Network Forensics
&
Malicious URL

Prepared By:
Kazim Ali Obad

Supervisor:

Anmar Mohammed
MOHAMMED .B. HASSAN

Contents

What is Network Forensics?

It's a type of digital investigation, meaning it sits within the broader world of forensic analysis. But what makes it different from other forensic disciplines is that it focuses specifically on computer network traffic.

So what does network forensics actually do? It helps experts collect evidence, identify cyber attacks, and understand what's happening inside a network. Now here's the critical thing and this is what separates network forensics from other types of digital investigation:

> ***Network forensics deals with data that disappears quickly. Once network traffic flows through the system, it's gone — unless someone captures it first.***

Think about it this way: when you log into Facebook, HTTP traffic is generated. When you SSH into a server, that creates SSH traffic. All of this network activity is happening right now, and the moment it passes through the system it's gone forever unless an investigator captures it in real time. That's why network forensics investigators need to collect and analyze this information while it's actually happening.

For example, imagine you're working at a company. Traffic is flowing in and out HTTP requests, HTTPS sessions, login attempts. This traffic contains real-time evidence of everything happening on the network. Once it's gone, it's gone. So we need tools like Wireshark or tcpdump to capture that traffic before it disappears.

Why Do We Conduct Network Forensics?

Network forensics serves two primary purposes. Let's go through both carefully.

Purpose 1: Security MonitoringThe first purpose is security. Network forensics helps experts watch network traffic to identify suspicious activity and unauthorized access. This is especially critical because and this is something really important to understand

> **Hackers almost always delete their tracks. After they compromise a computer, server, or any system, they remove the logs and evidence they left behind. That means the only remaining proof of the attack might be the captured network traffic.**

For example: suppose you see an unfamiliar IP address making repeated HTTP requests to your organization's system. That's suspicious. Or you see someone sending a flood of HTTP requests that suggests a brute-force attack attempt. The captured network traffic is your evidence even if the attacker deleted everything on the compromised machine.

Purpose 2: Law Enforcement & Investigation

The second purpose is law enforcement and post-incident investigation. Here, investigators look at all the files and details that moved over the network. They can piece together communications emails, chat messages and understand exactly what the attacker did, how they did it, and what data they accessed or stole.

How Do We Conduct Network Forensics?

Network forensics can be applied in two distinct ways, depending on when you're doing it relative to an incident.

Approach 1: Proactive Network Forensics (Before the Incident)

This is about continuous surveillance and analysis of network data to detect threats before they cause damage. Think of it as ongoing monitoring using threat intelligence.

How does it work in practice? Imagine you're working at a bank. You know that APT29 a known hacker group typically uses certain IP addresses, certain domains, certain signatures. These are called network indicators. Your job is to continuously monitor for those indicators on your network.

Network indicators include:

- IP addresses associated with known threat actors
- Suspicious domain names
- Known malware signatures
- Malicious URLs

If you spot one of those indicators, you can block it immediately before any attack actually happens. That's the power of proactive forensics. You're not waiting to be attacked; you're actively watching for warning signs.

Approach 2: Reactive (Post-Incident)

This is what you do after an incident has already occurred. The goal here is to understand what happened and reconstruct the full attack timeline. In network forensics, you examine all stages of the attack:

- **Reconnaissance** how did they gather information about us?
- **Delivery** how did they deliver the malware?
- **Exploitation** how did they exploit the system?
- **Installation** how did the malware install itself?

- **Command and Control (C2)** how did the attacker communicate with the infected system?
- **Post-exploitation / Data Exfiltration** what data did they steal and how?

All of these stages involve network activity. And if you captured the traffic, you can reconstruct every step the attacker took.

What is PCAP?

PCAP is a file format that stores the network packet data collected from a network interface. What's inside a PCAP file? It contains a complete copy of every byte of every packet observed on the network covering data from OSI Layer 2 all the way up to Layer 7. It also includes timestamps, so you can see exactly when each packet was captured. That makes PCAP files extremely useful for in-depth analysis and troubleshooting.

Packet Capture File Formats

pcap (Packet Capture)

The classic pcap format, developed by the tcpdump team, is the most widely adopted standard. It stores captured network traffic in binary format and preserves all packet details: headers, payloads, and timestamps. Virtually every network analysis tool supports pcap files it's the universal language of packet capture.

pcapng (Next Generation pcap)

The newer pcapng format was designed to overcome the limitations of the original pcap. It supports advanced features like capturing metadata from multiple interfaces simultaneously, adding comments, and storing extra context. However, some older tools still don't support pcapng, so you'll sometimes need to convert

between the two formats. Wireshark makes this easy it can convert capture files from one format to the other.

> *Key difference: pcap captures one interface at a time. pcapng can capture multiple interfaces simultaneously which is a massive advantage when you need visibility across different network zones at the same time.*

Quick conversion command:

```
editcap -F pcap input.pcapng output.pcap
```

Why Are PCAPs So Important?

Compared to every other data source available to a forensics analyst, PCAP is the most comprehensive. It captures every single byte traveling through the network. That granularity is invaluable in two critical scenarios:

Data Exfiltration Cases

When an attacker is trying to steal data from your organization — say they've infiltrated Amazon and are exfiltrating sensitive data to their C2 server — the PCAP file lets you identify exactly which data was taken. You can see what type of data it was, what size, where it went. This is concrete, legally admissible evidence of the breach.

Command and Control (C2) Communications

When communications are unencrypted, PCAP files can reveal the exact commands the attacker was executing on your system. You can literally see what they told the malware to do. This gives you complete visibility into the attacker's methods and intentions.

Drawbacks of PCAPs

PCAPs are powerful, but they come with real limitations that you need to be aware of as an analyst.

1. Large File Sizes

PCAP files can grow enormously, especially in high-traffic networks. Think about this: on a 1 Gbps link, you can generate tens of terabytes of data per day. Where do you store all of that? This is one of the biggest practical challenges of network forensics at scale the storage costs alone can be massive.

2. Processing Overhead

The sheer volume of data makes it impractical to work with PCAPs directly at scale. Analysts typically need to convert or distill PCAPs into more manageable formats before they can do useful analysis.

3. Legal and Privacy Concerns

Capturing all network traffic raises serious legal and privacy issues especially in regions with strict data protection laws like GDPR. When you capture traffic, you're potentially capturing people's private communications, passwords, personal data. You need to handle PCAPs very carefully and make sure you're operating within legal boundaries.

4. Encryption

Modern traffic is increasingly encrypted. HTTPS, for example, means you can see that traffic is happening, but you can't see what's inside the packets. This significantly limits the usefulness of PCAP in environments that heavily use encryption. You can see the connection, but not the content.

Network Forensics Tools: Choosing the Right One

This is where many people get confused they think it's all about the tools. But let me tell you something important: the tools are only 20-30% of this work. The other 70-80% is thinking and problem solving. The market is flooded with courses that focus on tools, and that's exactly why there are three million unfilled cybersecurity jobs because people can use tools but can't think through a problem.

That said, you do need to know your tools. Each one has a purpose. Let's go through them:

Wireshark

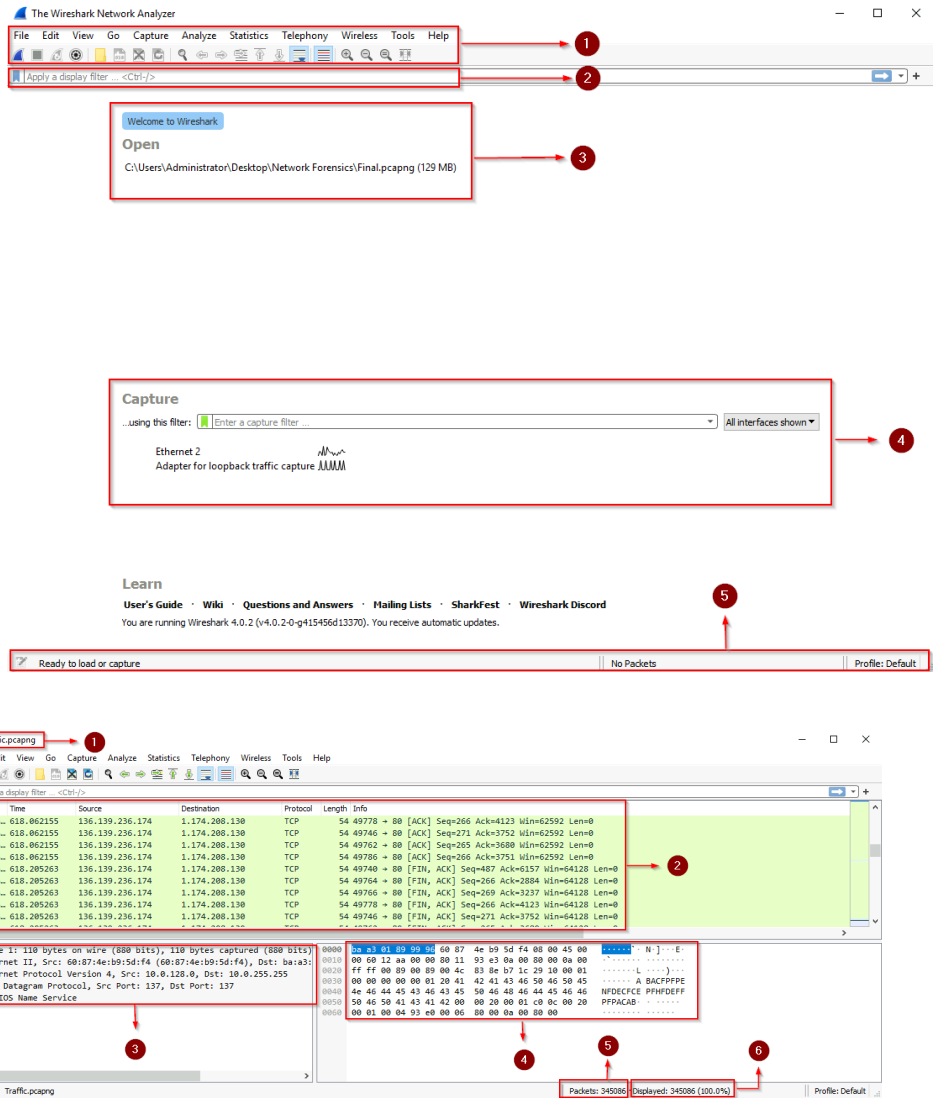
Wireshark is the most widely used network analysis tool. It gives you a graphical interface to capture and analyze packets in real time. Here's what it's great at:

- Capture and deep packet analysis — you can see every header, every payload
- Protocol dissection — it decodes and interprets hundreds of network protocols
- Powerful filtering and searching — find exactly the traffic you need
- Visualization and reporting — charts, stats, conversation logs
- Compliance and auditing support

> **Limitation: Wireshark struggles with very large PCAP files (above ~200MB). It slows down significantly and becomes impractical for large-scale analysis.**

The Wireshark interface has five key areas:

1. Toolbar — quick access to capture, export, and analysis functions
2. Display Filter Bar — filter which packets you see
3. Recent Files — quickly reopen previous captures
4. Capture Filter and Interfaces — choose which network interface to monitor
5. Status Bar — shows details about the currently loaded file



Zeek

Zeek is a network security monitoring tool that's excellent at reading PCAP files and their associated logs. Unlike Wireshark, Zeek works from the command line no graphical interface. This makes it less beginner-friendly, but much more powerful for large-scale or automated analysis.

> **Zeek's strength: it can handle large files that would bring Wireshark to its knees. It also produces structured logs for each protocol HTTP logs, SSL logs, DNS logs making correlation much easier.**

Limitation: No GUI. You're working purely in the terminal.

Brim (with Zed)

Brim is the best of both worlds. It has a graphical interface AND can handle large files. It uses Zed (the query language) and integrates Zeek under the hood. This makes it the go-to tool when you need visual analysis of large PCAP files.

Brim's key advantages:

- Handles large PCAP files (200MB+) efficiently
- Modern packet processing engine for fast file loading
- Zed query language for precise, powerful searches
- Built-in Zeek integration — automatically generates Zeek logs from PCAPs
- Suricata integration for IDS/IPS alert correlation
- Visualization — time series graphs of network traffic
- One-click export to Wireshark for deep packet inspection

Network Miner

Network Miner takes a completely different approach. Instead of capturing raw packets like Wireshark or Brim, it analyzes traffic and gives you a summary. It doesn't store the full packet data instead it extracts meaningful information and presents it as a digest.

> Advantage: Massively reduces storage requirements. Instead of terabytes of raw packets, you get a compact summary of what happened on the network. Also useful for getting visibility across multiple network locations without storing everything.

Trade-off: You lose the granular packet-level detail that a full PCAP provides.

When to Use Which Tool: The Decision Framework

A question that comes up a lot: "If these tools all do similar things, why use different ones?" Great question and the answer is that each tool is optimized for a specific scenario.

- Use Wireshark when you need deep packet inspection examining individual headers, payloads, and low-level protocol behavior on a reasonably-sized file.
- Use Zeek when you're working command-line, need automated log generation, or are integrating with scripts and pipelines.
- Use Brim when you have large files and need a visual interface it's the sweet spot of Wireshark's GUI and Zeek's power.
- Use Brim + Suricata when you want to correlate IDS/IPS alerts with network traffic so your analysts can quickly spot and investigate the serious events.

Example scenario: You have a large enterprise with Suricata or Snort generating alerts. Instead of having your analysts swim through thousands of raw log lines, Brim lets you see all those alerts in a clean visual interface, correlate them with actual packet data, and drill down into the ones that matter.

Brim (Zed): A Closer Look at the Interface

Let's go through the Brim interface in detail, because this is what you'll be using for your practical tasks.

Main Interface Panels

- Left Panel — Contains queries, history, and data type views
- Main Panel — Displays query results and event listings
- Right Panel — Detailed view of selected events
- Visualization — Time series graph of network traffic over time

The Zed Query Language

Zed queries are how you interrogate your data in Brim. Think of them like commands. They're much more powerful and specific than Wireshark display filters.

Example queries:

```
event_type == "alert"
```

This shows only Suricata alert events — filtering out all regular traffic so you can focus on what matters.

```
event_type == "alert" | count by alert.severity
```

Groups and counts alerts by severity level — gives you a quick picture of how serious the threats are.

Correlating with Wireshark

One of the best features of Brim is that from any log entry or event, you can click to open that specific traffic directly in Wireshark for deep packet inspection. This gives you the best of both tools Brim's speed and visualization, Wireshark's packet-level detail.

Malicious URLs: Understanding the Threat

Now let's shift to another critical topic that feeds directly into network forensics work: malicious URLs. These have become one of the primary attack vectors in modern cybersecurity.

Why Are URLs So Dangerous?

Unlike email attachments which can be flagged and blocked relatively easily URLs are dynamic. A URL can look completely safe at one moment and redirect to something malicious the next. And consider this data point from 2018: while attachment-based attacks decreased by 22%, URL-based attacks increased by 31%. Attackers shifted their methods because URLs are harder to defend against.

Types of Malicious URLs

1. Phishing URLs

Attackers create look-alike login pages to steal credentials. The URLs are crafted to resemble legitimate domains. Classic example: an email that looks like it's from Microsoft Support, asking you to verify your account. The link looks real, but it redirects you to a fake login page.

2. Web Downloader URLs

These URLs point to a file hosted on a cloud service or compromised server. When clicked, they download malware directly to your machine. The link might appear to be downloading a PDF but it's actually downloading a malicious executable.

3. Exploit Kit URLs

Exploit kits are automated tools that probe your browser for vulnerabilities and deliver a payload if they find one. Attackers often host these on compromised legitimate websites so you visit what looks like a normal blog, and the exploit kit silently scans your browser for weaknesses.

How Attackers Bypass URL Filters

- Encoding and Obfuscation — URL-encoding or Base64-encoding parts of the URL to fool automated scanners
- URL Shortening — using Bitly or TinyURL to hide the real destination
- Dynamic Redirection — the URL changes its destination after passing initial security checks
- Domain Shadowing — hijacking a legitimate domain's DNS to create malicious subdomains
- Disposable Domains — newly registered domains that haven't been blacklisted yet
- HTTPS Abuse — using HTTPS encryption to hide malicious content from inspection
- Cloud Hosting — uploading malware to trusted platforms like Google Drive or Dropbox so the URL appears legitimate

Defense Strategies for Malicious URLs

- URL Disarm/Rewrite — Secure Email Gateways rewrite HTTP to hxxp, making links non-clickable so users must manually inspect them
- URL Reputation Filters — automatically score URLs using threat intelligence feeds, adding warning tags to suspicious emails
- URL Sandboxing — analyze URLs in an isolated environment to detect malicious behavior before they reach users
- Click-Time Evaluation — inspect URLs both when the email arrives AND when the user clicks, catching delayed redirections
- Remote Browser Isolation (RBI) — render web content in a cloud-based environment so malware can't reach the user's device
- DNS Sinkholing — redirect malicious domain requests to a controlled server
- Content Disarm and Reconstruction (CDR) — strip malicious content from emails while preserving the original appearance

The Right Mindset for Cybersecurity Work

Let me close with something that's more important than any tool or technique we've covered today. Cybersecurity is 70-80% thinking and problem-solving. The tools are just a means to an end.

Companies like Amazon have entire teams dedicated to network forensics and packet analysis. When they interview candidates, they don't ask "Which tool did you use?" They ask "How would you think through this problem?" Someone with three years of experience who's been doing routine operations might be outperformed by someone with one year of experience who has genuine hands-on project work and strong analytical thinking.

> ***Experience is not just about time spent working — it's about the depth and variety of problems you've solved. A strong portfolio of real-world projects is your most powerful credential.***

Focus on building your problem-solving skills. Learn the tools, yes — but always ask yourself: "Why am I using this tool? What am I looking for? What would it mean if I found it?" That kind of thinking is what separates good analysts from great ones.

Alright, before we get into today's topic — I want to address something. The previous lectures didn't have solutions attached to them, and that was intentional. Those lectures were basically configuration guides, extra content for your tasks. But then some of you said it wasn't clear enough or it was too difficult, so I removed them. We'll finish the main material first, and then we can add the extras later.

So we have about 17–18 days left, and we should be covering roughly 5 lectures in that time — two per week. We're going to work through this properly, no gaps, no missed topics. We'll make up those days. The goal is to cover everything, and by the end we'll have you fully prepared.

Malicious URLs in Email Security

what they are, how attackers use them, and how to defend against them.

What Is a URL?

Let's start from the beginning. A URL you all know what a link is. It usually starts with the protocol, which is either HTTP or HTTPS. Then after that comes the domain, and sometimes a subdomain before the domain. Then the TLD the top-level domain which could be .com, .net, .org, and so on.

For example, take Facebook. A URL for Facebook would look like: https://www.facebook.com. Here, HTTPS is the protocol, 'www' is the subdomain, 'facebook' is the domain, and '.com' is the TLD. Simple enough.

Why Are URLs Dangerous?

So what's happening right now with attacks? The attackers have shifted they're exploiting URLs heavily because it's much more effective than attachments in many ways. Let me give you a real-world example.

Imagine you get an email that says it's from Facebook or Meta. It tells you: 'We charged your ad account $67.' You look at that and you're like, wait $67? That doesn't seem right. And then the email says: 'If you have an issue, click here.' Or maybe it says your ad account was suspended.

Now you're emotional. You react. You click the link.

But when you hover over or copy that link, you notice it doesn't say facebook.com it says something like faceb00k.com, or facebok.com. The attacker registered a fake domain that looks almost identical to the real one. And because you're in an emotional state, you didn't even notice.

That's the attack. That's how it works. And it's way more dangerous than attachments because a link can look completely normal and redirect you anywhere.

Example phishing email pretending to be from Instagram claiming someone logged in from an unknown location. There was a 'Secure Your Account Here' button. The link appeared to go to instagram.com, but it was fake.

Types of Malicious URL Attacks

1. Phishing URLs

This is the most common one. The attacker sends you an email that looks legitimate it could be from your company's helpdesk, or from Microsoft telling you to verify your email or fix an issue. You click the link, and it takes you to a phishing page designed to steal your credentials.

The URL might look like it belongs to Microsoft or your company, but it's actually a fake domain that the attacker registered.

2. Web Downloader URLs

These URLs redirect you to a file hosted on the cloud like Amazon S3 or Google Drive. The email tells you there's an important PDF or update you need to download. You click it, and instead of going to a real website, you get redirected to download a malicious file.

The reason attackers use legitimate cloud services like Google Drive or Amazon is because your security tools won't automatically block them. The domain looks trusted.

3. Exploit Kit URLs

This one is more technical. When you click the link, it doesn't just take you to a phishing page it actually scans your browser and your system. It checks what software you have installed, what version of Java you're running, what browser you're using.

If it finds a vulnerability say, you're running an old version of Chrome or Firefox it delivers a payload specifically crafted for that vulnerability. This is called an exploit kit.

Attackers compromise legitimate websites and install these exploit kits. So you might visit a normal blog and get infected without even downloading anything. The browser itself gets exploited.

How Attackers Bypass URL Security Controls

You might be thinking okay, but don't we have security tools that block malicious URLs? Yes, we do. But attackers have figured out ways to bypass them. Let's go through the main techniques.

1. URL Encoding

The attacker takes the malicious URL and encodes it for example, in Base64. So instead of the URL looking like a normal link, it becomes a string of random characters that your security tool can't recognize or match against its blocklist. The link still works when decoded, but the security filter doesn't catch it. This is why sometimes even WAFs (Web Application Firewalls) can be bypassed with encoding tricks.

2. URL Shorteners

You know sites like bit.ly or TinyURL? They shorten long URLs into something like bit.ly/a3x. Attackers use these to hide where the link actually goes.

For example, the attacker has a command-and-control server, or a malicious site. Instead of putting that URL directly in the email where your security tool might flag it they run it through a URL shortener. Now the link looks completely innocent. The victim clicks it, gets redirected to the real malicious destination, and the security filter never saw what it was pointing to.

3. Open Redirects

This is a clever one. You have a legitimate, trusted website like a big corporate site that has a redirect feature. The attacker crafts a URL on that trusted domain that redirects to their malicious site.

So when the security tool scans the link, it sees a trusted domain, lets it through, and then the redirect happens after the email reaches the inbox.

4. Subdomain Abuse

The attacker compromises a legitimate website and creates a malicious subdomain under it. For example, if they hack a site called company.com, they might create login.company.com which points to their phishing page. The main domain looks real, so security tools might miss it.

They also manipulate DNS records they change the DNS so that anyone who types in a legitimate domain gets redirected to the attacker's server instead.

5. Newly Registered Domains

Attackers register brand-new domains that look similar to legitimate ones. For example, instead of facebook.com, they register facebo0k.com or face-book.com.

The reason this bypasses security is that reputation-based filters rely on historical data. A brand new domain has no reputation it hasn't been flagged yet. So it passes through email security filters. Once it gets blocked, the attacker registers a new one. It's an ongoing cycle.

6. HTTPS Inspection Bypass

Most modern traffic is encrypted with HTTPS. Some security tools don't inspect encrypted traffic they just see that it's HTTPS and assume it's fine. Attackers use this to hide malicious content inside encrypted connections that never get inspected.

7. Cloud Service Hosting

As mentioned earlier, attackers host malicious content on Google Drive, Dropbox, Amazon S3, or other cloud platforms. When you click the link and download the file, it comes from a trusted cloud domain. Security tools don't block Google Drive or Amazon, so the malicious file gets through without any alerts.

Dynamic Analysis Tool: Any.run

Now that we understand how these URL attacks work and how attackers bypass defenses, let's talk about how we investigate suspicious links and files.

The main tool we're going to use is called Any.run. Think of it as an interactive sandbox it runs a link or a file inside a controlled virtual environment and shows you exactly what happens in real time.

What Does Any.run Do?

- Opens suspicious URLs inside a virtual browser in the cloud — so your real machine never touches it
- Provides detailed behavioral analysis — it tracks every action the file or link performs
- Shows network requests — what domains or IPs does the malware try to connect to?

- Shows file system changes — what files does it create, modify, or delete?
- Shows registry activity — what registry keys does it write to?
- Captures screenshots — so you can see exactly what the victim would have seen
- Generates a full report with a network graph and correlation of all suspicious activities

How to Use Any.run

The workflow is straightforward:

1. Submit the suspicious email, file, or URL to Any.run
2. It runs the content inside a sandboxed virtual machine (you can choose the OS — Windows 7, 10, 11, Android, Linux)
3. Watch the behavior in real time — see what processes start, what connections are made, what files are created
4. Review the full report — static info, network events, screenshots, process tree, registry changes
5. Export the report or share it with your team

Any.run opened Microsoft Edge inside the sandbox, navigated to the URL, and captured everything that happened what processes spawned, what network requests were made, what the page looked like. The full report showed the MITRE ATT&CK tactic being used (technique T1112), the process tree, and all file activity. It also showed the Outlook process opening and interacting with the link.

Another example: a link that downloads an Adobe Reader installer. Any.run downloaded the file and executed it in the sandbox. The report showed the installer

process starting, what files it created, registry activity, and screenshots of every step. Everything is recorded.

Note Any.run has a public submissions view where you can see what other people are analyzing right now. If you want to analyze something private, use the private analysis mode don't submit sensitive files publicly.

Defense Strategies Against Malicious URLs

Okay, now let's flip it. We've talked about how attackers work. Now how do we defend against them?

1. URL Rewriting (Safe Links)

configure this in your SEG (Secure Email Gateway) or your email security solution.

What URL rewriting does: before the email reaches your inbox, the security system rewrites every hyperlink inside it. Instead of the original link, it converts it to HXXP (the safe notation). This forces the user to manually copy-paste the link into their browser they can't just click it.

More importantly, it routes the click through the security platform first. So when someone clicks a link, the security system checks it in real time before the user actually reaches the destination. If it's malicious, it blocks it. If it's clean, it lets the user through.

2. Reputation Filters

These filters automatically score URLs based on reputation. Using services like RiskIQ or similar threat intelligence platforms, every domain gets a score from

0 to 10. Based on that score, the system decides whether the link is safe, risky, or dangerous.

This is important because it's automated your security team doesn't have to manually review every link. The system does it and acts accordingly.

3. Sandboxing

This is what Any.run does but you can implement it at the organizational level. Any link that comes through email gets automatically detonated in a sandbox before reaching the user. If it's malicious, it gets blocked. If it's clean, it's delivered.

4. Click-Time Evaluation

This is different from just scanning the link when the email arrives. Because attackers know that links get scanned at delivery time, they sometimes make the link harmless at first and then switch the content to malicious later.

Click-time evaluation means the link is re-scanned every time someone actually clicks it. So even if the link looked clean when it arrived, if it becomes malicious by the time you click it, the system catches it.

5. Remote Browser Isolation (RBI)

This is a newer technology and one you can actually implement in your organizations. Here's how it works: instead of opening the link in your local browser, the browser runs in the cloud or on an isolated server. You see the result the webpage but the actual execution happens somewhere else.

Platforms like Cisco Umbrella, Cloudflare, and others offer Remote Browser Isolation. Even if the page is malicious, the malware executes in the isolated environment and never touches your machine.

6. Hover Preview

Simple but effective — train your users to hover over links before clicking them. The URL shown in the status bar or tooltip reveals the actual destination. If what's written in the email doesn't match what appears in the hover preview, that's a red flag.

7. Email Security Gateway (SEG) Best Practices

- Make sure your SEG is configured with the best possible settings — don't leave defaults
- Enable advanced threat intelligence features
- Always enforce multi-factor authentication (MFA) for email access
- Configure DMARC, DKIM, and SPF on your domain — we covered DNS security before, this is critical
- Monitor DNS and HTTP traffic continuously — watch for unusual patterns
- Practice SOC exercises regularly — tabletop exercises, incident response drills
- Train your users — awareness is still one of the most important defenses
- Use CDR (Content Disarm and Reconstruction) — this strips potentially malicious content from emails and keeps only the safe parts

SAMMRY

URL-based attacks are generally more dangerous and more difficult to detect than attachment-based attacks. Attachments can often be identified and blocked by email security systems because suspicious file types are easier to recognize and quarantine.

However, URLs are far more flexible. They can be modified, redirected, hidden inside shortened links, or even hosted on trusted platforms. Because of this adaptability, attackers rely heavily on URLs to bypass traditional defenses.

To achieve this, they use techniques such as encoding, URL shortening, open redirects, abusing subdomains, registering new domains, leveraging HTTPS encryption, and hosting malicious content on cloud services.

From a defensive perspective, protection relies on multiple mechanisms including URL rewriting, reputation-based filtering, sandbox analysis, click-time inspection, remote browser isolation, and properly configured email security solutions.

Malicious Attachments — Entry Point of Attacks

A malicious attachment is a file sent to a victim, typically through email, with the intention of executing harmful code once opened.

Attackers frequently use Microsoft Office files such as Excel and Word because they are widely trusted and commonly used in organizations. These files can include macros or embedded scripts that automatically execute when opened by the user.

There are two fundamental concepts to understand. An exploit targets a vulnerability in software to directly execute malicious code. In contrast, a dropper does not attack immediately but instead installs or delivers the actual malware after execution.

In simple terms, an exploit breaks into the system, while a dropper delivers the payload. This is why even a single click on an attachment can lead to full system compromise.

Email Security — Understanding SPF, DKIM, and DMARC

Email authentication is built on three core technologies, each addressing a different aspect of trust.

SPF verifies whether the sending server is authorized to send emails on behalf of a domain. It focuses only on the source of the email, specifically the IP address, but it does not validate the content or the visible sender.

DKIM, on the other hand, ensures that the content of the email has not been altered. It uses a cryptographic signature that allows the receiver to verify that the message was signed by the legitimate domain.

DMARC adds a critical layer by addressing spoofing. Even if SPF and DKIM both pass, an attacker can still fake the visible “From” address. DMARC enforces alignment by checking whether the “From” domain matches either SPF or DKIM. If alignment fails, the email is considered suspicious.

This makes DMARC essential for preventing impersonation attacks.

Malware Analysis — Safe Investigation Methods

When dealing with suspicious files, direct execution is unsafe. Instead, analysts rely on sandbox environments. A sandbox is an isolated system designed to execute files safely while monitoring their behavior.

Within this environment, analysts can observe actions such as system modifications, network connections, and process creation.

However, modern malware is often designed to detect sandbox environments and suppress its behavior, making detection more challenging.

VirusTotal — Benefits and Risks

VirusTotal is widely used to analyze files by scanning them across multiple antivirus engines. It helps determine whether a file is known to be malicious and provides insight into detection rates.

Despite its usefulness, there is an important risk. Uploading files to such platforms may expose them publicly, potentially revealing sensitive or proprietary information.

Defense Strategies — Building Strong Protection

Security should never rely on a single solution. The concept of defense-in-depth emphasizes the use of multiple protective layers such as antivirus tools, email filtering systems, endpoint detection and response (EDR), and network monitoring.

This layered approach ensures that if one control fails, others remain in place to mitigate the threat.

Another important concept is the difference between whitelisting and blacklisting. Blacklisting blocks known malicious files, whereas whitelisting allows only approved and trusted files. Whitelisting is generally more effective because it blocks unknown threats by default.

Content Disarm and Reconstruction (CDR) takes a different approach by removing potentially harmful components from files and rebuilding a safe version. Instead of detecting malware, it eliminates the possibility of malicious content entirely, making it particularly effective against zero-day threats.

Malicious URLs — Advanced Attack Techniques

Malicious URLs present a greater challenge than attachments because they are dynamic and can change behavior over time.

Attackers use various techniques to disguise their intentions. Obfuscation hides the true destination of a link, while URL shorteners mask the final address.

Redirection chains lead users through multiple steps before reaching the malicious site, and domain shadowing allows attackers to exploit legitimate domains.

The primary objective of these techniques is to steal user credentials or deliver malware.

What makes URLs particularly dangerous is that they may appear safe initially but change after the user interacts with them, making detection significantly more difficult.

Advanced Web Protection Mechanisms

To counter these threats, modern security systems use advanced techniques. URL rewriting modifies links so they can be monitored. Click-time evaluation checks URLs at the moment the user clicks them, rather than when the email is received.

Remote Browser Isolation (RBI) further enhances security by executing web activity in a remote, isolated environment, preventing direct interaction with the user's system.

These approaches focus on real-time threat detection rather than relying solely on pre-delivery scanning.

Network Forensics — Investigating Attacks

Network forensics involves analyzing network traffic to understand attacker behavior and identify malicious activity.

There are two main approaches. Proactive forensics involves continuous monitoring to detect threats early, while reactive forensics focuses on investigating incidents after they occur.

Packet Capture (PCAP) is a key tool in this process. It records full network traffic, allowing analysts to reconstruct attacker actions and identify data exfiltration.

However, PCAP analysis comes with challenges. Files can be large and complex, and encryption can obscure the actual content of communications. Additionally, network traffic is highly volatile, meaning it can disappear quickly if not captured in time.

Real-World Attack Pattern

Modern attacks often combine multiple evasion techniques. For example, an attacker may send a password-protected attachment to bypass scanning, include a shortened URL to hide the destination, and rely on the absence of DMARC to spoof the sender.

The malware may initially appear inactive in a sandbox environment but activate later during real execution.

This combination of techniques demonstrates how attackers evade detection by using layered strategies.

A security analyst must approach every situation with skepticism. It is essential not to trust appearances, but instead verify sources and analyze behavior.

Effective defense requires combining multiple tools, continuously monitoring activity, and focusing on how systems behave rather than relying solely on static indicators.