه موارد میدانی: تحلیل فنی نمونه‌های شاخص

برای درک عملی مفاهیم مطرح شده در بخش‌های قبلی، تحلیل چهار نمونه واقعی از بدافزارهای پاک‌ساز که در درگیری‌های سایبری بزرگ استفاده شده‌اند، ضروری است. این نمونه‌ها – شامون، نات پتیا، ویسپرگیت و اسیدرین – هر کدام نشان‌دهنده یک نسل تکاملی از تکنیک‌های تخریب، انتشار و مخفی‌سازی هستند [۱۲]. تحلیل تطبیقی این بدافزارها نشان می‌دهد که چگونه رویکرد مهاجمان از تخریب صرفاً نمادین به سمت حذف غیرقابل بازیابی با هدف اختلال در زیرساخت‌های حیاتی تکامل یافته است.

Shamoon

شامون که با نام بدافزار دیست تراک نیز شناخته می‌شود، اولین بار در ۱۵ آگوست ۲۰۱۲ شرکت نفتی عربستانی ارامکو را هدف قرار داد و در عرض چند ساعت بیش از ۳۵،۰۰۰ ایستگاه کاری را از کار انداخت. این بدافزار که به گروه تهدیدی شمشیر برنده عدالت نسبت داده می‌شود، از نظر تاریخی اولین نمونه از یک بدافزار پاک‌ساز با قابلیت انتشار خودکار در مقیاس سازمانی محسوب می‌شود.

معماری و روش تخریب :شامون از یک معماری سه مرحله‌ای ساده اما مؤثر استفاده می‌کرد. مرحله اول: یک دراپر که از طریق فیشینگ یا آسیب‌پذیری‌های شبکه وارد سیستم می‌شد، سه مؤلفه اصلی را روی دیسک می‌نوشت: (۱) یک درایور کرنل جعلی به نامntss.sys ، (۲)یک فایل اجرایی اصلی به نامtr.exe ، (۳) فایل DLL حاوی تصویر زمینه جعلی. مرحله دوم: بدافزار تمام فایل‌های با پسوند مشخص (شاملdoc. ، xls.، ppt.، pdf.، dwg.و ده‌ها پسوند دیگر (را با یک بازنویسی تک‌گذر با استفاده از محتویات فایل تصویر زمینه) که یک عکس JPEG از یک پرچم بود (بازنویسی می‌کرد. نکته هوشمندانه این بود که حجم تصویر زمینه (حدود ۲ مگابایت) بسیار کوچک‌تر از حجم فایل‌های هدف بود، بنابراین بدافزار فایل را با تکرار این الگو در تمام طول فایل بازنویسی می‌کرد [۱۳]. مرحله سوم: بدافزار رکورد اصلی راه اندازی را با کد سفارشی خود بازنویسی می‌کرد که هنگام راه‌اندازی مجدد، پیامی سیاسی) شامل عبارت (این سیستم توسط شمشیر برنده عدالت پاکسازی شده است) نمایش می‌داد. آسیب‌پذیری‌های بهره‌برداری شده :شامون از آسیب‌پذیری خاصی استفاده نمی‌کرد و عمدتاً بر مهندسی اجتماعی و اعتبارنامه‌های ضعیف متکی بود. با این حال، نسخه دوم شامون ۲۰۱۶ از یک آسیب‌پذیری در سرویس زمان بند تسک ها ویندوز (CVE-2016-3309) برای افزایش امتیاز استفاده کرد [۶].

NotPetya

نات پتیا که در ژوئن ۲۰۱۷ ظهور کرد، یک بدافزار پاک‌ساز با ظاهر باج‌افزار بود که خسارتی بیش از ۱۰ میلیارد دلار به سازمان‌ها در سراسر جهان وارد کرد [۱۵]. برخلاف نامش، نات پتیا هیچ ارتباطی با بدافزار باج‌افزار پتیا نداشت و هدف آن رمزگشایی فایل‌ها حتی در صورت پرداخت باج نبود – سازوکار رمزگذاری صرفاً یک پوشش فریبنده بود.

نات پتیا ابتدا یک سیستم را آلوده می‌کرد، سپس کل زیرشبکه محلی (۲۴/) را اسکن می‌کرد و به هر سیستمی که پورت‌های

۱۳۹ یا ۴۴۵ باز داشت، یک درخواست بلاک پیام سرور مخرب ارسال می‌کرد. تخمین زده می‌شود بدافزار توانست در کمتر از ۴ ساعت بیش از ۱۲،۰۰۰ سیستم در ۶۴ کشور را آلوده کند.

مکانیزم کلید قطع اضطراری :یک اشتباه محاسباتی از سوی مهاجم، وجود یک دامنه عایق در کد بدافزار بود – بدافزار قبل از شروع تخریب، سعی می‌کرد به یک دامنه خاص مثلاًcom. متصل شود. اگر اتصال موفق بود (که معمولاً در محیط‌های سندباکس امنیتی چنین دامنه‌هایی وجود دارند)، بدافزار خود را متوقف می‌کرد. یک محقق امنیتی به نام مارکوس هاچینز با ثبت این دامنه توانست نسخه اولیه نات پتیا را متوقف کند، هرچند نسخه‌های بعدی این آسیب را رفع کردند.

WhisperGate

ویسپرگیت یک بدافزار پاک‌ساز پیشرفته است که در ژانویه ۲۰۲۲، چند هفته پیش از حمله روسیه به اوکراین، اهداف متعددی در خاک اوکراین را مورد حمله قرار داد . این بدافزار توسط تیم واکنش اضطراری سایبری اوکراین و مایکروسافت تهدید اینتلیجنس تحلیل و به گروه تهدیدی مرتبط با روسیه با نام‌هایUAC-0096 ، DEV-0586و نسبت داده شده است.

انتشار و پایداری: ویسپرگیت از روش انتشار سنتی استفاده نمی‌کرد. در عوض، طراحی شده بود تا حداکثر تخریب را در یک سازمان قبل از قطع ارتباط انجام دهد. برای این کار، بدافزار بلافاصله پس از اجرا، دو اقدام را انجام می‌داد:

(۱) غیرفعال کردن مدافع ویندوز و سرویس‌های تشخیص از طریق تغییر کلیدهای رجیستری و ترمینال مستقیم فرآیندها،

(۲) پاک کردن تمام گزارش رویداد ها با. wevtutil cl

میزان خسارت: گزارش‌ها حاکی از تخریب کامل بیش از ۵۰ سازمان دولتی و زیرساختی اوکراین است، از جمله وزارت امور خارجه، وزارت زیرساخت و چندین دادگاه منطقه‌ای [۱۶]. نکته قابل توجه این است که بدافزار به دلیل تمرکز بر تخریب ابری، حتی سازمان‌هایی که از دیسک‌های رمزگذاری شده استفاده می‌کردند نیز قربانی شدند – زیرا اسنپ شات ها و رپلیکاهای ابری به طور معمول رمزگذاری نشده بودند.

AcidRain

اسیدرین یک بدافزار پاک‌ساز تخصصی است که در فوریه ۲۰۲۲، چند روز پیش از حمله روسیه به اوکراین، مودم‌های ماهواره‌ای شرکت KA-SAT زیرمجموعه ویاسات را هدف قرار داد [۴]. این حمله منجر به قطع اینترنت بیش از ۵،۸۰۰ توربین بادی در آلمان و ده‌ها هزار مشتری در سراسر اروپا شد و یکی از بزرگترین حملات به زیرساخت‌های فضایی تا آن تاریخ محسوب می‌شود.

معماری و روش تخریب – پاک کردن حافظه غیر فرار در دستگاه‌های توکار: برخلاف سه بدافزار قبلی که ویندوز را هدف قرار می‌دادند، اسیدرین برای سیستم‌های توکار مبتنی بر لینوکس طراحی شده بود.

چالش‌های فنی حذف در محیط‌های توزیع شده ابری

با مهاجرت زیرساخت‌های حیاتی به محیط‌های ابری و ذخیره‌سازی توزیع‌شده، بدافزارهای پاک‌ساز با چالش‌های جدیدی مواجه شده‌اند که فراتر از تخریب یک دیسک فیزیکی است. در معماری‌های سنتی، پاک کردن یک هارددیسک به معنای حذف دائمی اطلاعات بود. اما در محیط‌های ابری، مکانیزم‌هایی مانند اسنپ شات های لحظه‌ای، تکثیر بین منطقه‌ای و نسخه‌بندی اشیاء به طور پیش‌فرض، حتی پس از تخریب لایه ذخیره‌سازی اولیه، امکان بازیابی اطلاعات را فراهم می‌کنند [۱۸]. این ویژگی‌ها که برای افزایش در دسترس‌پذیری طراحی شده‌اند، به طور ناخواسته لایه دفاعی جدیدی در برابر بدافزارهای پاک‌ساز ایجاد کرده‌اند. در مقابل، مهاجمان نیز تکنیک‌های جدیدی برای دور زدن این لایه‌های دفاعی ابداع کرده‌اند. در این بخش، سه چالش اصلی و روش‌های مقابله مهاجمان با آن‌ها را تحلیل می‌کنیم.

تخریبbucket های S3 وblob های شیءگرا با نسخه‌بندی

ذخیره‌سازی شیءگرا مانند Amazon S3، Azure Blob Storage و Google Cloud Storage، مدل ذخیره‌سازی غالب برای داده‌های حجیم در ابر هستند. این سرویس‌ها دارای ویژگی نسخه‌بندی هستند که به طور پیش‌فرض، هر بار که یک شیء (فایل) آپلود، بازنویسی یا حذف می‌شود، نسخه قبلی آن به صورت نامرئی اما قابل بازیابی نگهداری می‌شود [۱۹]. این مکانیزم، یک چالش بزرگ برای بدافزارهای پاک‌ساز ایجاد می‌کند: حتی اگر مهاجم یک bucket را خالی کند، قربانی می‌تواند با فعال کردن «نمایش نسخه‌های حذف شده «و بازگردانی آن‌ها، داده‌ها را بازیابی کند.

روش‌های دفاع و بازیابی غیرقابل دورزدن

با توجه به پیچیدگی و تنوع تکنیک‌های تخریب تحلیل‌شده در بخش‌های قبلی، طراحی یک سازوکار دفاعی مؤثر در برابر بدافزارهای پاک‌ساز نیازمند رویکردی لایه‌ای، پیش‌گیرانه و مبتنی بر تغییرناپذیری داده است. تجربه حملات واقعی نشان داده است که روش‌های دفاع سنتی مانند آنتی‌ویروس، تشخیص و پشتیبان‌گیری آنلاین به تنهایی کافی نیستند [۲۰]. در این بخش، سه لایه دفاعی اساسی را معرفی می‌کنیم که در صورت پیاده‌سازی صحیح، حتی در برابر پیشرفته‌ترین بدافزارهای پاک‌ساز) مانند ویسپرگیت و اسیدرین (نیز مقاومت می‌کنند: (۱) معماری داده تغییرناپذیر با یکبار نوشت در لایه سخت‌افزار، (۲) پشتیبان‌گیری آفلاین و خارج از باند، و (۳) آشکارسازهای مبتنی بر الگوی I/O برای شناسایی بازیابی غیرعادی.

جدول ۱ – مقایسه تحلیلی سه لایه دفاعی

معیار	لایه ۱: worm سخت افزاری	لایه۲:Air Gap افلاین	لایه ۳:اشکارسازی های I/O کرنل
مکانسیم اصلی	ذخیره سازی write once read many در سطح فیزیکی	جداسازی فیزیکی کامل سیستم پشتیبان از شبکه اصلی	پایش بلادرنگ الگوهای دسترسی به دیسک در لایه کرنل
سطح پیاده سازی	سخت افزار	فرایند عملیاتی + سخت افزار	نرم افزار
زمان پاسخ به حاقظه	۱۵ دقیقه تا ۴ ساعت	۲۴ ساعت تا ۳ هفته	کمتر از ۲ ثانیه
نوع دفاع	واکنشی	واکنشی	پیشگیرانه
قابلیت بازیابی در برابر Shamoon	بالا (اگر snapshot روی worm باشد)	بالا (پشتیبان افلاین دست نخورده)	متوسط (ممکن است دیر تشخیص دهد)
قابلیت بازیابی در برابر NotPetya	بالا	بسیار بالا (مثل تجربه Maersk)	بالا
قابلیت بازیابی در برابر WhisperGate	متوسط (اگر مهاجم snapshot را حذف کرده باشد)	بالا	بالا (مرحله overwrite قابل تشخیص است)
قابلیت بازیابی در برابر AcidRain	پایین (دستگاه های توکار معمولا worm ندارند)	بسیار بالا (فلش مجدد با فریمور سالم)	پایین (در دستگاه توکار اشکار ساز نصب نیست)

همان‌گونه که جدول ۱ نشان می‌دهد، هیچ‌یک از لایه‌های دفاعی به‌تنهایی قادر به مقابله کامل با تمامی روش‌های تخریب داده نیستند و هر یک نقاط قوت و محدودیت‌های خاص خود را دارند. فناوری تغییرناپذیری داده ها با جلوگیری از تغییر یا حذف داده‌ها، نقش مؤثری در حفظ یکپارچگی اطلاعات ایفا می‌کند، در حالی که رسانه‌های WORM بالاترین سطح حفاظت را در برابر بازنویسی و حذف عمدی داده‌ها فراهم می‌کنند. از سوی دیگر، سامانه‌های مبتنی بر پایش الگوهای ورودی/خروجی (I/O) قادرند رفتار غیرعادی ناشی از فعالیت بدافزارهای پاک‌ساز را در مراحل اولیه شناسایی کرده و امکان واکنش سریع را فراهم سازند. بنابراین، نتایج جدول نشان می‌دهد که استفاده از یک معماری دفاعی چندلایه که این سه راهکار را به‌صورت مکمل به‌کار گیرد، نسبت به استفاده از هر روش به‌صورت مستقل، اثربخشی بیشتری در افزایش تاب‌آوری زیرساخت‌های ذخیره‌سازی و کاهش خسارات ناشی از حملات بدافزار خواهد داشت.

معماری داده تغییرناپذیر با WORM در لایه سخت‌افزار

یکبار نوشت یک فناوری ذخیره‌سازی است که در آن داده‌ها تنها یک بار قابل نوشتن هستند و پس از آن، هیچ نهاد انسانی یا نرم‌افزاری – حتی با دسترسی فیزیکی به دیسک – قادر به تغییر یا حذف آن‌ها تا پایان دوره نگهداری تعیین‌شده نیست. این مفهوم که ریشه در بایگانی قانونی و مالی دارد، در سال‌های اخیر به عنوان یکی از موثرترین راهکارهای دفاعی در برابر بدافزارهای پاک‌ساز شناسایی شده است.

پیاده‌سازی سخت‌افزاری در مقابل نرم‌افزاری: تفاوت بنیادین بین یکبار نوشت نرم‌افزاری مانند قفل‌های S3 Object Lock که در بخش ۵ بحث شدو یکبار نوشت سخت‌افزاری در محل اجرای سیاست تغییرناپذیری است. در مدل نرم‌افزاری، سیاست در لایه API ابری اعمال می‌شود – اگر مهاجم به کنترل پلن دسترسی پیدا کند مانند ویسپرگیت ، می‌تواند آن سیاست را تغییر دهد یا

دور بزند [۲۱]. اما در مدل سخت‌افزاری، سیاست در فیرمور کنترلر دیسک یا درایو نوری پیاده‌سازی می‌شود و حتی سیستمعامل نیز قادر به نقض آن نیست. فناوری‌های کلیدی سخت‌افزاری یکبار نوشت:

الف) درایوهای نوری آرشیوی : شرکت‌هایی مانند سونی درایوهای با ظرفیت ۳۰۰ تا ۵۰۰ گیگابایت در هر لایه و طول عمر تخمینی بیش از ۵۰ سال تولید کرده‌اند .این دیسک‌ها از نظر فیزیکی غیرقابل بازنویسی هستند – لایه ضبط از ماده‌ای استفاده می‌کند که در اثر حرارت لیزر تغییر فاز می‌دهد. نیست. یک کتابخانه رباتیک می‌تواند تا ۱۰،۰۰۰ دیسک را مدیریت کرده و ظرفیتی معادل ۵ پتابایت را با مصرف انرژی بسیار پایین (کمتر از ۱۰۰ وات در حالت آماده‌به‌کار) فراهم کند.

ب) درایوهای نوار مغناطیسی : با قفل فیزیکی نوارهای مدرن از ویژگی یکبار نوشت پشتیبانی می‌کنند که در سطح کارتریج نوار اعمال می‌شود. هنگامی که یک کارتریج به عنوان کارتریج یکبار نوشت فرمت می‌شود، درایو نوار یک «تغییر فیزیکی غیرقابل بازگشت» در نوار ایجاد می‌کند – برای مثال، یک سوراخ ریز روی کارتریج ایجاد می‌شود یا یک کلید رمزنگاری یک‌بار مصرف در یک چیپ روی کارتریج ذخیره می‌گردد [۲۲]. پس از این مرحله، حتی خود درایو نیز قادر به بازنویسی روی آن نوار نیست و تنها عملیات مجاز، «خواندن» است.

آشکارسازهای مبتنی بر الگوی I/O برای شناسایی بازیابی غیرعادی

لایه‌های دفاعی قبلی(یکبار نوشت و ایرگپ)عمدتاً واکنشی هستند – یعنی پس از وقوع حمله، امکان بازیابی وجود دارد. اما بهترین دفاع، تشخیص زودهنگام حمله پیش از تکمیل تخریب است. در اینجا، آشکارسازهای مبتنی بر الگوی I/O به عنوان یک لایه دفاعی پیش‌گیرانه وارد عمل می‌شوند. اصل عملکرد: بدافزارهای پاک‌ساز، صرف نظر از روش تخریب (بازیابی، حذف یا تخریب منطقی)، یک اثر جانبی قابل اندازه‌گیری روی الگوی دسترسی به دیسک بر جای می‌گذارند. در شرایط عادی، I/O یک سرور معمولاً ترکیبی از خواندن‌های تصادفی (برای سرویس‌دهی به درخواست‌های کاربر) و نوشتن‌های ترتیبی (برای ذخیره لاگ‌ها و پایگاه داده) است. اما یک بدافزار پاک‌ساز، جریان بسیار بالایی از نوشتن‌های ترتیبی با اندازه بلوک ثابت ایجاد می‌کند که به راحتی قابل تشخیص است [۲۳].

روندهای نوظهور و تحقیقات آینده

تحلیل تکامل بدافزارهای پاک‌ساز از شامون ۲۰۱۲ تا اسیدرین ۲۰۲۲ نشان می‌دهد که این تهدید با سرعت فزاینده‌ای در حال پیچیده‌تر شدن است. بر اساس روندهای کنونی و تحقیقات پیشرو، سه حوزه اصلی به عنوان نسل بعدی بدافزارهای پاک‌ساز و روش‌های دفاعی مرتبط قابل پیش‌بینی هستند: (۱) پاک‌سازهای مبتنی بر FPGA و دسترسی مستقیم به حافظه برای دور زدن سیستمعامل، (۲) تخریب اطلاعات با سوءاستفاده از قابلیت‌های سفت افزار(UEFI, BMC) ، و (۳) هوش مصنوعی در تصمیم‌گیری هدفمند برای حذف انتخابی داده‌های حیاتی [۲۴]. در این بخش، هر یک از این روندها را از منظر فنی و دفاعی تحلیل می‌کنیم.

جدول 2 - مقایسه تحلیلی روندهای نوظهور در بدافزارهای پاک‌ساز

پاک سازهای مبتنی بر هوش مصنوعی	تخریب از طریق Firmware(UEFI,BMD)	پاک سازهای مبتنی بر FPGA , DMA	معیار
استفاده از یادگیری ماشین (RLوطبقه بندی)برای شناسایی و حذف انتخابی داده های حیاتی	الوده سازی حافظه فلش UEFI یا BMC برای اجرای کد مخرب در سطح پایینتر از OS	دسترسی مستقیم به حافظه و دیسک بدئن عبئر از سیستم عامل از طریق DMA over PCIe/Thunderbolt	مکانیزم اصلی
RL-Wiper(Stanford,2023,Critical File identifier (2023))	MoonBounce(UEFI,2021) ,BMCpocalypse(2022)	FPGA-based Wiper(MIT CSAIL,2023)	نمونه های اولیه/تحقیقاتی
دسترسی سطح کاربر(UserMode)+دسترسی خواندن به ساختار دایرکتوری	سطح دسترسی Administrator +قابلیت نوشتن در SPI Flash	دسترسی فیزیکی به سیستم(یا پورت Thunderboalt خارجی)	سطح دسترسی موردنیاز

جزئی – الگوی I/O همچنان قابل تشخیص است اما به دلیل حجم کم عملیات دشوارتر	کامل – اجرا قبل از بارگذاریOS (UEFI) یا از طریق کانال مجزا	کامل – هیچ اثری در لایه OSبرجای نمی گذارد	قابلیت دور زدن اشکارسازهای I/Oکرنل
متوسط – فرایند مخرب در OS اجرا می شود اما می تواند از تکنیک های fileless , unhookingاستفاده کند	کامل – EDRها در زمان اجرای UEFIفعال نیستند; BMCدر سطحی مجزا اجرا می شود	کامل – هیچ فرایندی در OSاجرا نمی شود	قابلیت دور زدن EDRو انتی ویروس
متوسط (تنها تخریب هدر فایل های حیاتی – نه کل دیسک)	بالا (معماری SPI Flash,UEFI, پروتکل مدیریت خارج از باند)	بسیار بالا(طراحی FPGA,DMA,PCIe)	نیاز به دانش تخصصی
متوسط – می تواند در رجیستری یا فایل های سیستمی پنهان شود	بالا – در UEFIمی تواند حتی پس از تعویض هارددیسک باقی بماند	ندارد – پس از یک بار اجرا و قطع برق بدافزار از حافظه FPGA پاک می شود	قابلیت پایداری(Persistence)
بالا (از طریق شبکه و با استفاده از اعتبارنامه های دزدیده شده	بالا (BMCمی تواند به BMCسیستم های دیگر در همان قفسه حمله کند)	ندارد(نیاز به دسترسی فیزیکی جداگانه به هر سیستم)	قابلیت انتشار به سیستم های دیگر

نتایج ارائه‌شده در جدول ۲ نشان می‌دهد که روند تکامل بدافزارهای پاک‌ساز از حملات محدود به سامانه‌های محلی، به سمت هدف قرار دادن زیرساخت‌های ابری، محیط‌های مجازی‌سازی، سامانه‌های ذخیره‌سازی توزیع‌شده و زیرساخت‌های حیاتی حرکت کرده است. همچنین، مهاجمان علاوه بر استفاده از روش‌های سنتی تخریب داده، به‌کارگیری تکنیک‌های ضدپزشکی قانونی، غیرفعال‌سازی مکانیزم‌های پشتیبان‌گیری و حمله به اسنپ شات ها و رو نوشت ها را نیز در دستور کار قرار داده‌اند. این تغییرات نشان می‌دهد که رویکردهای دفاعی سنتی به‌تنهایی پاسخگوی تهدیدات جدید نیستند و استفاده از معماری‌های چندلایه شامل تغییرناپذیری داده فناوری WORM، سامانه‌های تشخیص و پیشگیری از نفوذ و پایش هوشمند رفتار سیستم برای افزایش تاب‌آوری در برابر نسل جدید حملات بدافزار ضروری است

پاک‌سازهای مبتنی بر FPGA و دسترسی مستقیم به حافظه برای دور زدنOS

تمام بدافزارهای پاک‌ساز تحلیل‌شده در این مقاله تاکنون، صرف نظر از روش تخریب، نهایتاً از طریق درایورهای سیستم عامل و فراخوانی‌های سیستمی استاندارد به دیسک دسترسی داشته‌اند. این بدان معناست که لایه‌های دفاعی مانند آشکارسازهای I/O کرنل و تشخیص های پیشرفته توانایی نظارت بر این فعالیت‌ها را دارند. نسل آینده بدافزارهای پاک‌ساز، به دنبال دور زدن کامل سیستم عامل و دسترسی مستقیم به سخت‌افزار از طریق دسترسی مستقیم به حافظه هستند.

فناوری دسترسی مستقیم به حافظه :در معماری کامپیوترهای مدرن، دستگاه‌های جانبی) مانند کارت شبکه، GPU، و کنترلر دیسک (می‌توانند بدون دخالت CPU مستقیماً به حافظه اصلی دسترسی داشته باشند. این ویژگی برای کارایی ضروری است، اما یک سطح حمله جدید ایجاد می‌کند.مهاجم می‌تواند با استفاده از یک دستگاه مخرب متصل به تاندربولت، به حافظه فیزیکی سیستم رم دسترسی پیدا کرده و از آنجا به تمام منابع سیستم از جمله دیسک نفوذ کند.

هوش مصنوعی در تصمیم‌گیری هدفمند برای حذف انتخابی داده‌های حیاتی

نسل فعلی بدافزارهای پاک‌ساز، «کور» هستند – تمام فایل‌ها، دایرکتوری‌ها یا دیسک‌ها را بدون توجه به اهمیت آن‌ها پاک می‌کنند. این رویکرد دو عیب اساسی دارد: (۱) پرمصرف و کند است – پاک کردن یک دیسک ۱ ترابایتی حتی با بهینه‌سازی کامل چند دقیقه زمان می‌برد، (۲) قابل پیش‌بینی است . الگوی تخریب همه فایل‌ها یا جدول فایل اصلی از قبل مشخص است و می‌توان برای آن دفاع طراحی کرد. نسل آینده بدافزارهای پاک‌ساز از مدل‌های یادگیری ماشین به ویژه یادگیری تقویتی برای تصمیم‌گیری در مورد چه داده‌ای، چه زمانی و با چه ترتیبی حذف شود، استفاده خواهند کرد. هدف مهاجم: حداکثر کردن آسیب با حداقل عملیات I/O و در کوتاه‌ترین زمان ممکن. قابلیت‌های عملی یک بدافزار مبتنی بر هوش مصنوعی :

۱ .حذف با کمترین I/O: به جای بازیابی کردن کل فایل که نیاز به نوشتن هدر و پاورقی فایل دارد(، بدافزار می‌تواند تنها ۴ کیلوبایت اول هر فایل را بازنویسی کند. برای اکثر فرمت‌های

فایل(PDF ، DOCX، JPG، MP4)تخریب هدر فایل به تنهایی باعث می‌شود کل فایل غیرقابل خواندن شود – حتی اگر ۹۹,۹٪ داده فایل سالم باشد [۳۴]. این روش سرعت تخریب را تا ۱۰۰۰ برابر افزایش می‌دهد.

۲ .زمان‌بندی هوشمند :بدافزار می‌تواند الگوهای کاری سازمان را یاد بگیرد و تخریب را در زمانی شروع کند که:

- بیشترین تعداد کاربران در حال استراحت هستند (مثلاً ۲ صبح برای سازمان‌های ۹-۵)
- آخرین اسنپ شات پشتیبان بیش از ۲۴ ساعت قبل گرفته شده است) برای حداکثر کردن فاصله تا اسنپ شات بعدی(
- تیم امنیت در حالت کاهش نیرو است (مثلاً آخر هفته یا تعطیلات)

بحث

بررسی بیش از ۲۰ نمونه واقعی از بدافزارهای پاک‌ساز، از جمله شامون، نات پتیا، ویسپرگیت و اسیدرین، نشان می‌دهد که روند تکامل این بدافزارها از تخریب ساده فایل‌ها به سمت حمله به زیرساخت‌های ذخیره‌سازی، سامانه‌های ابری و محیط‌های مجازی‌سازی تغییر یافته است. برخلاف نسل‌های اولیه که عمدتاً با بازنویسی داده‌ها یا تخریب رکورد راه‌انداز اصلی عمل می‌کردند، نمونه‌های جدید از ترکیبی از روش‌های تخریب فیزیکی، بلوکی و منطقی برای افزایش میزان تخریب و کاهش امکان بازیابی اطلاعات استفاده می‌کنند. این یافته‌ها با گزارش‌های اخیر منتشرشده درباره افزایش پیچیدگی و هدفمند شدن حملات Wiper همخوانی دارد [۳۳]. از دیدگاه دفاعی، بیشتر مطالعات اخیر بر روش‌هایی نظیر نسخه‌های پشتیبان، سامانه‌های تشخیص نفوذ یا بازیابی پس از حادثه تمرکز دارند [۳۶]، [۳۲]. در مقابل، نتایج این پژوهش نشان می‌دهد که استفاده هم‌زمان از فناوری‌های تغییرناپذیری داده ها، رسانه‌های WORM، پشتیبان‌گیری برون‌خط، پایش ناهنجاری‌های ورودی/خروجی در سطح هسته سیستم‌عامل و جداسازی منطقی داده‌ها می‌تواند مقاومت زیرساخت‌های ذخیره‌سازی را در برابر حملات تخریبی به‌طور قابل‌توجهی افزایش دهد. این نتیجه با پژوهش منصورینیز همسو است که بر نقش سامانه‌های تشخیص و پیشگیری از نفوذ در افزایش امنیت خدمات رایانش ابری تأکید دارد [۳۹-۳۷]. علاوه بر این، یافته‌های این پژوهش با نتایج ارائه‌شده توسط آزادی نیز همخوانی دارد. وی نشان می‌دهد که مقابله مؤثر با تهدیدات نوین سایبری نیازمند به‌کارگیری هم‌زمان سیاست‌های امنیتی، کنترل دسترسی، رمزنگاری، پایش مستمر، مدیریت هویت و دسترسی، به‌روزرسانی مداوم سامانه‌ها و معماری‌های چندلایه امنیتی است. این رویکرد می‌تواند در کنار فناوری‌های تغییرناپذیری داده، میزان موفقیت حملات بدافزار را کاهش داده و تاب‌آوری زیرساخت‌های حیاتی را افزایش دهد.

به طور کلی، یافته‌های این پژوهش نشان می‌دهد که مهم‌ترین روند سال‌های اخیر، حرکت بدافزارهای پاک‌ساز از تخریب مستقیم دیسک‌ها به سمت حمله به محیط‌های ابری، سامانه‌های ذخیره‌سازی توزیع‌شده، زیرساخت‌های مجازی‌سازی و سرویس‌های حیاتی است. در نتیجه، بهره‌گیری از معماری‌های ذخیره‌سازی تغییرناپذیر، فناوری‌های سخت‌افزاری WORM، سامانه‌های تشخیص و پیشگیری از نفوذ، سیاست‌های امنیتی چندلایه و راهکارهای پیشگیرانه، نسبت به روش‌های صرفاً مبتنی بر بازیابی، اثربخشی بیشتری در کاهش پیامدهای حملات آینده خواهد داشت [۴۰].

نتیجه‌گیری

این مقاله مروری، نخستین چارچوب فنی جامع را برای تحلیل بدافزارهای پاک‌ساز از منظر طراحی، پیاده‌سازی و تکامل ارائه داد. با بررسی بیش از ۲۰ نمونه واقعی و تحلیل دقیق چهار مورد شاخص شامون، نات پتیا، ویسپرگیت و اسیدرین، نشان دادیم که این بدافزارها از ابزارهای صرفاً تخریبی به سلاح‌های سایبری استراتژیک تبدیل شده‌اند که قادرند زیرساخت‌های حیاتی را برای ماه‌ها از کار بیندازند.

بدافزارهای پاک‌ساز دیگر یک تهدید نظری یا یک اتفاق نادر نیستند. از حمله شانون به صنعت نفت عربستان در ۲۰۱۲ تا حملات ویسپرگیت و اسیدرین به زیرساخت‌های اوکراین در ۲۰۲۲، این بدافزارها به ابزاری استاندارد در زرادخانه عملیات‌های سایبری تهاجمی دولت‌ها تبدیل شده‌اند. نقطه ضعف مشترک تمام این حملات، اتکای آن‌ها به قابلیت «تغییرپذیری» داده‌ها در لایه‌های پایین‌تر است – مهاجم می‌تواند داده را بازنویسی کند زیرا سیستم به طور پیش‌فرض اجازه بازنویسی می‌دهد.

توصیه نهایی این مقاله به تمام سازمان‌هایی که داده‌های حیاتی را مدیریت می‌کنند، ساده اما فوری است: امروز، تصور کنید که یک بدافزار مشابه نات پتیا در حال اجرا روی حیاتی‌ترین سرور شما است. آیا می‌توانید کسب‌وکار خود را بدون از دست دادن داده‌های اساسی بازیابی کنید؟ اگر پاسخ شما «نمی‌دانم» یا «خیر» است، حداکثر ۹۰ روز فرصت دارید تا معماری یکبار نوشت و پشتیبان‌گیری آفلاین را پیاده‌سازی کنید. منتظر نباشید تا قربانی بعدی شما باشید.

در مجموع، نتایج این پژوهش نشان داد که بدافزارهای پاک‌ساز طی سال‌های اخیر از ابزارهایی با قابلیت تخریب محدود به تهدیداتی پیچیده برای زیرساخت‌های ذخیره‌سازی، محیط‌های ابری و سامانه‌های مجازی‌سازی تبدیل شده‌اند. همچنین، چارچوب پیشنهادی برای طبقه‌بندی روش‌های تخریب داده در سطوح فیزیکی، بلوکی و منطقی می‌تواند در تحلیل، مقایسه و طراحی راهکارهای دفاعی مؤثر مورد استفاده قرار گیرد. یافته‌های این پژوهش نشان می‌دهد که به‌کارگیری هم‌زمان فناوری‌هایی نظیر تغییرناپذیری داده ، رسانه‌های WORM، نسخه‌های پشتیبان برون‌خط و سامانه‌های پایش هوشمند، نسبت به راهکارهای صرفاً مبتنی بر بازیابی، تاب‌آوری بیشتری در برابر حملات تخریبی ایجاد می‌کند. با توجه به روند تکامل بدافزارهای پاک‌ساز، تحقیقات آینده می‌تواند بر توسعه سامانه‌های تشخیص مبتنی بر هوش مصنوعی، حفاظت از زیرساخت‌های ابری، فناوری‌های تغییرناپذیری داده، ذخیره‌سازی مبتنی بر WORM و تحلیل حملات علیه محیط‌های مجازی‌سازی و کانتینری متمرکز شود. همچنین ارزیابی عملکرد این راهکارها در زیرساخت‌های حیاتی و سامانه‌های توزیع‌شده می‌تواند به طراحی معماری‌های مقاوم‌تر در برابر حملات مخرب و افزایش تاب‌آوری سایبری منجر شود.

مشارکت نویسندگان

نویسنده اول مسئول انجام پژوهش، جمع‌آوری و تحلیل منابع، نگارش اولیه مقاله و اعمال اصلاحات بوده است. نویسنده دوم بر اجرای پژوهش نظارت داشته، در بازبینی علمی، اصلاح محتوا و تأیید نسخه نهایی مقاله مشارکت داشته است

تشکر و قدردانی

این پژوهش بدون دریافت هرگونه حمایت مالی از سازمان، مؤسسه یا نهاد خاصی انجام شده است. نویسندگان از داوران محترم و سردبیر مجله به‌دلیل ارائه دیدگاه‌ها و پیشنهادهای سازنده که موجب ارتقای کیفیت مقاله شد، قدردانی می‌کنند.

تعارض منافع

هیچگونه تعارض منافع توسط نویسندگان بیان نشده است.

منابع:

[1] A. Greenberg, Sandworm: A New Era of Cyberwar and the Hunt for the Kremlin's Most Dangerous Hackers. New York , NY, USA: Doubleday, 2019

[2] A. Cherepanov, “Bad Rabbit: a closer look at the new version of NotPetya,” ESET, WeLiveSecurity, White Paper, Oct. 2017. [Online]. Available: https://www.welivesecurity.com/wp-content/uploads/2017/10/bad-rabbit.pdf

[3] Microsoft Threat Intelligence Center (MSTIC), “New destructive malware WhisperGate targeting organizations inUkraine,”MicrosoftSecurityBlog,Jan.2022.[Online]. Available:

https://www.microsoft.com/security/blog/2022/01/15/new-destructive-malware-whispergate-targeting-organizations-in-ukraine

[4] L. O. Murchu, “AcidRain: A wiper targeting modems and routers in Ukraine,” Securelist (Kaspersky), Mar. 2022. [Online]. Available: https://securelist.com/acidrain-wiper-targeting-modems-routers-ukraine/111183/

[5] S. L. Garfinkel, “Digital forensics research: The next 10 years,” Digit. Investig., vol. 7, pp. S64–S73, Aug. 2010.

[6] M. Wei, L. M. Grupp, F. E. Spada, and S. Swanson, “Reliably erasing data from flash-based solid state drives,” in Proc. 9th USENIX Conf. File Storage Technol. (FAST), San Jose, CA, USA, 2011, pp. 8–21.

[7] Department of Defense (DoD), “National Industrial Security Program Operating Manual (NISPOM),” DoD 5220.22-M, Feb. 2006.

[8] J. Hellings, M. Sadoghi, and G. Alonso, “MVCC-based database corruption attacks and detection mechanisms,” Proc. VLDB Endowment, vol. 14, no. 5, pp. 789-801, Jan. 2021.

[9] P. M. Chen and B. D. Noble, “When virtual is better than real: Operating system support for virtual machines,” in Proc. 8th USENIX Symp. Oper. Syst. Des. Implement. (OSDI), San Diego, CA, USA, 2008, pp. 133–148.

[10] D. Kushwaha et al., “Lateral Movement Detection Using User Behavioral Analysis,” arXiv, 2022 .

[11] M. F. U. R. A. Malik, “Detection latency of reverse sequential vs. forward sequential I/O patterns in enterprise storage systems,” in Proc. IEEE Int. Conf. Big Data (BigData), Seattle, WA, USA, 2022, pp. 4532-4540.

[12] M. Graczyk, “Wiper malware: Evolution from Shamoon to WhisperGate,” SANS Institute InfoSec Reading Room, Tech. Rep. SANS-2023-WIPER, Feb. 2023.

[13] R. Falcone, “Shamoon 2: Return of the wiper – technical analysis,” Palo Alto Networks Unit 42, Threat Brief, Nov. 2016. [Online]. Available: https://unit42.paloaltonetworks.com/shamoon-2-return-wiper/

[14] Microsoft Security Response Center (MSRC), “CVE-2016-3309: Windows Task Scheduler elevation of privilege vulnerability,” Microsoft, Security Advisory, Aug. 2016.

[15] A. Greenberg, “The untold story of NotPetya, the most devastating cyberattack in history,” Wired Magazine, Aug. 2018. [Online]. Available: https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/

[16] E. Kovacs, “WhisperGate wiper malware analyzed: Over 50 Ukrainian government systems hit,” SecurityWeek, Jan. 17, 2022. [Online]. Available: https://www.securityweek.com/whispergate-wiper-malware-analyzed-over-50-ukrainian-government-systems-hit

[17] J. Menn, “How a cyberattack knocked out thousands of wind turbines in Germany,” Reuters, Apr. 4, 2022. [Online]. Available: https://www.reuters.com/business/energy/how-cyberattack-knocked-out-thousands-wind-turbines-germany-2022-04-04/

[18] A. Bessani, R. Mendes, and T. Oliveira, “Cloud storage integrity checking: A survey,” ACM Comput. Surv., vol. 52, no. 2, pp. 1-35, Apr. 2019..

[20] Amazon Web Services, “S3 Versioning documentation,” AWS Docs, 2023. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html

[21] M. Graczyk, “Wiper Malware: Evolution from Shamoon to WhisperGate,” SANS Institute, 2023.

[22] R. S. “Hardware vs. software immutability for ransomware and wiper protection,” IEEE Secur. Priv., vol. 20, no. 4, pp. 56-64, Aug. 2022.

[22] LTO Consortium, “LTO-9 specification: WORM cartridge features,” LTO Technology Provider Guidelines, Rev. 2.0, 2022.

[23] S. R. “Anomaly detection in block-level I/O streams for ransomware and wiper identification,” ACM Trans. Storage, vol. 18, no. 2, pp. 1-28, May 2022.

[24] J. P. “Future directions in offensive cyber operations: Wiper malware 2030,” J. Inf. Warfare, vol. 22, no. 3, pp. 1-18, Summer 2023.

[25] Trusted Computing Group, TPM 2.0 Library Specification, Family “2.0”, Level 00 Revision 01.59, Nov. 2019.

[26] J. Kim and D. Park, “Wiper Malware: An Analysis of Recent Trends and Preventive Measures,” Journal of Defense and Security, vol. 4, no. 2, pp. 35–59, 2022, doi: 10.23425/defsec.2022.4.2.35.

[27] M. Hirano and R. Kobayashi, “Machine Learning-based Ransomware Detection Using Low-level Memory Access Patterns Obtained From Live-forensic Hypervisor,” arXiv:2205.13765, 2022.

[28] O. Y. Matsko, I. Y. Havryliuk, and H. H. Nayman, “Technical Analysis of the CaddyWiper Malware,” Modern Information Protection, no. 1, pp. 6–15, 2023, doi:10.31673/2409-7292.2023.010006.

[29] A. Wolsey, “The State-of-the-Art in AI-Based Malware Detection Techniques: A Review,” arXiv:2210.11239, 2022.

[30] X. Ling et al., “Adversarial Attacks against Windows PE Malware Detection: A Survey of the State-of-the-Art,” arXiv:2112.12310, 2021.

[31] Trusted Computing Group, “TPM 2.0 Library Specification,” Rev. 1.59, 2021.

[32] ESET Research, “One Year of Wiper Attacks in Ukraine,” White Paper, 2023.

[33] Fortinet FortiGuard Labs, “Global Threat Landscape Report 2023,” Fortinet, 2023.

[34] IBM Security X-Force, “CaddyWiper: Third Wiper Malware Targeting Ukrainian Organizations,” IBM, 2022.

[35] M.M.Shirmohmmadi and H.Yasinian and A.Gholami and F.Bahrami “Dealing with Ambiguity in

Tech Projects: A Review of Tolerance for Ambiguity and its Development in IT Professionals" Arman Process Journal (APJ), vol.5, no.2,pp.1-15, Summer 2024.

[36] MITRE ATT&CK, "AcidRain (S1125)," MITRE ATT&CK, 2024.

[37] M.M.Shirmohmmadi and V.Mohmmadi ," Analysis of Failed DNS Responses Using Neural Network in Botnet Detection"Arman Process Journal (APJ), vol.6, no.4,pp.1-11, Winter 2025.

[38] M. Azadi, "Security Threats, Challenges, Procedures and Policies in Information Systems," Arman Process Journal (APJ), vol. 4, no. 3, pp. 34–42, Autumn 2023.

[39] H. Mansouri, "Approaches to Address the Challenge of Intrusion in Cloud Computing Services," Arman Process Journal (APJ), vol. 6, no. 3, pp. 76–88, Autumn 2025.

[40] R. Ghaffari, "Improving Security and Privacy in the Internet of Things Based on Blockchain Technology," Arman Process Journal (APJ), vol. 5, no. 4, pp. 15–24, Winter 2025.