Strategic Report: MUDDYWATER DLL Tradecraft and the Cyber Crucible OS Independence Innovation

Introduction: The Erosion of the Secure Perimeter

In the contemporary cybersecurity ecosystem, the concept of a hardened network perimeter has been entirely dismantled. Advanced Persistent Threats (APTs) no longer seek to batter down firewalls; instead, they compromise endpoints through social engineering, weaponize legitimate administrative tools, and operate silently within the volatile memory of trusted processes. For the financial sector, where the integrity of transactions and the confidentiality of client data are paramount, this shift in adversarial tradecraft represents an existential threat. Traditional security paradigms-heavily reliant on static file analysis, application whitelisting, and reactive, cloud-based telemetry-are experiencing catastrophic, systemic failures against these memory-centric attack vectors.1

This exhaustive research report investigates the recent tactical evolution of MUDDYWATER, a premier cyber espionage apparatus affiliated with the Iranian Ministry of Intelligence and Security (MOIS).2 During the first quarter of 2026, MUDDYWATER orchestrated a sprawling intrusion campaign-tracked globally as Operation Olalampo-that aggressively targeted the financial, diplomatic, and telecommunications sectors across North Africa and the broader Middle East and North Africa (MENA) region.4 This campaign demonstrated a devastating mastery of Dynamic Link Library (DLL) side-loading, memory manipulation, and advanced credential theft specifically designed to subvert Endpoint Detection and Response (EDR) solutions.4

However, analyzing the anatomy of the threat is only half the battle. As adversaries increasingly manipulate the underlying Windows operating system libraries that traditional security tools depend upon, a radical paradigm shift in defensive architecture is mandatory. To effectively neutralize machine-speed, in-memory threats, defense systems must decouple from the fragile operating systems they protect. This report details the necessity of Operating System (OS) independence and real-time DLL live inspection, culminating in a comprehensive ten-part executive presentation detailing how Cyber Crucible's autonomous, kernel-level capabilities provide an unparalleled "unfair advantage" against modern threat actors.1

Part I: The MUDDYWATER Threat Actor and Operation Olalampo

MUDDYWATER, operating under aliases such as Seedworm, TEMP.Zagros, Static Kitten, and Earth Vetala, has historically functioned as a highly active, albeit sometimes noisy, state-sponsored espionage group.3 However, recent telemetry indicates a profound maturation in their operational discipline and technical sophistication.10 The group has transitioned from utilizing off-the-shelf scripts to engineering bespoke, memory-safe

malware, indicating a significant investment in long-term, stealthy network persistence.

The North African and MENA Campaign

In early 2026, MUDDYWATER launched Operation Olalampo, a highly coordinated offensive targeting over 100 government, telecommunications, and financial entities across North Africa and the Middle East.5 This campaign utilized compromised email accounts, routed through anonymizing VPNs, to distribute sophisticated spear-phishing lures that appeared as legitimate diplomatic or financial correspondence.5

The primary objective of these intrusions was not immediate financial extortion via ransomware, but rather deep network espionage, the prepositioning of persistent access, and the systemic harvesting of highly privileged credentials.2 To achieve this, the threat actors deployed a newly developed arsenal of malware families designed specifically to evade detection by operating entirely within system memory.

The Evolution of the Malicious Arsenal

The tooling deployed in Operation Olalampo marks a stark departure from previous MUDDYWATER campaigns. The integration of memory-safe programming languages and the utilization of trusted communication channels for Command and Control (C2) demonstrate a highly evolved evasion strategy.14

Malware Component

Primary Function

Technical Characteristics and Execution Strategy

GhostFetch

Initial Downloader

Functions as a first-stage, in-memory downloader. It retrieves and executes subsequent malware payloads directly within active RAM, explicitly designed to evade disk-based scanning technologies.7

HTTP_VIP

Windows-Native Downloader

A custom-built web application utilizing the Flask Python framework. It manages connections to compromised hosts and facilitates the delivery of legitimate Remote Monitoring and Management (RMM) tools like AnyDesk for persistent access.7

GhostBackDoor

Persistent Implant

Provides advanced, long-term post-exploitation command and control, utilizing sophisticated string decoding techniques to obfuscate its operations from memory

scanners.7

CHAR

Advanced Backdoor

Developed in Rust (a memory-safe language). It is controlled entirely via a Telegram bot API (e.g., "stager_51_bot"). Debug strings contain emojis and unconventional Unicode sequences, indicating Large Language Model (LLM) assisted development.7

The development of the CHAR backdoor is particularly alarming for financial sector defenders. By utilizing the Telegram API for C2 communications, MUDDYWATER operators successfully blend their malicious traffic with normal, encrypted web traffic, bypassing perimeter network detection systems.13 Furthermore, the evidence of AI-assisted code generation in CHAR's development implies that the threat actor can rapidly iterate and compile polymorphic variations of their tooling, rendering traditional signature-based Anti-Virus (AV) solutions entirely ineffective.14

Part II: The Mechanics of Subversion - DLL Side-Loading Tradecraft

The linchpin of MUDDYWATER's operational success in North Africa is the disciplined, systemic abuse of DLL side-loading.4 To understand the severity of this threat, one must understand how the Windows Operating System dynamically loads supporting code.

When a Windows executable (an .exe file) is launched, it rarely contains all the code necessary to function. Instead, it relies on shared libraries, known as Dynamic Link Libraries (DLLs), to perform routine tasks such as networking, cryptography, or rendering graphics. If the executable does not specify a hardcoded, absolute file path for these required DLLs, the Windows OS utilizes a predefined "Search Order" to find them. Crucially, the very first location Windows checks is the local directory from which the executable itself was launched.4

Adversaries exploit this mechanism by placing a maliciously crafted DLL in the same directory as a legitimate, highly trusted, and digitally signed executable. When the legitimate program is executed, the OS unwittingly loads the malicious library directly into the trusted process's memory space.

To visualize this, consider the concept of Application Whitelisting as a school security protocol.18 A "hall pass" allows a student to move freely through the corridors. Traditional security tools look at the executable and see a valid, digitally signed "hall pass" (e.g., a Microsoft or vendor-signed certificate). Because the core executable is legitimate, the security tool ignores the process. However, DLL side-loading allows an attacker to effectively hollow out the student carrying the hall pass and replace their intent with malicious directives. The security guard (the EDR) sees the valid pass, but fails to realize the entity holding it is now a hostile actor.18

Weaponizing the Supply Chain: Fortemedia and SentinelOne

In the North African Operation Olalampo intrusions, MUDDYWATER operators demonstrated an exceptional understanding of enterprise software ecosystems by specifically targeting two distinct legitimate binaries to facilitate their DLL side-loading chains. These attacks were orchestrated seamlessly without user interaction, driven entirely by a malicious Node.js script dropping components into temporary directories (node.exe).10

The first side-loading vector utilized fmapp.exe, a legitimate, digitally signed audio-driver utility developed by Fortemedia Inc..17 MUDDYWATER delivered this benign executable alongside a malicious library named fmapp.dll. Upon execution, the trusted Fortemedia application inadvertently loaded the hostile DLL into its memory space. The primary function of this compromised library was to establish a reverse SOCKS5 proxy tunnel, creating a stealthy, persistent connection back to an attacker-controlled C2 server (frequently hardcoded to IPs such as 165.227.82.147 or 157.20.182.49).17 This encrypted tunnel allowed the adversaries to route lateral movement commands and exfiltrate data directly through the compromised host, bypassing enterprise firewalls.

The second side-loading vector, deployed approximately fifteen minutes after the Fortemedia proxy was established, is far more concerning for enterprise security teams. The attackers executed sentinelmemoryscanner.exe, a legitimate, digitally signed component of the SentinelOne endpoint security suite.17 This highly privileged EDR binary was used to side-load a malicious library named sentinelagentcore.dll.4

The selection of a premier EDR component for side-loading is a calculated, devastating maneuver. Security platforms universally trust their own components, and operating systems are configured to allow these security binaries extensive, unrestricted access to the kernel, physical memory, and the entire file system.4 By hijacking an EDR binary, the attackers effectively turned the organization's watcher into a weapon. Behavioral heuristics that would normally flag a standard application for performing deep system memory enumeration remain silent, because an EDR memory scanner is inherently designed to read the memory spaces of other processes.4

The Execution Flow and Living Off the Land (LotL)

Following the successful execution of these malicious DLLs, MUDDYWATER extensively utilized "Living Off the Land" (LotL) techniques. Rather than dropping custom hacking tools onto the disk, they leveraged native Windows administrative utilities to conduct their operations.22

PowerShell was heavily leveraged to execute obfuscated scripts, perform deep network reconnaissance, capture system screenshots, and extract Security Account Manager (SAM) registry hives for privilege escalation.3 Because tools like PowerShell and WMI (Windows Management Instrumentation) are native to the OS and universally whitelisted by traditional security controls, their abuse allows attackers to operate in plain sight, blending malicious espionage with standard IT administrative tasks.1

Part III: Financial Sector Risk - ChromElevator and App-Bound Encryption

For the financial sector, the ultimate objective of these sophisticated initial access and evasion techniques is not mere disruption, but the systematic acquisition of highly privileged credentials, cryptographic keys, and active session tokens. In the Operation Olalampo campaign, both the fmapp.dll and sentinelagentcore.dll malicious libraries were embedded with a specialized, devastating post-exploitation tool known as ChromElevator.4

ChromElevator is an open-source utility specifically engineered to extract saved passwords, active session cookies, and payment card data from Chromium-based web browsers (such as Google Chrome, Microsoft Edge, Brave, and Vivaldi).4 Historically, credential theft from web browsers involved simple, noisy file reads of the local SQLite databases where this data was stored.

However, in an attempt to thwart this exact type of theft, Google introduced App-Bound Encryption (ABE) in Chrome version 127. ABE cryptographically ties the encryption of browser data to the specific digital identity of the browser application itself, theoretically preventing external malware or unauthorized scripts from decrypting the stored secrets.26

MUDDYWATER's implementation of ChromElevator directly and elegantly circumvents this advanced protection.27 The tool utilizes highly advanced memory manipulation techniques-specifically Reflective Process Hollowing and Direct Syscalls-to bypass App-Bound Encryption. By hollowing out a legitimate process and reflectively injecting malicious code, the malware perfectly mimics the application identity required by the Windows Cryptography API: Next Generation (CNG) to decrypt the browser data.11 Furthermore, by utilizing Direct Syscalls, ChromElevator bypasses the user-mode API hooks that EDR solutions use to monitor system behavior, interacting directly with the Windows kernel.27

The strategic implications of this capability for financial institutions are catastrophic. Modern financial sector security architectures rely almost exclusively on Identity and Access Management (IAM), Multi-Factor Authentication (MFA), and conditional access policies. However, by stealing active, authenticated session cookies via ChromElevator, MUDDYWATER operators can execute "Pass-the-Cookie" attacks.1

This technique allows the adversary to hijack live, authenticated web sessions, entirely bypassing MFA prompts and gaining direct, unimpeded access to corporate cloud environments, financial transaction portals, and internal communication platforms. In financial environments where users routinely store credentials in unprotected facilities or browsers, the deployment of tools like ChromElevator leads directly to rapid lateral movement and systemic, unrecoverable compromise.28 Alongside ChromElevator, MUDDYWATER has deployed a suite of credential harvesters, including CE-Notes (a Chromium data stealer that mimics ChromElevator), LP-Notes (a fake Windows Security dialog used for phishing), and Blub (a cross-browser data stealer targeting Firefox and Opera).11

Part IV: The Structural Failure of Legacy Security Architectures

The unprecedented success of MUDDYWATER's campaign in North Africa illuminates the deep, structural vulnerabilities inherent in traditional endpoint security models. Financial sector technology leadership must acknowledge that reliance on legacy architectures-optimized for file scanning and reactive cloud analytics-presents an unacceptable level of risk against modern, memory-centric tradecraft.

The Illusion of Application Whitelisting

Application Whitelisting (AWL) operates on the fundamental premise that if an executable file is not on a predefined, static "safe list," it is blocked from executing.1 This methodology provided robust defense against traditional malware that relied on dropping a recognizable, malicious file onto a hard drive. However, AWL is structurally blind to the fileless, in-memory tradecraft utilized by APTs today.1

When a threat actor delivers a malicious macro that utilizes PowerShell to download a payload, AWL sees nothing wrong. PowerShell is a universally trusted administrative tool. When that payload is subsequently injected into the memory space of a whitelisted web server, or when a legitimate utility like sentinelmemoryscanner.exe is tricked into loading a malicious DLL, no new, unapproved executable is ever written to the disk.1

In essence, traditional Application Whitelisting creates a safe list that doubles as an attacker's target list.1 By focusing exclusively on the static attributes of a file on disk, AWL technologies allow hostile code, hidden within the RAM of approved processes, to execute with total impunity.

The Latency of Cloud-Based "Cloud Collectors"

The modern Endpoint Detection and Response (EDR) market is saturated with commoditized "cloud collectors"-vendors that operate as rudimentary sensors, shipping Microsoft-provided user-space telemetry to remote cloud servers for probabilistic analysis.1 While these cloud analytics platforms provide excellent value for historical forensic investigations and post-breach threat hunting, they are fundamentally inadequate for real-time threat prevention.

The latency introduced by this architecture is fatal. An event must occur, data must be packaged, transmitted across the internet to a cloud server, processed by an algorithm, and a response command must be sent back down to the endpoint. In the context of automated, machine-speed "smash and grab" attacks, this round-trip latency takes seconds to minutes.1 Adversaries exploiting process injection and DLL side-loading execute their critical actions-such as the theft of an MFA session cookie or the encryption of a critical database-in milliseconds. By the time a cloud-based EDR registers the alert and issues a kill command, the data has already been exfiltrated.1

To illustrate this, consider the "Security Camera versus Armed Guard" analogy.8 Legacy EDR acts like a security camera; it records the disaster in high definition so that human analysts can investigate the crime scene the next day. It watches, but it cannot

act in time. Proactive, kernel-level defense must act as an armed guard, capable of intercepting and tackling the thief the exact millisecond they touch the vault door.8

Part V: The Dependency Trap and Weaponized System Instability

Perhaps the most profound vulnerability of legacy EDR and XDR solutions is their absolute dependency on the very operating system they are designed to protect. Almost every modern security tool relies heavily on the core Windows OS APIs and System32 DLLs to function.1

This shared architecture creates a critical "shared fate" vulnerability.31 When sophisticated adversaries like MUDDYWATER compromise core Windows APIs, or manipulate System32 libraries directly in memory, the security tools relying on those exact same APIs are simultaneously blinded, subverted, or crashed.31

The Rise of EDR Killer Attacks

Attackers have recognized this architectural flaw and actively weaponize system stability. In modern intrusion campaigns, we observe "EDR Killer" system disruption attacks. These attacks deliberately target the Windows subsystems that security tools monitor. Legacy tools, attempting to function through compromised or heavily overloaded APIs, initiate aggressive operational loops that can result in catastrophic system-wide performance degradation. The symptoms include:

System-wide performance drops exceeding 20%.31

Disk I/O latencies increasing by a factor of 4x.31

Parallel processing capabilities being reduced to linear, sequential operations, causing critical system lag.31

This is not a byproduct of the attack; it is the goal. The adversaries intentionally cause client instability through a combination of OS bugs, kernel-level zero-days, and resource exhaustion. The objective is to force the IT administrator to manually disable the EDR software in a desperate attempt to restore business operations.1 The moment the defense is lowered to "fix the glitch," the adversary executes their primary payload.

For a security product to be effective in this hostile environment, its job must be to provide full, automated preventative capability without increasing additional load on the system during these catastrophic disruption events. It must survive when the OS itself is failing.

Part VI: Executive Briefing: The Path to OS Independence (A 10-Part Architectural Presentation)

To defeat the sophisticated in-memory manipulation and DLL tradecraft exhibited by actors like MUDDYWATER, defense architectures must abandon reactive, OS-reliant models. Cyber Crucible has engineered a proprietary, patented architecture that decouples its

operations from the underlying Windows OS libraries.1

The following 10-part executive presentation details the architectural challenges presented by modern adversaries and the subsequent engineering innovations that resulted in Cyber Crucible's true OS independence.

Module 1: The Core Challenge - The Dependency Trap

The fundamental flaw in modern cybersecurity is structural. Almost every security tool relies on the exact same Windows OS DLLs it aims to protect.31 This creates a shared fate: when the OS is blind, buggy, or under attack, the security tool is blinded and impacts system performance. Cyber Crucible recognized that to protect the host, the defense mechanism must observe the host from a higher plane of privilege, rather than operating as a vulnerable peer within the user-space.

Module 2: Weaponizing System Stability (The EDR Killer)

Client instability can result from a combination of OS bugs, kernel-level zero-days, and intentional EDR killer system disruption attacks.31 Attackers intentionally manipulate security tool listener injections and in-memory capabilities to degrade system performance.31 Cyber Crucible's foundational engineering mandate is to provide full, autonomous preventative capability without increasing additional load on the system during these high-stress events [User Query]. The defense must remain lightweight and invisible, even when the OS is actively failing.

Module 3: The Philosophy of OS Independence

To break the Dependency Trap, Cyber Crucible embarked on a multi-year engineering initiative to sever reliance on Microsoft-provided data buses and APIs. By investing heavily in proprietary, kernel-level sensor development, Cyber Crucible achieved an engineering moat: the system is almost completely independent of the OS.1 This ensures that even if an attacker successfully compromises a System32 DLL using the tradecraft seen in Operation Olalampo, the Cyber Crucible analytics engine remains fully operational, untampered, and highly lethal to the adversary.

Module 4: Real-Time DLL Live Inspection

To counter the precise DLL side-loading techniques utilized by MUDDYWATER (e.g., sentinelmemoryscanner.exe loading sentinelagentcore.dll), Cyber Crucible implemented continuous, real-time DLL live inspection.1 It is insufficient to merely verify a program's signature at execution. Cyber Crucible continuously monitors the active RAM to conduct memory integrity checks on all loaded supporting libraries.1 By tracking parent-child relationships and utilizing Control-Flow Integrity (CFI), the engine detects the exact millisecond a trusted program's execution flow is hijacked by an untrusted, side-loaded module.1

Module 5: Engineering the Unfair Advantage (Version 4.5.1.0)

Released in November 2025, Cyber Crucible version 4.5.1.0 marked a significant milestone in OS decoupling. Recognizing that Windows SSL and HTTP libraries are frequent targets for exploitation and manipulation, Cyber Crucible severed its reliance on them. The platform transitioned to utilizing openSSL for all SSL encryption/decryption processes, and implemented libCurl to entirely bypass the native Windows HTTP libraries [User Query]. This ensures that critical telemetry and C2 communications remain secure and functional even if the host's networking stack is compromised.

Module 6: Precision Kernel Cryptography (Version 4.5.1.2)

By January 2026 (Version 4.5.1.2), Cyber Crucible had successfully deployed custom, kernel-level cryptography libraries developed directly from published Microsoft RFCs on Authenticode [User Query]. Initially, native Microsoft libraries were utilized simultaneously to double-check mathematical calculations and transmit deviations to CC servers for error checking. After 10 months of zero observed calculation errors, it was determined that the native Microsoft libraries were actually causing unacceptable system instability under heavy load. Consequently, the reliance on Microsoft backup-calculations was entirely dropped, relying solely on CC's optimized kernel cryptography [User Query].

Module 7: Severing the Metadata Umbilical (Versions 4.5.2.1 & 4.5.2.2)

In February 2026, subsequent updates dramatically advanced the OS independence initiative. Version 4.5.2.1 successfully removed dependency on multiple data bus and metadata production kernel APIs [User Query]. Shortly thereafter, Version 4.5.2.2 removed additional dependencies on tertiary Windows APIs. At this stage, the Cyber Crucible system achieved near-total independence from the underlying operating system, retaining only a few highly specific OS dependencies necessary for basic hardware interaction [User Query].

Module 8: The Cyber Crucible Core Evolution (Version 4.5.3.0)

April 2026 saw the release of the Cyber Crucible Core (Version 4.5.3.0) [User Query]. This update introduced configurable, asynchronous DLL analytics designed specifically for machines with minimal hardware resources, ensuring robust defense without performance degradation. Furthermore, process creation and process injection analytics and telemetry were upgraded to utilize disk persistence [User Query]. Crucially, this version entirely removed dependencies on Windows MiniFilters for these analytics, eliminating a major vector for kernel-level interference [User Query].

Module 9: Kernel Buffer Operation Enhancements (Version 4.5.3.2)

The Cyber Crucible kernel driver leverages three primary buffers to temporarily store telemetry before it is transmitted to the customer dashboard (distinct from local behavioral model operations). If a kernel buffer overloads due to a massive OS bug or a concentrated attack on the system, CPU load can spike. To mitigate this, Version 4.5.3.2

introduced critical enhancements: all three primary buffers' usage percentages are now continuously transmitted to CC infrastructure for metrics gathering [User Query]. Administrators can remotely adjust the size of these buffers, and an automated customer notification system for very high buffer usage will be fully available by June 1, 2026 [User Query]. This guarantees CC can manage system load dynamically during EDR killer attacks.

Module 10: The Final Frontier - Network Stack Independence Roadmap

The final, remaining vulnerability involves the ws2_32.dll (Winsocket) dependency. MUDDYWATER and similar actors have been observed launching targeted attacks designed to block telemetry from being transmitted by Network Interface Cards (NICs), effectively shutting down cloud-dependent MDR visibility [User Query]. The short-term mitigation provided by Cyber Crucible is a "Partially offline" notification alert [User Query]. However, the long-term, definitive mitigation is the total removal of the ws2_32.dll dependency, scheduled for release by August 2026 [User Query]. This will render Cyber Crucible completely immune to network stack disruption attacks.

Part VII: Autonomous Kernel-Level Prevention in Action

The culmination of OS independence, real-time DLL live inspection, and advanced memory analytics is a defense platform that does not react, but prevents. Cyber Crucible replaces the latency of cloud-based analysis with a patented, edge-computing analytics engine powered by Genetic AI.1

Unlike Large Language Models (LLMs) or probabilistic AI that analyze text patterns to guess an outcome, Cyber Crucible's Genetic AI uses massive, deeply technical variable sets to produce definitive, deterministic outcomes.8 Because these models are translated into raw machine language and execute locally within the kernel of the endpoint, they operate with absolute autonomy and zero network latency.8

When a malicious behavior is initiated-such as a side-loaded DLL attempting to utilize Direct Syscalls to bypass App-Bound Encryption and steal Chrome session tokens-the Genetic AI evaluates the intent of the data access in a fraction of a second.1 If malicious intent is detected, the engine autonomously intercepts and suspends the offending process in under 200 milliseconds.1

This immediate, localized suspension freezes the attacker and the compromised process in memory, preventing data exfiltration or encryption while seamlessly preserving the memory state for subsequent forensic root-cause analysis.8 Operations continue without downtime, disruption, or the need for a system reboot.8

Capability Metric

Legacy EDR/XDR

Cyber Crucible

Operating Architecture

User-space telemetry, Highly OS-dependent

Kernel-level sensors, OS-independent 1

Decision Location

Cloud-based analytics (High Latency)

Edge-native, on-endpoint processing 1

Response Time

Minutes to Hours (Network + Human delay)

< 200 Milliseconds (Fully Autonomous) 8

Detection Methodology

Signatures, basic heuristics, probabilistic AI

Deterministic Genetic AI, Behavioral Intent 1

DLL / Memory Defense

Blind to advanced in-memory library manipulation

Continuous DLL live inspection, memory integrity validation 1

System Load Management

Crashes under EDR-killer attacks

Dynamic kernel buffer scaling, OS decoupled [User Query]

Conclusion: Strategic Imperatives for the Financial Sector

The Q1 2026 MUDDYWATER Operation Olalampo campaign across North Africa and the MENA region serves as a stark, undeniable leading indicator of the future of cyber warfare. By mastering DLL side-loading, in-memory process hollowing, and the weaponization of digitally signed EDR binaries, state-sponsored adversaries have effectively bypassed the foundational principles of traditional endpoint security. The extraction of privileged session tokens via tools like ChromElevator, combined with AI-assisted malware development, creates a highly lethal environment where legacy, cloud-reliant detection platforms are simply outpaced and outmaneuvered.

For the financial sector, the implications are profound. An architecture that relies on the integrity of the operating system it is trying to protect is structurally doomed to fail. The defense paradigm must definitively shift from reactive resilience and cloud-based telemetry collection to proactive, autonomous, and OS-independent defense.

By decoupling from vulnerable Windows dependencies and implementing continuous, real-time live inspection of running programs and their supporting DLLs, next-generation

platforms like Cyber Crucible strip the adversary of their primary advantage: stealth in memory. Coupled with kernel-level, deterministic Genetic AI capable of neutralizing threats in under 200 milliseconds without relying on human intervention, financial institutions can construct an unyielding defense. In a threat landscape where cryptographic trust is constantly exploited, survival depends entirely on an architecture that verifies continuously, decides autonomously, and prevents instantly.

Works cited

24FEB2026_whitepapers_combined.pdf

Muddying the Tracks: The State-Sponsored Shadow Behind Chaos Ransomware - Rapid7, accessed June 1, 2026, https://www.rapid7.com/blog/post/tr-muddying-tracks-state-sponsored-shadow-behind-chaos-ransomware/

Iranian Government-Sponsored Actors Conduct Cyber Operations Against Global Government and Commercial Networks | CISA, accessed June 1, 2026, https://www.cisa.gov/news-events/cybersecurity-advisories/aa22-055a

MuddyWater Espionage Campaign Hits 9 Countries in Q1 2026 - Safestate, accessed June 1, 2026, https://www.safestate.com/post/muddywater-espionage-campaign-hits-9-countries-in-q1-2026

Cyber News Roundup - October 24 2025 - Integrity360, accessed June 1, 2026, https://www.integrity360.com/cyber-news-roundup-october-24-2025

Unmasking MuddyWater's New Malware Toolkit Driving International Espionage - Group-IB, accessed June 1, 2026, https://www.group-ib.com/blog/muddywater-espionage/

Operation Olalampo: Inside MuddyWater's Latest Campaign | Group-IB Blog, accessed June 1, 2026, https://www.group-ib.com/blog/muddywater-operation-olalampo/

Cyber Crucible Defense Dark Theme v9.pdf

MuddyWater - Cyber Operations Tracker, accessed June 1, 2026, https://www.cfr.org/cyber-operations/muddywater

Symantec uncovers Iran-linked Seedworm espionage campaign targeting airport, government, manufacturing sectors - Industrial Cyber, accessed June 1, 2026, https://industrialcyber.co/threats-attacks/symantec-uncovers-iran-linked-seedworm-espionage-campaign-targetin

Iranian hackers deploy new MuddyViper spyware against Israel - CyberInsider, accessed June 1, 2026, https://cyberinsider.com/iranian-hackers-deploy-new-muddyviper-spyware-against-israel/

MuddyWater, Earth Vetala, MERCURY, Static Kitten, Seedworm, TEMP.Zagros, Mango Sandstorm, TA450, MuddyKrill, Group G0069 | MITRE ATT&CK, accessed June 1, 2026, https://attack.mitre.org/groups/G0069/

Operation Olalampo: MuddyWater's Expanding Campaign Across MENA | Hive Pro, accessed

June 1, 2026,
https://hivepro.com/threat-advisory/operation-olalampo-muddywater-expanding-campaign-across-mena/

The Iranian Cyber Threat Landscape: Inside the Adversary Playbook - ExtraHop, accessed June 1, 2026,
https://cloud-assets.extrahop.com/resources/whitepapers/iranian-cyber-threat-landscape-inside-adversary-playbo

Hive Pro Brings Frontline Iranian Cyber Threat Intelligence to RS - The National Law Review, accessed June 1, 2026,
https://natlawreview.com/press-releases/hive-pro-brings-frontline-iranian-cyber-threat-intelligence-rsa-conference

Pre-Positioned Access: The Cyber Threat Behind the Iran Conflict - Centripetal, accessed June 1, 2026,
https://www.centripetal.ai/threat-research/pre-positioned-access-cyber-threat-iran-conflict

Seedworm: Iran-Linked Hackers Breached Korean Electronics Maker in Global Spying Campaign | SECURITY.COM, accessed June 1, 2026,
https://www.security.com/threat-intelligence/iran-seedworm-electronics

11MAR2026_confluence_export.pdf

knowledgebase_export_12MAY2026.pdf

Clearing the Water: Unmasking an Attack Chain of MuddyWater - Huntress, accessed June 1, 2026, https://www.huntress.com/blog/muddywater-attack-chain

Inside a MuddyWater Intrusion: Exploitation of SharePoint and Living-Off-the-Land Tactics, accessed June 1, 2026,
https://kudelskisecurity.com/research/inside-a-muddywater-intrusion-exploitation-of-sharepoint-and-living-off-the-

Threat actor leverages coin miner techniques to stay under the radar - here's how to spot them | Microsoft Security Blog, accessed June 1, 2026,
https://www.microsoft.com/en-us/security/blog/2020/11/30/threat-actor-leverages-coin-miner-techniques-to-stay-u

Threat Intelligence - Blumira, accessed June 1, 2026,
https://www.blumira.com/threat-intelligence-updates

Cyberaanval Nederland en Belgi | Dagelijks overzicht - Cybercrimeinfo, accessed June 1, 2026, https://www.ccinfo.nl/menu-nieuws-trends/actuele-cyberaanvallen

TCLBanker: Trojanized Logitech Installer Fuels Banking Malware Campaign | Hive Pro, accessed June 1, 2026,
https://hivepro.com/threat-advisory/tclbanker-trojanized-logitech-installer-fuels-banking-malware-campaign/

MuddyWater: Snakes by the riverbank - WeLiveSecurity, accessed June 1, 2026,
https://www.welivesecurity.com/en/eset-research/muddywater-snakes-riverbank/

17 Critical Alerts Covering APT AI Adoption, Pre-Auth Zero-Days, and Supply Chain Worms | SISA Weekly Threat Watch, accessed June 1, 2026,
https://www.sisainfosec.com/weekly-threat-watch/17-critical-alerts-covering-apt-ai-adoption-pre-auth-zero-days-a

2025 CBEST thematic | Bank of England, accessed June 1, 2026, https://www.bankofengland.co.uk/financial-stability/operational-resilience-of-the-financial-sector/2025-cbest-them

Contents - arXiv, accessed June 1, 2026, https://arxiv.org/html/2506.22323v1

Cyber Crucible Fintech Use Case.pdf

OS Independence Presentation-1.pdf

Cyber Crucible Overview 2025.2.pptx-4.pdf