E-ISSN: 3031-8521
P-ISSN: 3032-1867
Halaman 3193-3205
Volume 3 Nomor 4 Tahun 2026

EVIDENCE-GRADED NETWORK FORENSIC RECONSTRUCTION OF AN URSNIF (GOZI/ISFB) DOWNLOAD CHAIN: FROM HOST ATTRIBUTION TO A HASH-VERIFIED DLL ARTIFACT

Isnaini Nur Hasanah[1], Mohammad Riyadh[1]

[1,2] Institut Teknologi Sains Bandung, Kabupaten Bekasi, Indonesia
E-mail: isnaini.hasanah@student.itsb.ac.id[1], mohammad.riyadh@student.itsb.ac.id[2]

Abstrak

Rekaman jaringan sering menjadi satu-satunya catatan yang tersisa dari infeksi malware, tetapi laporan jarang menunjukkan bagaimana setiap pernyataan tentang insiden dikaitkan dengan bukti tertentu. Makalah ini merekonstruksi infeksi Ursnif (Gozi/ISFB) dari latihan PCAP berisi 24.000 paket menggunakan Wireshark dan hex editor, serta menilai setiap peristiwa hasil rekonstruksi sebagai teramati langsung (A), turunan (B), pihak ketiga (C), atau tidak terbukti (D). Rekonstruksi mengatribusikan host (10.1.21.101, DESKTOP-NB72TZA), membangun kembali rantai unduhan (GET /Lk9tdZ, HTTP 302, dan GET /files/1.bin 140 ms kemudian), dan mengekstrak berkas PE sebesar 340.824 byte dari aliran TCP 52. Artefak tersebut diperiksa dengan empat uji konsistensi: posisi MZ setelah penutup header, ukuran hasil ekstraksi terhadap Content-Length, nilai ETag, dan kecocokan awalan SHA-256. Linimasa relatif dijangkarkan ke UTC melalui header Date server. Lalu lintas pasca-infeksi, termasuk URL /postfix/ yang panjang, /grab32.rar, dan kueri DNS, dicantumkan beserta tingkat buktinya. Vektor infeksi awal, perilaku payload, dan pembuat sampel tidak terbukti, dan makalah ini menunjukkan di mana bukti berakhir.

Kata Kunci: *forensik jaringan, rekonstruksi peristiwa, Ursnif, Gozi/ISFB, penilaian bukti, PCAP, ekstraksi artefak.*

Abstract

Network captures are often the only surviving record of a malware infection, yet write-ups rarely show how each statement about an incident ties to specific evidence. This paper reconstructs an Ursnif (Gozi/ISFB) infection from a 24,000-packet PCAP exercise using Wireshark and a hex editor, and grades each reconstructed event as directly observed (A), derived (B), third-party (C), or not established (D). The reconstruction attributes the host (10.1.21.101, DESKTOP-NB72TZA), rebuilds the download chain (GET /Lk9tdZ, an HTTP 302, and GET /files/1.bin 140 ms later), carves a 340,824-byte PE file from TCP stream 52. It checks the artifact with four consistency tests: MZ placement after the header terminator, carved size versus Content-Length, the ETag

value, and a SHA-256 prefix match. A relative timeline is anchored to UTC through the server Date header. Post-infection traffic, including long /postfix/ URLs, /grab32.rar, and DNS lookups, is listed with grades. The initial infection vector, payload behavior, and authorship remain unestablished, and the paper shows where the evidence ends.

Keywords: *network forensics; event reconstruction; Ursnif; Gozi/ISFB; evidence grading; PCAP; artifact carving*

INTRODUCTION

A packet capture is often the only artifact that survives a malware infection: the endpoint may be reimaged, but the network record remains. Analysts therefore have to reconstruct what happened from packet data, and intrusion-detection alerts help only partly because signature-based alerts can be wrong or overlap (Sommer & Paxson, 2010; García-Teodoro et al., 2009). In the exercise studied here, for example, a JA3 alert suggested “possible Gozi” on traffic that the exercise authors judged benign (Duncan, 2021b).

Ursnif (also Gozi or Gozi-ISFB) is a long-lived Windows banking Trojan (MITRE ATT&CK, n.d.; Microsoft Security Intelligence, n.d.; HC3, 2020). Surveys of network traffic analysis for malware focus on automated detection and classification, including TLS-encrypted traffic and botnet traffic (Oh et al., 2022; Stevanovic & Pedersen, 2016; Gibert et al., 2020). Practitioner walkthroughs of the same malware show how to read the traffic (Duncan, 2019; Duncan, 2021b), but they seldom state, for each claim, whether it was observed, derived, or merely asserted by an external source.

This paper treats the investigation as an *event reconstruction* problem and makes four contributions:

1. **C1.** An evidence-graded reconstruction workflow (A–D grades, frame/time anchoring) from host attribution to a carved artifact.
2. **C2.** Four independent integrity tests for a carved PE file that do not depend on any external reputation service.
3. **C3.** A relative timeline anchored to UTC and a ledger that separates observed events from third-party assertions.
4. **C4.** An explicit statement of what the evidence does *not* establish: initial access, payload behavior, data theft, and authorship.

BACKGROUND AND RELATED WORK

Ursnif and Its Distribution

Ursnif is also reported as Gozi, Gozi-ISFB, ISFB, and Papras. Variants can steal credentials, inject content into web pages, act as a backdoor, and download further payloads (MITRE ATT&CK, n.d.; Microsoft Security Intelligence, n.d.; HC3, 2020). Documented distribution routes include spam with malicious attachments, malicious links, exploit kits, and follow-on infection by other malware (MITRE ATT&CK, n.d.; Duncan, 2019; HC3, 2020). The Gozi source code was reported leaked in 2015 and reused in many variants (HC3, 2020), which is why a family name alone does not identify

an operator. These routes are general context; none is evidence of how the studied host was infected.

Traffic Analysis and Network Forensics

Oh et al. (2022) survey traffic analysis of TLS-encrypted malware for security operations; Stevanovic and Pedersen (2016) review machine-learning identification of botnet traffic; Gibert et al. (2020) review machine learning for malware detection and classification. On the forensic side, packet-level analysis and Wireshark-based investigation are established practice (Sikos, 2020; Ndatinya et al., 2015; Wireshark Foundation, n.d.). The present work is complementary: instead of improving detection, it documents a reproducible *manual* reconstruction with explicit evidence grading, and it is a single-case descriptive study, not a classifier evaluation.

Table 1. Positioning of this work against the reviewed literature.

Work	Main focus	Gap addressed here
Oh et al. (2022)	Survey of traffic analysis for TLS-encrypted malware in security operations.	Manual, per-event grading of what a capture proves in one case.
Stevanovic and Pedersen (2016)	Machine-learning identification of botnet traffic.	Reconstruction of a single download chain, not classification.
Gibert et al. (2020)	Review of machine learning for malware detection and classification.	Artifact carving and verification without a classifier.
Sikos (2020); Ndatinya et al. (2015)	Packet analysis and Wireshark-based network forensics.	Explicit grades, frame/time anchoring, and stated unknowns.
This work	Evidence-graded event reconstruction of an Ursnif download chain.	—

Technical Background

An HTTP 302 response redirects the client through the *Location* header (Fielding et al., 2022). Windows executables and DLLs use the Portable Executable (PE) format, which begins with the two bytes “MZ” (4D 5A) and a DOS stub (Microsoft, n.d.). A cryptographic hash such as SHA-256 identifies a file without comparing it byte by byte; reputation services such as Virus Total report on known hashes but their verdicts change over time (VirusTotal, n.d.).

METHOD

Case Data and Provenance

The data are the 24,000-packet PCAP of the 21 January 2021 traffic-analysis exercise and its answer document (Duncan, 2021a; Duncan, 2021b). The scenario is a LAN 10.1.21.0/24 in domain wokemountain.com with domain controller WokeMountain-DC, gateway 10.1.21.1, and broadcast address 10.1.21.255 (Duncan, 2021b). The

authors performed the Wireshark and HxD steps (Wireshark Foundation, n.d.; Hörz, n.d.) and kept annotated screenshots; the answer document supplies facts that the authors did not observe themselves and is treated as third-party evidence. The PCAP hash, tool versions, and capture span were not recorded and are therefore not reported.

Evidence Model

Each reconstructed statement receives one grade (Table 2). A statement may be *downgraded* but never upgraded by repetition: a fact repeated in several third-party documents remains grade C.

Table 2. Evidence grades.

Gr.	Definition
A	Directly observed in the authors' PCAP/screenshots (value, frame or window visible).
B	Derived by the authors from A-grade items (arithmetic, ordering, consistency).
C	Asserted by a third-party document or service; not independently observed.
D	Not established by the available evidence.

Workflow

Figure 1 shows the seven steps. Steps 1–2 attribute the host and triage the traffic with filters ip.addr == 10.1.21.101, nbns || llmnr, and http.request. Step 3 rebuilds the conversation with *Follow TCP Stream* (Wireshark Foundation, n.d.). Step 4 saves the stream as Raw and carves the file in a hex editor. Step 5 runs the integrity tests below. Steps 6–7 inventory post-infection traffic and compile the ledger with times, frame IDs, and grades.

Table 3. Reproducible steps and display filters.

Step	Tool / filter	Output
1	Wireshark: ip.addr == 10.1.21.101; nbns || llmnr	IP, MAC, host and domain names
2	Wireshark: http.request	66 requests; candidates
3	Follow TCP Stream, tcp.stream eq 52, ASCII	Requests, 302, 200 OK
4	Show as Raw; Save as 1.bin; HxD	Raw export; MZ search
5	HxD: delete 0–0x766; Analysis → Checksums	Carved PE; SHA-256
6	Inventory of URL, DNS, TLS items	Ledger and IOC table

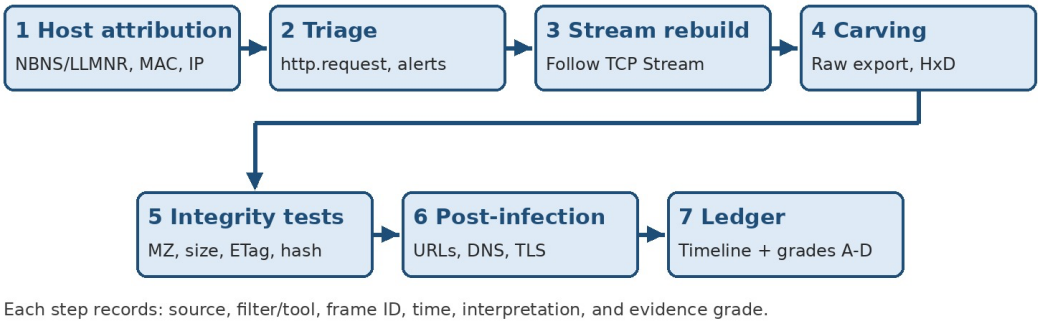


Figure 1. Seven-step evidence-graded reconstruction workflow.

Integrity Tests and Time Anchoring

Four tests are applied to the carved file. T1: the bytes 4D 5A must follow immediately after the CRLF CRLF (0D 0A 0D 0A) that ends the response header. T2: the carved length must equal *Content-Length*. T3: the numeric part of the *ETag* is compared with the size as supporting evidence. T4: the SHA-256 is compared with the reference value (Duncan, 2021b). Capture times are relative to the first packet; they are anchored to UTC by subtracting the relative time of the response from the server *Date* header, which assumes the server and capture clocks agree to within a few seconds (grade B).

Safety

The carved file was never executed; it was only opened in a hex editor and hashed.

RESULTS AND DISCUSSION

Host Attribution

Filtering on ip.addr == 10.1.21.101 shows traffic with 10.1.21.2 over LDAP, DNS, and TCP; a DNS response maps wokemountain-dc.wokemountain.com to 10.1.21.2, so that address acts as the domain controller (Figure 2). NetBIOS registrations from 10.1.21.101 for DESKTOP-NB72TZA and WOKEMOUNTAIN appear at frames 6–7 (t = 0.092 s), and the Ethernet layer shows MAC 00:60:67:d3:47:8b (grade A). The Windows account (orlando.mccoy) and the public IP (104.167.176.238) come only from the answer document (Duncan, 2021b) (grade C); IP and MAC identify a device, not a person.

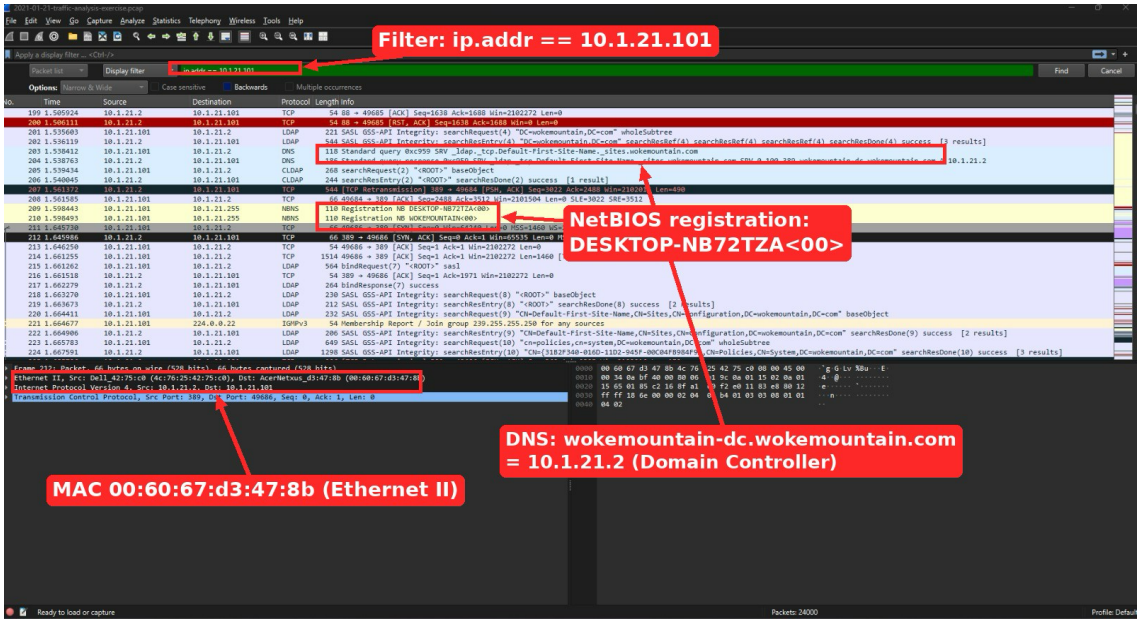


Figure 2. Host attribution: filter on the host IP, DNS to the domain controller, NetBIOS name registration, and Ethernet MAC.

Triage

The http.request filter returns 66 packets (Figure 3). Most early rows are SSDP *M-SEARCH* multicast to 239.255.255.250, which is local device discovery and not malicious. The standouts are GET /Lk9tdZ and GET /files/1.bin to 209.141.51.196, several GET /postfix/… requests with very long URLs to 185.186.244.130, and GET /grab32.rar to 162.0.224.165. The answer document lists 16 alert lines (Table 4); line counts are not unique events and are reported as given (grade C).

Table 4. IDS alert lines reported by the exercise (Duncan, 2021b**).**

#	Destination:port	Meaning
11	209.141.51.196:80	EXE/DLL download over plain HTTP (e.g., “PE EXE or DLL Windows file download HTTP”).
1	72.21.81.200:443	JA3 “possible Gozi”; judged false positive.
1	208.67.222.222:53	External IP lookup (myip.opendns.com).
1	162.0.224.165:80	Request with URL ending in .rar.
2	193.239.84.250:443; 84.252.95.102:443	Malicious SSL certificate (Zeus Panda Banker/Ursnif).

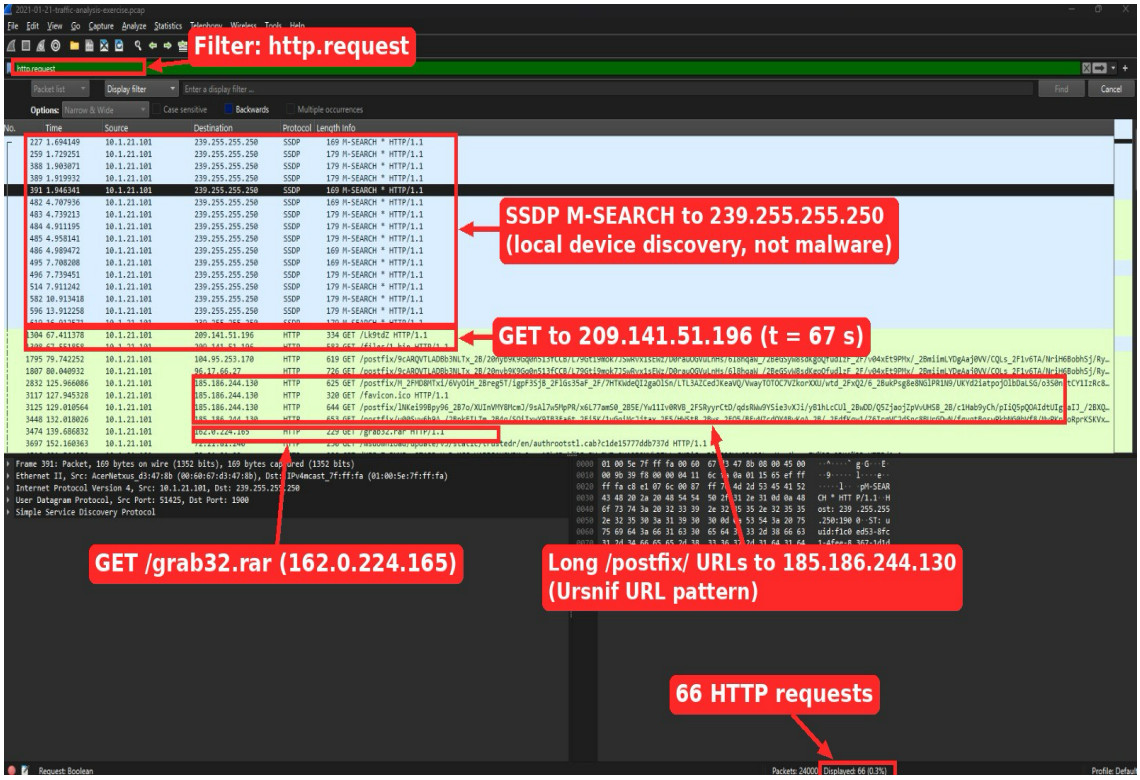


Figure 3. Triage with http.request: SSDP noise, the redirected download, long /postfix/ URLs, and /grab32.rar.

Download-chain Reconstruction

Follow TCP Stream on stream 52 (Figure 4) shows two client packets and 253 server packets in three turns. The reconstructed order is: (1) GET /Lk9tdZ; (2) *302 Found* with *Location: http://209.141.51.196/files/1.bin* and cookies; (3) GET /files/1.bin returning the cookie; (4) *200 OK*, *application/octet-stream*, *Content-Length: 340824*, then "MZ…This program cannot be run in DOS mode". The *Date* header reads 23:40:38 GMT. Frames 1304 and 1308 are 0.140 s apart (67.411 s and 67.552 s), consistent with an automated client following the redirect; this is an inference (grade B), not proof of automation. Anchoring gives a capture start near 23:39:31 UTC (±1 s), which places the later events at about 23:41 UTC and agrees with the minute shown in the answer document's screenshots (Duncan, 2021b). Table 5 lists the reconstructed events and Figure 5 plots them.

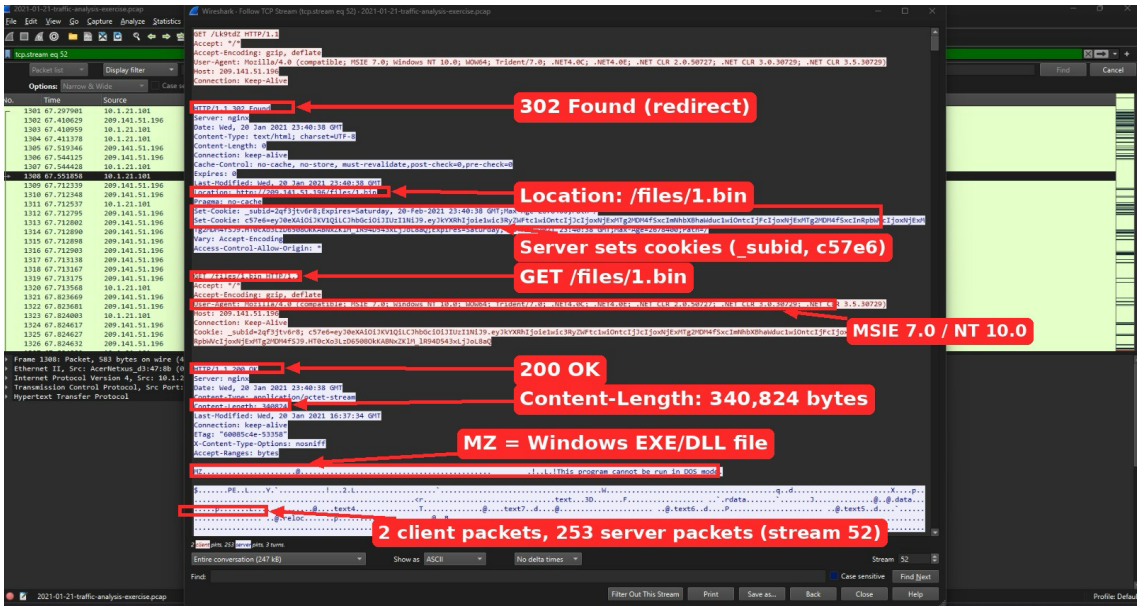


Figure 4. Stream 52: 302 redirect, second request carrying the cookie, 200 OK with 340,824 bytes, and the MZ header.

Table 5. Reconstructed event ledger (t = seconds since capture start; n/r = not recorded).

ID	t (s)	Frame	Event	Grade
E1	0.092	6–7	NBNS registration of DESKTOP-NB72TZA and WOKEMOUNTAIN from 10.1.21.101.	A
E2	67.298	1301	First packet listed for TCP stream 52 (client 10.1.21.101).	A
E3	67.411	1304	GET /Lk9tdZ to 209.141.51.196:80 (334-byte frame).	A
E4	n/r	n/r	302 Found, Location /files/1.bin, Set-Cookie.	A
E5	67.552	1308	GET /files/1.bin with cookie (583-byte frame), +0.140 s.	A
E6	n/r	n/r	200 OK, 340,824 bytes, MZ; Date 23:40:38 GMT.	A
E7	79.742; 80.041	1795; 1807	GET /postfix/… to 104.95.253.170 and 96.17.66.27 (unclassified).	A
E8	125.986–132.018	2832–3448	Three GET /postfix/… and one /favicon.ico to 185.186.244.130 (host greatewallfirewall.xyz per answer document).	A / C (host name)

ID	t (s)	Frame	Event	Grade
E9	139.687	3474	GET /grab32.rar to 162.0.224.165.	A
E10	≈ 23:41 UTC	n/r	DNS myip.opendns.com → 104.167.176.238; TLS to 193.239.84.250 and 84.252.95.102.	C

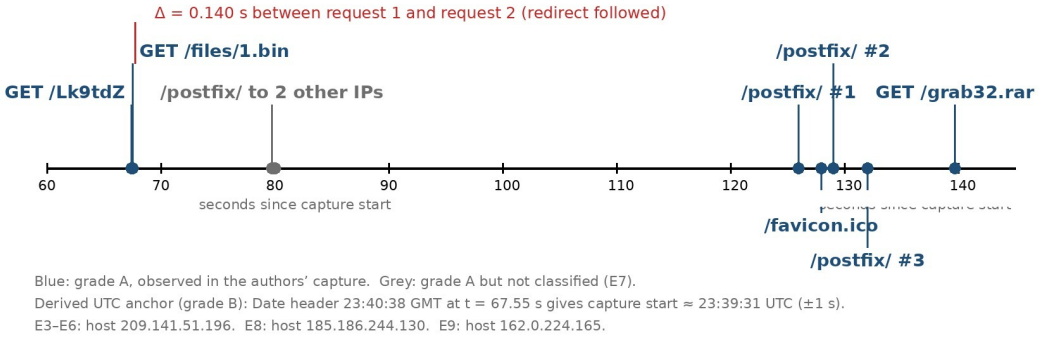


Figure 5. Reconstructed timeline of events E3–E9 (seconds since capture start); the 0.140 s gap separates the first request from the redirected one.

Artifact Carving and Integrity

Saving the stream as Raw yields a file holding both requests, both response headers, and the DLL. In the hex editor the bytes 4D 5A sit at offset 0x767, directly after 0D 0A 0D 0A (Figure 6a). Offsets in HxD are hexadecimal: row 0x760 holds 74 65 73 0D 0A 0D 0A 4D 5A, so MZ is at 0x760 + 7. Selecting and deleting bytes 0 to 0x766 (1,895 bytes) leaves a file that starts with MZ at offset 0 and ends at offset 0x53358, i.e., 340,824 bytes (Figure 7a). The SHA-256 prefix visible in HxD, 236491CFE870F6B374D80E427EF8F8BFBF24F50D, equals the first 40 of 64 hex characters of the reference hash (Duncan, 2021b) (Figure 7b). Table 6 summarizes the tests. Near the end of the file, text such as "Sectigo RSA Time Stamping CA" suggests an appended certificate block; signature validity was not checked.

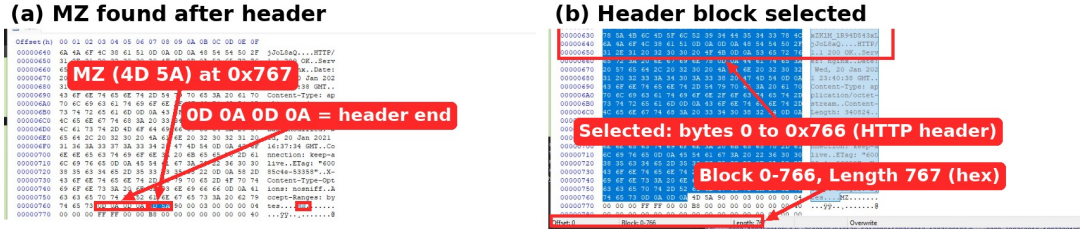


Figure 6. Carving in HxD: (a) MZ directly after the header terminator at offset 0x767; (b) header block 0–0x766 selected for removal.

Table 6. Integrity tests on the carved file.

Test	Expectation	Observation	Result
T1	MZ right after CRLF CRLF	4D 5A at 0x767	Pass (A)
T2	Size = Content-Length 340,824	End offset 0x53358 = 340,824	Pass (A/B)
T3	ETag hex = size	“60085c4e-53358”	Consistent (B)
T4	SHA-256 = reference	40 of 64 hex characters visible and equal	Partial

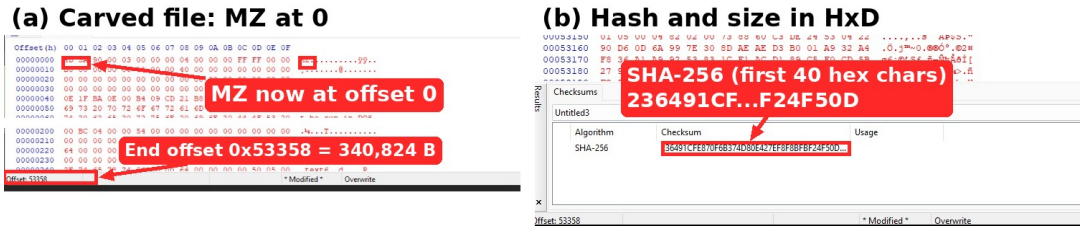


Figure 7. Verification in HxD: (a) carved file begins with MZ and ends at 0x53358; (b) SHA-256 prefix in the checksum panel.

Post-infection Traffic and Indicators

After the download, GET requests with very long /postfix/… URLs went to 185.186.244.130, plus /favicon.ico and /grab32.rar (Table 5). The exercise author reports that follow-on Ursnif components are fetched by GET requests ending in .rar and that the data are encoded or encrypted in transit and decoded on the host (Duncan, 2021b); this agrees with the walkthrough by Duncan (2019). The host also queried myip.opendns.com.wokemountain.com (no such name) and then myip.opendns.com at 208.67.222.222, receiving 104.167.176.238, a rarely legitimate lookup that triggered an alert (Duncan, 2021b). Table 7 lists indicators; they are historical (January 2021) and are not links to open.

Table 7. Indicators of compromise and evidence grade.

Type	Indicator	Gr.
IP:port	209.141.51.196:80 (GET /Lk9tdZ; /files/1.bin)	A
URL	http://209.141.51.196/files/1.bin	A
IP:port	185.186.244.130:80 (/postfix/…, /favicon.ico)	A
Domain	greatewallfirewall.xyz	C
IP:port	162.0.224.165:80 (/grab32.rar)	A
DNS	myip.opendns.com	C
IP:port	193.239.84.250:443; 84.252.95.102:443	C

Type	Indicator	Gr.
SHA-256	236491cfe870f6b374d80e427ef8f8bfbf24f50d4029128b001d95c8c90845cb	A (prefix)/ C

What the Evidence Does Not Establish

The capture shows no HTTP request before second 67 other than SSDP, and HTTPS content is unreadable, so the **initial access vector is not established (D)**. DLL behavior, credential theft, and exfiltration are **not established (D)** because nothing was executed and no payload was decrypted. Authorship is **not established (D)**: the leaked source code was reused widely (HC3, 2020).

DISCUSSION

What grading changed. Grading separated six items that a plain narrative would blend: the account name, public IP, domain name, DNS lookup, TLS alerts, and hash completeness all depend on the answer document or on a truncated value (C or partial), whereas host identity, the redirect chain, the carved size, and the MZ placement are directly observable (A). The independent checks T1–T3 give confidence in the carve even though the full hash comparison is pending.

Value of timing. The 0.140 s gap between the two requests, the Date-header anchoring, and the later requests at roughly 58–72 s after the download turn a list of indicators into an ordered chain. The ordering is a derived inference; it does not identify the process that issued the requests.

Alerts versus reconstruction. Eleven alert lines already pointed to the download, but the JA3 alert shows why an alert needs corroboration (Sommer & Paxson, 2010). The same reconstruction exposed two /postfix/ requests (E7) that none of the alerts explained; they are left unclassified.

Threats to validity. (i) One exercise case limits generalization. (ii) Clock anchoring assumes server and capture clocks agree. (iii) The Follow-Stream dialog labels the conversation “247 kB”, whereas the carved file alone is 340,824 bytes; we could not reconcile this label (possibly a display quirk) and therefore rely on T1–T4 rather than on the label. (iv) Only 40 of 64 hash characters were visible. (v) Several indicators are third-party only. (vi) No dynamic analysis was performed.

Reconstruction rules. The case suggests five working rules for network-forensic write-ups: (R1) anchor every event to a frame and time, and write “not recorded” when absent; (R2) separate transport (headers) from payload before hashing; (R3) test a carve internally (T1–T3) before consulting any reputation service; (R4) never upgrade a third-party statement to observed because several documents repeat it; and (R5) keep unknowns visible as grade D instead of omitting them.

Implications for incident response. The ledger can be handed to a responder as a work list: block or watch the A-grade endpoints, validate the C-grade ones, and pull endpoint logs to close the D-grade gaps (initial access, payload behavior). The grading

also tells a reader how much of a final report is the analyst's own observation, which matters when findings are escalated or reused in a later case.

Future work. Show the full hash and attach the reputation report; classify E7; verify the account through directory-authentication traffic; execute the sample in an isolated sandbox to test the C-grade claims; and compare this manual workflow with automated detectors.

CONCLUSION

A grade-annotated reconstruction of one Ursnif exercise attributes host 10.1.21.101, rebuilds the download chain (GET /Lk9tdZ → 302 → GET /files/1.bin), carves a 340,824-byte PE file that passes tests T1–T3 and matches the reference hash prefix, and orders the subsequent traffic on a UTC-anchored timeline. The method makes explicit which statements are observed, derived, or borrowed, and it states that initial access, payload behaviour, and authorship remain unestablished.

REFERENCES

Duncan, B. (2019). *Wireshark tutorial: Examining Ursnif infections*. Unit 42, Palo Alto Networks. https://unit42.paloaltonetworks.com/wireshark-tutorial-examining-ursnif-infections/

Duncan, B. (2021a). *2021-01-21 – Traffic analysis exercise*. Malware-Traffic-Analysis.net. https://www.malware-traffic-analysis.net/2021/01/21/index.html

Duncan, B. (2021b). *2021-01-21 – Traffic analysis quiz answers* [Answer document, PDF]. Malware-Traffic-Analysis.net. https://www.malware-traffic-analysis.net/2021/01/21/index.html

Fielding, R., Nottingham, M., & Reschke, J. (Eds.). (2022). *HTTP semantics* (RFC 9110). Internet Engineering Task Force. https://www.rfc-editor.org/info/rfc9110

García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security, 28*(1–2). https://doi.org/10.1016/j.cose.2008.08.003

Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications*. https://recercat.cat/handle/10459.1/68344

HC3. (2020, June 16). *Ursnif malware* (HC3 Sector Note 202006160800). U.S. Department of Health and Human Services, Health Sector Cybersecurity Coordination Center. https://www.aha.org/system/files/media/file/2020/06/hc3-sector-report-tlp-wite-ursnif-malware-6-16-20.pdf

Hörz, M. (n.d.). *HxD – Freeware hex editor and disk editor*. mh-nexus.de. https://mh-nexus.de/en/hxd/

Microsoft. (n.d.). *PE format*. Microsoft Learn. https://learn.microsoft.com/en-us/windows/win32/debug/pe-format

Microsoft Security Intelligence. (n.d.). *Trojan:Win32/Gozi*. Microsoft Malware Encyclopedia. https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Trojan%3AWin32%2FGozi

MITRE ATT&CK. (n.d.). *Ursnif, Software S0386*. https://attack.mitre.org/software/S0386/

Ndatinya, V., Xiao, Z., Manepalli, V. R., Meng, K., & Xiao, Y. (2015). Network forensics analysis using Wireshark. *International Journal of Security and Networks, 10*(2). https://doi.org/10.1504/IJSN.2015.070421

Oh, C., Ha, J., & Roh, H. (2022). A survey on TLS-encrypted malware network traffic analysis applicable to security operations centers. *Applied Sciences, 12*(1), Article 155. https://doi.org/10.3390/app12010155

Sikos, L. F. (2020). Packet analysis for network forensics: A comprehensive survey. *Forensic Science International: Digital Investigation, 32*, Article 200892. https://doi.org/10.1016/j.fsidi.2019.200892

Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *Proceedings of the IEEE Symposium on Security and Privacy*. https://doi.org/10.1109/SP.2010.25

Stevanovic, M., & Pedersen, J. M. (2016). On the use of machine learning for identifying botnet network traffic. *Journal of Cyber Security and Mobility, 4*(2–3). https://doi.org/10.13052/jcsm2245-1439.421

VirusTotal. (n.d.). *File report for SHA-256 236491cf…0845cb*. https://www.virustotal.com/gui/file/236491cfe870f6b374d80e427ef8f8bfbf24f50d4029128b001d95c8c90845cb/details

Wireshark Foundation. (n.d.). *Wireshark documentation*. https://www.wireshark.org/docs/