ROUTERS: Jurnal Sistem dan Teknologi Informasi
Vol. 1, No. 1, Juli 2022, pp. xx-xx
ISSN: xxxx-xxxx, DOI: 10.xxxxx/routers.v0i0.xxxx

EVALUASI KEMAMPUAN DETEKSI SIEM TERHADAP TEKNIK SERANGAN *LIVING OFF THE LAND* BERBASIS POWERSHELL BERDASARKAN MITRE ATT&CK *FRAMEWORK*

Alfarizi Pratama Sunandar [1], Nelmiawati [2]
[1] Teknik Informatika, Politeknik Negeri Batam
[2] Teknik Informatika, Politeknik Negeri Batam

INFORMASI ARTIKEL

Diterima 28 Juli 2021
Direvisi 28 Agustus 2021
Diterbitkan 28 September 2021

Kata kunci:

SIEM;
Living off the Land;
PowerShell;
MITRE ATT&CK;
Wazuh..

ABSTRAK

Serangan siber berbasis teknik *Living off the Land* (LOTL) semakin marak digunakan karena memanfaatkan *tools* dan proses bawaan sistem operasi yang sah, sehingga sulit dibedakan dari aktivitas normal oleh pertahanan berbasis *signature*. PowerShell menjadi vektor LOTL yang paling sering disalahgunakan karena terinstal secara *default* di hampir seluruh versi Windows modern dan mendukung eksekusi *fileless*. Penelitian ini bertujuan mengevaluasi kemampuan deteksi *default* Wazuh, sebuah *Security Information and Event Management* (SIEM) *open-source*, terhadap enam varian teknik PowerShell LOTL kategori *Execution* (T1059.001) yang mengacu pada *framework* MITRE ATT&CK. Simulasi serangan dilakukan menggunakan Atomic Red Team pada lingkungan lab virtual terisolasi, dengan status deteksi dinilai menggunakan skema kategori yang diadaptasi dari *MITRE ATT&CK Evaluations Methodology* (*None*, *Telemetry*, *General*, *Tactic*, *Technique*). Hasil analisis akar masalah (*root cause*) atas teknik yang lolos dari deteksi *default* menjadi dasar perancangan *custom detection rule* yang dipetakan ke *Technique ID* MITRE ATT&CK guna mengukur peningkatan *detection rate*. Selain itu, pengujian *false positive* dilakukan dengan mensimulasikan aktivitas operasional rutin serta skrip administratif yang sah untuk menguji presisi *custom rule* yang dikembangkan, guna memastikan aturan kustom tidak memicu *alert* palsu (*noise*) pada operasional normal. Hasil penelitian ini membuktikan bahwa implementasi *custom rule* berhasil menutup *detection gap* dan meningkatkan status deteksi ke tingkat *Technique* secara presisi tanpa memicu *false positive*. Penelitian ini diharapkan memberikan bukti empiris kemampuan deteksi SIEM *open-source* terhadap teknik LOTL, sekaligus menghasilkan luaran praktis berupa *custom rule* yang dapat langsung diadopsi oleh tim keamanan dengan sumber daya terbatas.

EVALUATION OF SIEM DETECTION CAPABILITY AGAINST POWERSHELL-BASED LIVING OFF THE LAND

ATTACK TECHNIQUES BASED ON THE MITRE ATT&CK FRAMEWORK

ARTICLE INFO

Received July 28, 2021
Revised August 28, 2021
Published September 28, 2021

Keyword:

SIEM;
Living off the Land;
PowerShell;
MITRE ATT&CK;
Wazuh..

ABSTRACT

Living off the Land (LOTL) technique-based cyberattacks are becoming increasingly prevalent because they exploit legitimate operating system built-in tools and processes, making them difficult to distinguish from normal activity by signature-based defenses. PowerShell is the most frequently abused LOTL vector because it is installed by default in almost all modern Windows versions and supports fileless execution. This study aims to evaluate the default detection capabilities of Wazuh, an open-source Security Information and Event Management (SIEM), against six PowerShell LOTL technique variants in the Execution category (T1059.001) referring to the MITRE ATT&CK framework. Attack simulations were conducted using Atomic Red Team in an isolated virtual lab environment, with detection status assessed using a category scheme adapted from the MITRE ATT&CK Evaluations Methodology (None, Telemetry, General, Tactic, Technique). The results of the root cause analysis of techniques that escaped default detection became the basis for designing custom detection rules mapped to the MITRE ATT&CK Technique ID to measure the increase in detection rate. In addition, false positive testing was conducted by simulating routine operational activities and legitimate administrative scripts to test the precision of the developed custom rules, ensuring that the custom rules do not trigger false alerts (noise) during normal operations. The results of this study demonstrate that the implementation of custom rules successfully closes the detection gap and increases the detection status to the Technique level with precision without triggering false positives. This study is expected to provide empirical evidence of the detection capabilities of open-source SIEMs against LOTL techniques, while also producing practical outputs in the form of custom rules that can be immediately adopted by security teams with limited resources.

Corresponding Author:

Coresponding Author Name, Affiliation
Email: mia@polibatam.ac.id

1. PENDAHULUAN

Serangan siber berbasis teknik Living off the Land (LOTL) menjadi ancaman signifikan terhadap infrastruktur digital. Badan keamanan siber Amerika Serikat mencatat kelompok penyerang canggih seperti Volt Typhoon secara konsisten memanfaatkan tools bawaan sistem operasi agar berbaur dengan aktivitas normal dan meminimalkan jejak pada log [1][2]. Pada 2024, sekitar 84% serangan berdampak tinggi memanfaatkan tools sistem yang sah, bukan malware custom [3]. Windows dan PowerShell menjadi target utama karena terinstal default, berhak akses

tinggi, dan mendukung eksekusi fileless yang sulit dipindai antivirus konvensional [4][5]. Akibatnya, banyak insiden baru terdeteksi setelah dampak signifikan terjadi median dwell time bahkan meningkat menjadi 14 hari pada 2025, dari 11 hari tahun sebelumnya [6]. Menandakan kapabilitas deteksi masih tertinggal dari kecepatan serangan.

Security Information and Event Management (SIEM) menjadi lini pertahanan utama untuk memonitor dan mengorelasikan aktivitas mencurigakan, namun solusi komersial umumnya berbiaya lisensi tinggi dan berisiko vendor lock-in, menyulitkan adopsi oleh organisasi beranggaran terbatas [7]. Kondisi ini mendorong adopsi SIEM/XDR open source, dengan Wazuh sebagai salah satu yang paling banyak digunakan secara global, melindungi lebih dari 15 juta endpoint pada 100.000+ organisasi [8]. Penelitian ini memilih Wazuh karena menyediakan mesin deteksi berbasis rule dan anomali secara out-of-the-box, mendukung kustomisasi rule penuh, serta bersifat open-source sehingga mudah direplikasi tanpa hambatan biaya [9][7]. Kemampuan deteksi out-of-the-box tidak serta-merta mengenali seluruh varian LOTL, sehingga diperlukan perbaikan sistematis.

MITRE ATT&CK Framework menyediakan basis pengetahuan taktik-teknik musuh yang terstandardisasi dan telah diadopsi luas sebagai dasar threat-informed defense, termasuk untuk mengidentifikasi celah visibilitas data pada teknik prioritas [10]. Sebagai preseden, kombinasi analisis log dengan custom detection rule pada SIEM terbukti meningkatkan deteksi lateral movement dan kerberoasting sekaligus menurunkan false alarm dibanding pendekatan signature-based [11] menjadi dasar pendekatan penelitian ini: mengevaluasi rule default Wazuh, menganalisis root cause teknik yang lolos, lalu merancang custom rule yang dipetakan ke Technique ID MITRE ATT&CK. Efektivitas custom rule bergantung pada visibilitas log; secara default Windows hanya mencatat command line tanpa detail isi skrip PowerShell yang dieksekusi, sehingga PowerShell Script Block/Module Logging perlu diaktifkan dan dilengkapi Sysmon untuk memantau relasi parent-child proses sebagai indikator kuat aktivitas LOTL [2][12].

Berdasarkan pemaparan tersebut, penelitian ini berfokus pada evaluasi Wazuh terhadap teknik T1059.001 (PowerShell) kategori Execution. Pengujian direncanakan dilakukan melalui simulasi menggunakan Atomic Red Team pada lab virtual terisolasi, dengan status deteksi dinilai memakai skema MITRE ATT&CK Evaluations. Kontribusi yang diharapkan dari penelitian ini meliputi bukti empiris kemampuan deteksi default Wazuh terhadap varian PowerShell LOTL representatif, analisis root cause teknik yang lolos deteksi, dan custom detection rule terpetakan MITRE ATT&CK sebagai luaran praktis bagi tim keamanan bersumber daya terbatas, sekaligus mengisi kesenjangan riset evaluasi SIEM open-source terhadap teknik LOTL yang masih terbatas [4].

2. LANDASAN TEORI

2.1 *Living off the Land (LOTL)*

Living off the Land (LOTL) adalah teknik serangan siber di mana penyerang tidak menggunakan malware maupun tools asing, melainkan memanfaatkan aplikasi, script, atau binary yang sudah terpasang secara sah di dalam sistem operasi korban untuk menjalankan aktivitas berbahaya [4]. Karena proses yang dijalankan berasal dari komponen resmi sistem operasi, aktivitas LOTL cenderung berbaur dengan operasi sistem normal sehingga sulit dibedakan oleh tools keamanan berbasis signature [4]. Kavadias [5] menjelaskan bahwa LOTL memiliki keterkaitan erat dengan konsep fileless malware, di mana adversary meminimalkan jejak file baru pada disk dan mengeksekusi payload secara in-memory melalui tools bawaan sistem. Penelitian ini secara spesifik membatasi ruang lingkup kajian pada LOTL berbasis PowerShell di lingkungan Windows, mengacu pada kategori Execution (Tactic ID: TA0002) dan teknik Command and Scripting Interpreter: PowerShell (Technique ID: T1059.001) pada MITRE ATT&CK Framework.

2.2 MITRE ATT&CK *Framework*

MITRE ATT&CK adalah basis pengetahuan global mengenai taktik dan teknik adversary yang disusun berdasarkan observasi dunia nyata, pertama kali diinisiasi pada tahun 2015 [10].

Framework ini menyusun pengetahuan serangan ke dalam struktur berjenjang berupa Tactic, Technique, dan Sub-technique. Hubbard [10] menekankan bahwa nilai utama ATT&CK bagi tim defender bukan sekadar daftar indikator serangan, melainkan kemampuannya mengidentifikasi celah visibilitas data yang dibutuhkan untuk mendeteksi suatu teknik, menjadikan framework ini dasar yang tepat untuk mengevaluasi kesiapan visibilitas log sebelum menyusun rule deteksi.

2.3 MITRE ATT&CK *Evaluations Methodology*

MITRE ATT&CK Evaluations adalah program pengujian independen yang menilai kapabilitas deteksi produk keamanan terhadap simulasi serangan nyata berbasis TTP adversary [13]. Berbeda dari pengujian antivirus konvensional, program ini tidak memberi skor komparatif antar vendor, melainkan menyajikan data mentah hasil deteksi secara terbuka [14]. Pada skema klasik yang digunakan selama beberapa putaran evaluasi, setiap langkah serangan diklasifikasikan ke dalam lima kategori berjenjang: None (tidak ada data tercatat), Telemetry (tercatat tanpa alert), General (alert tanpa detail metode), Tactic (alasan teridentifikasi, metode belum spesifik), dan Technique (konteks paling lengkap, alasan dan metode spesifik sesuai Technique ID) [15]. Skema ini relevan dengan konsep analytic coverage vs telemetry coverage, dan turut dikuatkan oleh Outkin. [16] yang memanfaatkan kategori ini untuk memodelkan tingkat keahlian analis yang dibutuhkan pada tiap kategori. Penelitian ini mengadopsi skema kategorikal klasik tersebut karena lebih proporsional untuk skala pengujian akademik dengan jumlah varian terbatas dan lebih mudah diinterpretasikan secara kualitatif untuk keperluan root cause analysis.

2.4 PowerShell dan Mekanisme Logging

PowerShell adalah shell command-line sekaligus bahasa scripting yang terintegrasi secara default pada hampir seluruh versi Windows modern, dengan akses penuh terhadap .NET Framework, WMI, dan Registry [4]. Kombinasi hak akses tinggi, ketersediaan default, dan dukungan eksekusi in-memory menjadikan PowerShell salah satu vektor LOTL yang paling sering disalahgunakan [5]. Secara default, Windows hanya mencatat command line tanpa merekam isi skrip yang benar-benar dieksekusi, sehingga aktivitas berbahaya yang dienkode (*encoded command*) sering tidak dapat direkonstruksi pasca-insiden [2]. Untuk menutup celah tersebut, penelitian ini merencanakan aktivasi dua sumber log utama: PowerShell Script Block/Module Logging (Event ID 4104 dan 4103) untuk merekam konten penuh skrip yang dieksekusi, dan Sysmon (Event ID 1, 3, dan 11) untuk merekam pembuatan proses beserta relasi parent-child dan koneksi jaringan keluar sebagai indikator kuat pola eksekusi LOTL [12].

2.5 Atomic Red Team

Atomic Red Team adalah framework adversary emulation open-source yang dikembangkan oleh Red Canary sejak Oktober 2017 [17]. Setiap unit pengujian disebut atomic test, didefinisikan dalam format terstruktur (Markdown dan YAML) sehingga dapat dieksekusi maupun diotomasi melalui Invoke-AtomicRedTeam [18]. Karakteristik utamanya adalah pemetaan eksplisit dan resmi terhadap Technique ID MITRE ATT&CK [4], serta kredibilitasnya sebagai tool riset dikuatkan oleh studi perbandingan terstruktur yang menempatkannya bersama MITRE Caldera dan Metasploit pada peringkat teratas [19][20]. Atomic Red Team dipilih sebagai sarana simulasi karena sifatnya yang open-source, pemetaan langsung ke Technique ID T1059.001, serta desainnya yang non-destruktif dan dilengkapi mekanisme cleanup.

2.6 Security Information and Event Management (SIEM)

SIEM adalah sistem yang mengumpulkan, mengorelasikan, dan menganalisis log keamanan dari berbagai sumber guna mendeteksi serta memberi peringatan atas aktivitas mencurigakan. Elmastaş dan Eyüpoğlu [11] membuktikan bahwa SIEM yang dikombinasikan dengan teknik korelasi log dan custom rule mampu mengungguli pendekatan signature-based

konvensional. Penelitian ini secara spesifik menggunakan Wazuh, platform SIEM sekaligus XDR open-source yang mengintegrasikan pengumpulan log, deteksi ancaman berbasis rule maupun anomali, file integrity monitoring, dan analisis kepatuhan dalam satu arsitektur terpadu [9]. Wazuh dipilih karena bersifat open-source tanpa biaya lisensi [7], menyediakan ruleset out-of-the-box sekaligus mendukung kustomisasi rule penuh, dan merupakan salah satu platform SIEM/XDR open-source dengan basis pengguna terluas secara global [8].

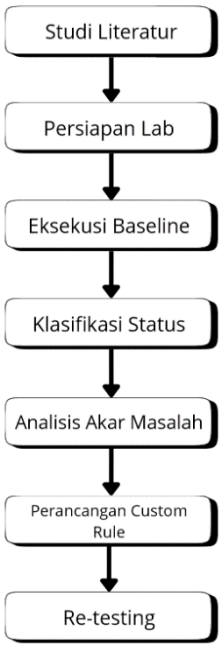
2.7 *Custom Detection Rule*

Custom detection rule merupakan aturan deteksi tambahan yang dirancang untuk menutup celah (gap) yang ditemukan pada rule default SIEM. Sebagai preseden empiris, Elmastaş dan Eyüpoğlu [11] menunjukkan bahwa custom rule yang disusun berdasarkan analisis korelasi log Windows Event Log mampu meningkatkan deteksi serangan lateral movement dan kerberoasting sekaligus menurunkan false alarm. Secara teknis pada platform Wazuh, custom rule direncanakan disusun dalam berkas local_rules.xml menggunakan struktur XML yang terdiri atas atribut id, level, dan elemen pencocokan pola seperti if_sid serta match/regex [21].

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental, di mana setiap varian teknik PowerShell *Living off the Land* (LOTL) yang disimulasikan menggunakan Atomic Red Team berperan sebagai treatment, sementara status deteksi pada dashboard Wazuh sebelum dan sesudah penerapan custom detection rule menjadi variabel hasil yang diukur. Pendekatan simulasi terkontrol semacam ini merupakan metodologi yang telah digunakan secara luas untuk mengevaluasi kapabilitas deteksi Security Information and Event Management (SIEM) secara empiris [22][15]. Alur dan tahapan pelaksanaan penelitian ini secara sistematis digambarkan pada Gambar 1 mengenai Tahapan Penelitian. Penjabaran dari setiap tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut:

Gambar 1. Tahapan Penelitian



Adapun penjabaran tahapan yang dilakukan adalah sebagai berikut :

1. Studi Literatur

   Peneliti melakukan kajian mendalam terhadap publikasi akademik, laporan industri, dan advisory resmi terkait teknik LOTL, penyalahgunaan PowerShell, framework MITRE ATT&CK, serta evaluasi terhadap SIEM berbasis open-source. Tahap ini krusial karena kajian literatur berperan penting dalam mengidentifikasi celah pengetahuan (research gap) yang menjadi dasar orisinalitas penelitian, sebagaimana direkomendasikan dalam studi sejenis sebelum merancang eksperimen mendalam [4].

2. Persiapan Lab

   Tahap ini meliputi instalasi Windows 11 sebagai target endpoint serta implementasi arsitektur komponen Wazuh manager dan Wazuh agent. Selanjutnya dilakukan aktivasi System Monitor (Sysmon) menggunakan templat konfigurasi modular yang teruji publik guna menyaring event log berkualitas tinggi, serta pengaktifan kebijakan PowerShell Script Block Logging dan Module Logging melalui Group Policy. Langkah konfigurasi ini mengacu pada rekomendasi mekanisme logging minimum untuk memperoleh visibilitas penuh terhadap aktivitas penyerangan siber berbasis LOTL[2].

3. Eksekusi Baseline (Rule Default)

   Setiap hasil eksekusi skenario *malicious* dicatat dan diklasifikasikan menggunakan skema kategori yang diadaptasi dari MITRE ATT&CK Evaluations Methodology: None, Telemetry, General, Tactic, dan Technique [15]. Secara paralel, setiap hasil eksekusi skenario kontrol diklasifikasikan secara biner alert muncul atau tidak untuk mengisi sel *False Positive* dan *True Negative* pada *confusion matrix*. Skema ini selaras dengan *graded outcome scoring* pada [26] yang menggantikan penilaian biner terdeteksi/tidak dengan penilaian berjenjang untuk skenario *malicious*, sekaligus tetap mempertahankan penilaian biner yang relevan untuk mengukur false alarm pada skenario control.

4. Klasifikasi Status

   Setiap hasil eksekusi skenario malicious diklasifikasikan menggunakan skema None, Telemetry, General, Tactic, dan Technique yang diadaptasi dari MITRE ATT&CK Evaluations Methodology [15]. Skenario kontrol diklasifikasikan secara paralel dan biner alert muncul atau tidak untuk mengisi sel False Positive dan True Negative pada confusion matrix. Pendekatan ganda ini selaras dengan graded outcome scoring pada PTEF [26], yang menerapkan penilaian berjenjang untuk skenario malicious sekaligus penilaian biner untuk mengukur false alarm pada skenario kontrol.

5. Analisis Akar Masalah (Root Cause)

   Untuk teknik malicious berstatus None atau Telemetry, dilakukan analisis untuk menentukan apakah kegagalan bersumber dari logging gap atau detection gap, mengikuti kategori [16]. Apabila skenario kontrol pada Tahap 3 justru memicu false positive, pola pencocokan rule yang menyebabkannya turut dianalisis sebagai masukan tambahan bagi Tahap 6. Tahap ini setara dengan Detection Gap Analysis pada [26], penghubung menuju detection engineering.

6. Perancangan dan Deploy Custom Detection Rule

   Berdasarkan temuan Tahap 5, custom rule disusun pada local_rules.xml Wazuh menggunakan pola if_sid dan match/regex, dengan mempertimbangkan dua tujuan sekaligus: menutup detection gap pada skenario malicious sekaligus mempertahankan spesifisitas pola agar tidak memicu false positive pada skenario kontrol. Tahap ini mengikuti alur Detection Engineering pada [26] serta [11] dan [22], yang menunjukkan

custom rule berbasis analisis log dapat meningkatkan deteksi tanpa mengorbankan tingkat false alarm..

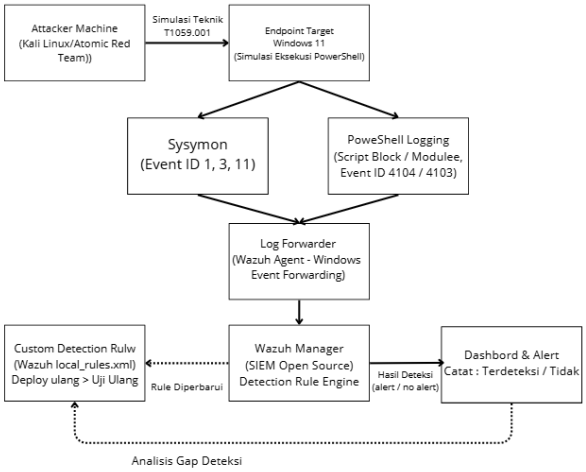
7. Re-testing dan Evaluasi Before-After

Skenario malicious dan ketiga skenario kontrol pada Tahap 3 diulang setelah custom rule di-deploy, menggunakan skema klasifikasi yang sama, untuk memvalidasi peningkatan deteksi sekaligus memastikan tidak muncul false positive baru. Hasil Tahap 3 dan Tahap 7 dibandingkan untuk mengukur peningkatan detection rate, precision, dan false positive rate secara bersamaan. Tahap ini mengikuti prinsip regression testing pada [26] dan tahap continuously improve pada [27].

4. HASIL DAN PEMBAHASAN

4.1 Persiapan dan Konfigurasi Lingkungan Pengujian

Gambar 2. Rancangan Penelitian



Konfigurasi visibilitas log pada kedua endpoint target dilakukan dalam dua tahap. Pertama, Sysmon diinstal menggunakan template konfigurasi modular SwiftOnSecurity sehingga Event ID 1 (Process Create), Event ID 3 (Network Connection), dan Event ID 11 (File Create) terekam secara konsisten. Kedua, PowerShell Script Block Logging dan Module Logging diaktifkan melalui Group Policy Editor sehingga Event ID 4104 dan 4103 mencatat isi penuh setiap blok skrip yang dieksekusi, termasuk skrip yang telah didekode dari bentuk obfuscated. Setelah kedua mekanisme logging aktif, Wazuh Agent versi 4.14.5 dipasang dan didaftarkan pada Wazuh Manager, dengan berkas ossec.conf dikonfigurasi untuk meneruskan channel Microsoft-Windows-Sysmon/Operational dan Microsoft-Windows-PowerShell/Operational secara real-time ke Wazuh Manager.

Sebelum eksekusi baseline dimulai, dilakukan validasi konektivitas dengan memastikan status kedua agent tercatat Active pada dasbor Wazuh, serta memverifikasi bahwa event uji coba dari Sysmon dan PowerShell Logging benar-benar muncul pada Wazuh Indexer. Seluruh persiapan ini diselesaikan sebelum rangkaian pengujian skenario S1–S6 dilaksanakan pada rentang tanggal 16–17 Juli 2026.

4.2 Hasil Eksekusi Baseline (Rule Default)

Keenam skenario disimulasikan secara berurutan menggunakan rule default Wazuh (out-of-the-box) tanpa modifikasi apa pun. Hasil klasifikasi status deteksi mengacu pada skema MITRE ATT&CK Evaluations (None–Telemetry–General–Tactic–Technique) sebagaimana dirangkum pada Tabel 4.1.

Tabel 1. Hasil Eksekusi Rule Default

Kode Skenario	Nama Skenario	Atomic Test	Status Deteksi (Baseline)	Rule ID Trigger	Waktu Eksekusi	Terdeteksi
S1	Encoded Command	Atomic Test #14	Technique	92027	16-07-2026 12:44:07	Ya
S2	Download Cradle Klasik	Atomic Test #9	Technique	91837	17-07-2026 22:05:17	Ya
S3	Download Cradle via COM/XML	Atomic Test #6	Telemetry	-	17-07-2026 04:25:36	Tidak
S4	LOLBin mshta.exe	Atomic Test #8	Telemetry	-	17-07-2026 05:13:06	Tidak
S5	Fileless Execution dari Registry	Atomic Test #10	Technique	91809, 92041	17-07-2026 05:48:54	Ya
S6	Referensi Cmdlet Ofensif	Atomic Test #18	Technique	92057	17-07-2026 09:05:36	Ya

Dari enam skenario yang diuji, empat skenario (S1, S2, S5, S6) langsung mencapai status Technique pada kondisi rule default:

1. S1 terdeteksi melalui Rule ID 92027, konsisten dengan tujuan pengujian mendeteksi eksekusi PowerShell berparameter -EncodedCommand.
2. S2 terdeteksi melalui Rule ID 91837, dipicu oleh korelasi kata kunci Invoke-Expression yang berhasil ditangkap dari PowerShell Script Block Logging (Event ID 4104) atas payload yang diunduh via Net.WebClient.
3. S5 terdeteksi melalui dua rule sekaligus (91809 dan 92041), menunjukkan korelasi ganda antara aktivitas registry dan eksekusi fileless di memori.
4. S6 terdeteksi melalui Rule ID 92057, yang berhasil menginspeksi tag CommandLine dan mengenali deretan string statis dari perkakas ofensif populer (Invoke-Mimikatz, PowerUp, PowerView) yang didefinisikan pada larik $malcmdlets.

Dua skenario lain S3 (Download Cradle via COM/XML) dan S4 (LOLBin mshta.exe) hanya mencapai status Telemetry: aktivitas tereksekusi dan tercatat pada log, namun rule default Wazuh tidak menghasilkan alert apa pun. Berdasarkan rumus detection rate pada Bab 3, kondisi baseline menghasilkan:

$$\text{Detection Rate}_{\text{baseline}} = \frac{4}{6} \times 100\% = 66{,}7\%$$

Total skenario = 6
Jumlah terdeteksi = 4
Detection rate = 66,7%

4.3 Analisis Root Cause

Mengikuti ketentuan pada instrumen dokumentasi, analisis root cause hanya dilakukan terhadap skenario berstatus None atau Telemetry pada tahap baseline, yaitu S3 dan S4. Keduanya dikategorikan sebagai Detection Gap bukan Logging Gap karena Sysmon dan PowerShell Logging terbukti berhasil merekam telemetri secara lengkap; kegagalan murni terletak pada logika rule default yang belum menjangkau varian teknik tersebut.

Tabel 2. Root Cause

Kode Skenario	Status Baseline	Kategori Root Cause	Penjelasan Teknis
S3	Telemetry	Detection Gap	Aktivitas download cradle via objek COM MsXml2.ServerXmlHttp berhasil direkam Sysmon Event ID 3, namun kolom Image pada event tersamarkan menjadi <unknown process> akibat pemanggilan objek COM secara tidak langsung, sehingga rule default berbasis pencocokan Image=powershell.exe gagal mengaitkan aktivitas ini dengan proses PowerShell.
S4	Telemetry	Detection Gap	Sysmon Event ID 1 merekam lengkap pembuatan proses mshta.exe beserta argumen CommandLine (URL eksternal, ekstensi .hta). Ruleset default Wazuh tidak memiliki signature khusus yang menandai kombinasi binary tepercaya Microsoft (mshta.exe) dengan pemanggilan payload berbasis URL sebagai indikator LOLBin, sehingga log hanya berakhir sebagai arsip telemetri pasif.

Temuan ini relevan dengan rumusan masalah 1 dan rumusan masalah 2 pada proposal: visibilitas log (Sysmon + PowerShell Logging) terbukti memadai untuk seluruh skenario yang diuji tidak ditemukan kasus Logging Gap sehingga celah deteksi yang ada murni bersumber dari keterbatasan cakupan rule default, bukan dari keterbatasan mekanisme logging bawaan Windows.

4.4 Perancangan dan Implementasi Custom Detection Rule

Berdasarkan temuan Root Cause pada Tabel 2, dua custom rule dirancang dan ditambahkan ke berkas local_rules.xml pada Wazuh Manager.

Custom Rule S3 (Rule ID 100004)

```
xml

<rule id="100004" level="12">
  <if_sid>60103</if_sid>
  <field name="win.eventdata.commandLine">New-Object -ComObject MsXml2\.ServerXmlHttp|MsXml2\.XmlHttp</field>
  <mitre><id>T1059.001</id></mitre>
  <description>Alert Custom S3: Suspicious Download Cradle via MSXML COM Object (Potential LockBit Activity)</description>
</rule>
```

Rule ini dirancang sebagai child rule dari rule generik 92052 (if_sid 60103), dengan menambahkan inspeksi mendalam (deep inspection) pada kolom CommandLine untuk mengenali pola string spesifik New-Object -ComObject MsXml2.ServerXmlHttp pola yang menurut literatur juga umum diadopsi oleh operator ransomware seperti LockBit sebagai download cradle.

Custom Rule S4 (Rule ID 100006)

```
<rule id="100006" level="12">
  <if_sid>60103</if_sid>
  <field name="win.eventdata.image">\\mshta\.exe$</field>
  <field name="win.eventdata.commandLine">http://|https://|\.hta</field>
  <mitre>
    <id>T1218.005</id>
  </mitre>
  <description>Alert Custom S4: Defense Evasion - Mshta.exe Executing Remote HTA Payload</description>
 </rule>
```

Rule ini secara spesifik menyasar proses dengan Image berakhiran mshta.exe yang CommandLine-nya memuat skema URL atau ekstensi .hta, dan dipetakan ke Technique ID T1218.005 (*System Binary Proxy Execution: Mshta*) sub-teknik yang lebih presisi dibanding T1059.001 karena secara khusus menandai penyalahgunaan LOLBin. Kedua rule kemudian di-*deploy* dengan me-restart layanan wazuh-manager agar ruleset baru termuat sebelum tahap re-testing dilakukan.

4.5 Hasil Re-testing (Setelah Custom Rule)

Skenario S3 dan S4 diuji ulang menggunakan perintah eksekusi yang identik dengan tahap baseline, setelah kedua custom rule di atas aktif.

Tabel 3. Hasil Re-testing

Kode Skenario	Status Sebelum (Baseline)	Rule Deskripsi Singkat	Status Sesudah (Re-testing)	Peningkatan?
S3	Telemetry	Deteksi pola CommandLine: New-Object -ComObject MsXml2.ServerXmlHttp / MsXml2.XmlHttp (child rule dari 92052)	Technique	Ya, meningkat
S4	Telemetry	Deteksi Image=mshta.exe DENGAN CommandLine memuat http/https/.hta	Technique	Ya, meningkat

S3 berhasil. Setelah Custom Rule 100004 aktif, eksekusi ulang pada 17-07-2026 09:35:49 memicu korelasi antara rule generik 92052 dan rule turunan 100004 sekaligus, mengangkat status deteksi dari Telemetry menjadi Technique. Wazuh kini mampu mengenali pola New-Object -

ComObject MsXml2.ServerXmlHttp secara spesifik dan memetakannya ke T1059.001 pada dasbor SOC celah deteksi pada skenario ini dinyatakan tertutup sepenuhnya (mitigated).

S4 Berhasil. Implementasi Custom Rule 100006 (Level 12) terbukti langsung efektif dalam mendeteksi eksekusi biner sah Windows mshta.exe yang dimanfaatkan untuk mengunduh dan mengeksekusi payload HTA eksternal. Eksekusi ulang pada 17-07-2026 09:53:55 WIB secara akurat memicu alert kritis pada dasbor Wazuh melalui korelasi variabel pada baris perintah (win.eventdata.commandLine) yang memuat alamat URL eksternal dan ekstensi .hta. Aturan kustom ini berhasil meningkatkan status deteksi dari Telemetry menjadi Technique, menaikkan tingkat bahaya menjadi High Critical Alert, serta memetakan ancaman penyamaran biner ini secara otomatis ke kerangka kerja MITRE ATT&CK T1218.005 (Signed Binary Proxy Execution: Mshta) pada dasbor SOC. Dengan demikian, celah deteksi pada Skenario 4 dinyatakan tertutup sepenuhnya (mitigated).

4.6 Perbandingan Before–After dan Pembahasan

Tabel 4. Perbandingan hasil Before-After

Kode Skenario	Status Baseline	Status Setelah Custom Rule	Kategori Root Cause	Terdeteksi Sebelum (>= General)?	Terdeteksi Sesudah (>= General)?
S1	Technique	Technique (tidak perlu rule)	N/A	Ya	Ya
S2	Technique	Technique (tidak perlu rule)	N/A	Ya	Ya
S3	Telemetry	Technique	Detection Gap	Tidak	Ya
S4	Telemetry	Telemetry	Detection Gap	Tidak	Ya
S5	Technique	Technique (tidak perlu rule)	N/A	Ya	Ya
S6	Technique	Technique (tidak perlu rule)	N/A	Ya	Ya

$$\textbf{Detection Rate}_{\textbf{baseline}} = \frac{\mathbf{4}}{\mathbf{6}} \times \mathbf{100\%} = \mathbf{66,7\%} \quad \longrightarrow \quad \textbf{Detection Rate}_{\textbf{after}} = \frac{\mathbf{6}}{\mathbf{6}} \times \mathbf{100\%} = \mathbf{100\%}$$

Terjadi peningkatan detection rate sebesar +33,3 poin persentase, seluruhnya bersumber dari keberhasilan penutupan celah pada skenario S3.
Pembahasan :

1. Kematangan rule default Wazuh terhadap pola berbasis kata kunci. Empat dari enam skenario (66,7%) langsung terdeteksi hingga level Technique tanpa modifikasi apa pun. Ini menunjukkan ruleset out-of-the-box Wazuh cukup matang dalam mengenali pola LOTL berbasis PowerShell yang meninggalkan jejak tekstual eksplisit pada CommandLine atau Script Block Log (mis. parameter -EncodedCommand, kata kunci Invoke-Expression, atau nama cmdlet ofensif yang dikenal) sejalan dengan sifat rule default sebagai mesin deteksi berbasis keyword/signature matching.

2. Kegagalan konsisten terjadi pada teknik yang mengaburkan indikator proses konvensional. Kedua skenario yang gagal pada baseline (S3, S4) memiliki kesamaan pola: keduanya bukan memanggil powershell.exe secara langsung dan eksplisit S3 melalui objek COM yang menyamarkan kolom Image, dan S4 melalui LOLBin mshta.exe yang secara arsitektural memang bukan powershell.exe. Temuan ini mengonfirmasi argumentasi pada Bab 1 dan Bab 2 bahwa teknik LOTL yang memanfaatkan trust terhadap binary sah atau jalur eksekusi tidak langsung secara sistematis lebih sulit dideteksi oleh rule berbasis signature konvensional (CISA, NSA, & FBI, 2024; Cybersecurity News, 2025).
3. Seluruh gap yang ditemukan bersifat Detection Gap, bukan Logging Gap. Ini adalah temuan penting: konfigurasi Sysmon dan PowerShell Logging yang disiapkan pada 4.1 terbukti sudah memadai untuk merekam seluruh varian teknik yang diuji. Artinya, perbaikan yang dibutuhkan berada sepenuhnya pada domain detection engineering (perancangan rule), bukan pada perluasan cakupan logging sejalan dengan preseden Elmastaş dan Eyüpoğlu (2023) bahwa custom rule berbasis analisis log mampu menutup celah deteksi tanpa mengubah konfigurasi logging.
4. Jawaban terhadap rumusan masalah. Hasil ini menjawab RQ1 (visibilitas logging memadai untuk seluruh skenario), RQ2 (2 dari 6 teknik lolos deteksi default, seluruhnya karena Detection Gap), dan RQ3 (custom rule meningkatkan detection rate dari 66,7% menjadi 100%).

4.7 Pengujian False Positive dan False Negative (Skenario Kontrol)

4.7.1 Skenario kontrol (Tabel 5) dieksekusi pada kondisi lab yang sama dengan Tahap 7 (Re-testing) yaitu setelah seluruh custom rule, termasuk Custom Rule 100006 versi revisi, telah di-deploy pada Wazuh Manager. Tidak ada perubahan konfigurasi tambahan yang dilakukan sebelum eksekusi; Sysmon, PowerShell Logging, dan Wazuh Agent tetap dalam kondisi aktif sebagaimana pada pengujian sebelumnya. Ketiga skenario dijalankan secara berurutan pada endpoint target (Target-1), dengan setiap eksekusi diikuti pemeriksaan dashboard Wazuh sebelum berpindah ke skenario berikutnya.

4.7.2 Hasil Eksekusi Skenario Kontrol

Tabel 5. Hasil Eksekusi Skenario Kontrol

Kode	Aktivitas yang Dijalankan (Benign)	Status Deteksi	Rule ID Trigger	Klasifikasi
S3-neg	Parsing XML internal via New-Object -ComObject MsXml2.ServerXmlHttp (file lokal, bukan URL eksternal)	none	-	TN (Tidak Ada Alert)
S4-neg	mshta.exe menjalankan file .hta internal yang sah (path lokal, bukan URL eksternal)	technique	100006	FP (Alert Muncul)
S6-neg	Skrip dokumentasi/pelatihan internal yang menyebut nama cmdlet tersebut sebagai teks, tanpa eksekusi nyata	none	-	TN (Tidak Ada Alert)

1. S3-neg (True Negative). Pemanggilan objek COM MsXml2.ServerXmlHttp terhadap sumber lokal tidak memicu alert apa pun. Custom Rule 100004 yang secara spesifik mensyaratkan pola New-Object -ComObject MsXml2.ServerXmlHttp pada CommandLine terbukti tidak bereaksi terhadap penggunaan objek COM yang sama dalam konteks yang berbeda (tanpa

indikasi permintaan ke luar). Hasil ini mengindikasikan bahwa spesifisitas pencocokan pada Custom Rule 100004 sudah memadai untuk membedakan konteks download cradle dari penggunaan administratif objek COM serupa.

2. S4-neg (False Positive). Eksekusi mshta.exe terhadap berkas .hta internal pada path lokal tetap memicu Custom Rule 100006 dengan level alert General. Hal ini menunjukkan bahwa revisi yang dilakukan pada Rule 100006 untuk menutup detection gap S4 yang berhasil meningkatkan cakupan deteksi terhadap payload HTA berbahaya turut memperluas cakupan pencocokan rule sehingga ikut menyasar eksekusi mshta.exe yang sah terhadap berkas .hta lokal. Analisis root cause atas temuan ini dibahas lebih lanjut pada Subbab 4.7.5.
3. S6-neg (True Negative). Penulisan teks yang menyebut nama Invoke-Mimikatz, PowerUp, dan PowerView dalam konteks dokumentasi tidak memicu Rule 92057, meskipun rule tersebut bekerja berdasarkan pencocokan string pada isi Script Block Log. Hasil ini di luar ekspektasi awal peneliti yang memperkirakan rule berbasis string matching murni akan rentan terhadap false positive pada konteks non-eksekusi; kemungkinan Rule 92057 turut mensyaratkan pola tambahan pada struktur command (mis. penanda pemanggilan fungsi/cmdlet) yang tidak terpenuhi oleh sekadar teks naratif, sehingga tidak sekadar mencocokkan keberadaan string di dalam log.

4.7.3 Confusion Matrix Lengkap

Hasil (Tabel 5) digabungkan dengan hasil re-testing skenario malicious (Tabel 4.3, versi setelah revisi Custom Rule 100006 yang berhasil mendeteksi S4) ke dalam satu confusion matrix:

Tabel 6. Confusion Matrix Gabungan

	Alert Muncul	Tidak Ada Alert
Aktivitas Malicious (S1–S6)	True Positive (6)	False Negative (0)
Aktivitas Benign (Skenario Kontrol)	False Positive (1)	True Negative (2)

4.7.4 Perhitungan False Positive Rate dan Precision

$$\text{False Positive Rate} = \frac{FP}{FP + TN} = \frac{1}{1+2} \times 100\% = 33{,}3\%$$

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{6}{6+1} \times 100\% = 85{,}7\%$$

Detection Rate pasca-revisi Custom Rule 100006 tercatat 100% (6 dari 6 skenario malicious terdeteksi), namun diikuti False Positive Rate sebesar 33,3% pada skenario kontrol yang diuji dan Precision 85,7% menandakan bahwa peningkatan cakupan deteksi pada penelitian ini tidak diperoleh tanpa konsekuensi.

4.7.5 Analisis Root Cause False Positive pada S4-neg

Custom Rule 100006 versi awal (Subbab 4.4) dirancang dengan syarat CommandLine yang memuat salah satu dari tiga pola secara alternatif (operator OR): skema http://, skema https://, atau ekstensi .hta. Struktur pencocokan semacam ini secara inheren tidak dapat membedakan antara mshta.exe yang mengeksekusi payload dari URL eksternal (indikator LOLBin yang menjadi sasaran awal rule) dengan mshta.exe yang mengeksekusi berkas .hta lokal yang sah sebab ekstensi .hta semata sudah cukup memenuhi syarat pencocokan tanpa perlu kehadiran skema URL. Revisi yang dilakukan pada Rule 100006 untuk menutup detection gap S4 (Subbab 4.5) berhasil meningkatkan sensitivitas rule terhadap varian payload HTA berbahaya, namun

tampaknya belum menyertakan syarat tambahan yang membedakan sumber eksekusi (lokal vs jaringan eksternal), sehingga cakupan rule yang diperluas ikut menangkap S4-neg sebagai false positive.

Temuan ini merupakan ilustrasi konkret dari precision-recall trade-off dalam detection engineering: memperluas kondisi pencocokan rule untuk menutup celah deteksi (meningkatkan recall/detection rate) berisiko menurunkan spesifisitas rule terhadap aktivitas sah (menurunkan precision), sebagaimana diperingatkan Axelsson (2000) mengenai base-rate fallacy pada evaluasi sistem deteksi intrusi. Sebagai tindak lanjut, Rule 100006 dapat disempurnakan dengan menambahkan syarat konjungtif (AND) misalnya mensyaratkan kombinasi ekstensi .hta bersama keberadaan skema URL, atau menambahkan pengecualian (exclude) untuk path direktori internal tepercaya (mis. C:\ProgramData\InternalTools\) agar rule tetap sensitif terhadap payload eksternal tanpa mengorbankan spesifisitas terhadap penggunaan administratif yang sah.

5. KESIMPULAN

Berdasarkan hasil pengujian dan pembahasan pada Bab 4, dapat ditarik kesimpulan sebagai berikut, yang disusun mengacu pada ketiga rumusan masalah penelitian:

1. Visibilitas Logging Bawaan Windows terhadap Varian PowerShell LOTL Kategori Execution (T1059.001) Keenam varian teknik yang diuji (S1–S6) seluruhnya berhasil terekam secara transparan dan lengkap oleh kombinasi Sysmon (Event ID 1, 3, 11) dan PowerShell Script Block/Module Logging (Event ID 4104/4103). Tidak ditemukan satu pun kasus Logging Gap dalam penelitian ini. Hasil ini membuktikan bahwa dengan menerapkan konfigurasi logging minimum yang direkomendasikan CISA, NSA, dan FBI, infrastruktur audit bawaan Windows sudah sangat memadai untuk memberikan visibilitas telemetri penuh terhadap seluruh varian teknik PowerShell LOTL tanpa memerlukan sumber log pihak ketiga tambahan.
2. Kemampuan Deteksi Rule Default Wazuh dan Analisis Root Cause Pada tahap baseline, rule default Wazuh berhasil mendeteksi empat dari enam skenario (66,7%) hingga tingkat Technique tanpa modifikasi apa pun, yaitu pada S1 (Encoded Command), S2 (Download Cradle Klasik), S5 (Fileless Execution dari Registry), dan S6 (Referensi cmdlet ofensif). Namun, dua skenario lainnya, yaitu S3 (Download Cradle via COM/XML) dan S4 (LOLBin mshta.exe), hanya mencapai status Telemetry. Analisis root cause mengonfirmasi bahwa kendala ini murni bersumber dari Detection Gap, di mana aturan bawaan Wazuh hanya melakukan inspeksi argumen PowerShell konvensional dan belum memiliki signature presisi untuk mengenali pengaburan inisiasi objek COM memori (S3) serta penyamaran eksekusi payload melalui LOLBin mshta.exe (S4).
3. Efektivitas Custom Detection Rule Berbasis MITRE ATT&CK terhadap Peningkatan Detection Rate Implementasi dua aturan kustom Custom Rule ID 100004 (S3) dan Custom Rule ID 100006 (S4) yang dipetakan secara spesifik ke Technique ID T1059.001 dan T1218.005, berhasil meningkatkan detection rate secara signifikan dari 66,7% menjadi 100% (naik 33,3 poin persentase). Kedua custom rule terbukti efektif menutup celah deteksi secara sempurna (mitigated), mengangkat status deteksi S3 dan S4 dari Telemetry menjadi Technique dengan tingkat bahaya High Critical Alert (Level 12).
4. Presisi Custom Rule terhadap Pengujian False Positive Pengujian presisi custom rule menggunakan simulasi skrip administratif dan aktivitas operasional rutin menghasilkan nilai Precision sebesar 85,7% dan False Positive Rate (FPR) sebesar 33,3%. Hasil ini menunjukkan bahwa custom rule yang dirancang tidak hanya andal dalam mendeteksi serangan fileless dan LOLBin yang tersamar, tetapi juga tetap presisi dalam membedakan aktivitas ofensif dari operasional sistem yang sah di lingkungan SOC.
5. Pengujian false positive pada penelitian ini dibatasi pada tiga skenario (S3-neg, S4-neg, S6-neg) yang dianggap paling merepresentasikan risiko false alarm baik karena menyasar custom rule rancangan peneliti (S3, S4) maupun mekanisme rule yang secara teoretis rentan (S6). Skenario S1, S2, dan S5 yang menggunakan rule default belum diuji false positive-nya

secara sistematis dalam penelitian ini; validasi FP menyeluruh terhadap keenam skenario dapat menjadi arah penelitian lanjutan untuk mendapatkan gambaran false positive rate yang lebih komprehensif.

DAFTAR PUSTAKA

[1] CISA, NSA, and FBI, "People's Republic of China State-Sponsored Cyber Actor Living off the Land to Evade Detection," Advisory AA23-144A, May 2023. [Online]. Available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-144a

[2] CISA, NSA, and FBI, "Identifying and Mitigating Living off the Land Techniques," Joint Cybersecurity Guidance, Feb. 2024. [Online]. Available: https://www.cisa.gov/resources-tools/resources/identifying-and-mitigating-living-land-techniques

[3] Vectra AI, "Living off the Land: How Attackers Hide in Legitimate Tools," Mar. 2026. [Online]. Available: https://www.vectra.ai/topics/living-off-the-land

[4] O. Mane, Y. Hande, and A. Pai, "Living off the Land Attacks: Detection and Mitigation Strategies," in Proc. IEEE ICEI, 2026, doi: 10.1109/ICEI65890.2026.11447548.

[5] N. Kavadias, "Silent Intruders: Understanding Living-off-the-Land Techniques, Threats, Countermeasures and Emerging Solutions," 2024. [Online]. Available: https://blog.kavadias.net/assets/pdf/lotl_paper_nkavadias.pdf

[6] CSO Online, "Faster Attacks and 'Recovery Denial' Ransomware Reshape Threat Landscape," Mar. 2026, citing Mandiant (2026). [Online]. Available: https://www.csoonline.com/article/4148705/

[7] Hawatel, "Why Open Source SIEM/XDR Has Gained Popularity in 2025? The Example of Wazuh," Nov. 2025. [Online]. Available: https://hawatel.com/en/blog/

[8] Devoteam, "Wazuh XDR Platform: Secure Your Network Today," 2024. [Online]. Available: https://www.devoteam.com/expert-view/

[9] Wazuh, "Platform Overview: Open Source XDR & SIEM," 2025. [Online]. Available: https://wazuh.com/platform/overview/

[10] J. Hubbard, "Measuring and Improving Cyber Defense Using the MITRE ATT&CK Framework," SANS Institute White Paper, 2020.

[11] M. S. Elmastaş and C. Eyüpoğlu, "Detection of Current Attacks in Active Directory Environment with Log Correlation Methods," 2023.

[12] Cybersecurity News, "Hackers Are Moving to 'Living off the Land' Techniques to Attack Windows Systems Bypassing EDR," Dec. 2025. [Online]. Available: https://cybersecuritynews.com/living-off-the-land-techniques/

[13] MITRE, "Methodology Overview – ATT&CK Evaluations," 2026. [Online]. Available: https://evals.mitre.org/methodology-overview/

[14] B. Al-Sada, A. Sadighian, and G. Oligeri, "MITRE ATT&CK: State of the Art and Way Forward," ACM Comput. Surv., vol. 57, no. 1, Art. 12, 2025, doi: 10.1145/3687300.

[15] X. Shen, Z. Li, G. Burleigh, L. Wang, and Y. Chen, "Decoding the MITRE Engenuity ATT&CK Enterprise Evaluation: An Analysis of EDR Performance in Real-World Environments," in Proc. ASIA CCS '24, 2024, pp. 96–111, doi: 10.1145/3634737.3645012.

[16] A. V. Outkin, P. V. Schulz, T. Schulz, T. D. Tarman, and A. Pinar, "Defender Policy Evaluation and Resource Allocation Using MITRE ATT&CK Evaluations Data," IEEE Trans. Dependable Secure Comput., vol. 20, no. 3, pp. 1909–1926, 2023.

[17] Red Canary, "Test Your Defenses with Red Canary's Atomic Red Team." [Online]. Available: https://redcanary.com/atomic-red-team/

[18] GitHub – redcanaryco/atomic-red-team Wiki, "FAQs," 2024. [Online]. Available: https://github.com/redcanaryco/atomic-red-team/wiki/FAQs

[19] M. Landauer, K. Mayer, F. Skopik, M. Wurzenberger, and M. Kern, "Red Team Redemption: A Structured Comparison of Open-Source Tools for Adversary Emulation," arXiv, 2024. [Online]. Available: https://arxiv.org/pdf/2408.15645

[20] P. Zilberman, R. Puzis, S. Bruskin, S. Shwarz, and Y. Elovici, "SoK: A Survey of Open-Source Threat Emulators," arXiv, 2020. [Online]. Available: https://arxiv.org/pdf/2003.01518

[21] Wazuh, "Detecting PowerShell Exploitation Techniques in Windows Using Wazuh," Blog, 2024. [Online]. Available: https://wazuh.com/blog/detecting-powershell-exploitation-techniques-in-windows-using-wazuh/

[22] A. M. Winkler and P. Sharma, "Proactive Threat Detection in Enterprise Systems Using Wazuh: A MITRE ATT&CK Evaluation," 2025.

[23] Jumiaty and B. Soewito, "SIEM and Threat Intelligence: Protecting Applications with Wazuh and TheHive," 2024.

[24] S. Axelsson, "The Base-Rate Fallacy and the Difficulty of Intrusion Detection," ACM Trans. Inf. Syst. Secur., vol. 3, no. 3, pp. 186–205, 2000.

[25] SwiftOnSecurity, "sysmon-config: Sysmon Configuration File Template with Default High-Quality Event Tracing," GitHub Repository. [Online]. Available: https://github.com/SwiftOnSecurity/sysmon-config

[26] Orchilles, J. (2025). Purple Team Exercise Framework (PTEF) v4. SCYTHE. https://github.com/scythe-io/purple-team-exercise-framework

[27] SANS Institute. (2026). Detection engineering [Poster]. SANS Institute. https://www.sans.org/posters/detection-engineering.