Cybersecurity attacks explained: a technical guide to how attackers actually work

Francesco Barbato · Full Stack · Cybersecurity · AI

Published 12 March 2026 · 22 min read

ABSTRACT

Most breaches don't start with a sophisticated exploit: they start with a phishing email, a reused password or an unpatched server. A technical tour of how attacks really work, from the attacker's lifecycle and social engineering to SQL injection, ransomware-as-a-service and supply chain compromises, with real cases and a layered defence framework.

Keywords: Cybersecurity · Hacking · Phishing · Malware · Ransomware · Network security · Web security · Social engineering · Supply chain

Cybersecurity is no longer a topic for IT departments and security specialists alone. It belongs to every developer who ships code, every manager who approves a software purchase, every employee who handles credentials and every executive who signs off on infrastructure. The threat landscape has matured, professionalised and scaled to the point where ignorance is no longer a defensible position for anyone who builds or operates digital systems.

This article is a technical primer: the main attack categories, how each works at a mechanical level, real examples and what effective defence looks like. The aim is understanding, not just vocabulary.

1 How attackers think: the attack lifecycle

Serious attacks, whether run by states, organised criminal groups or skilled individuals, tend to follow a recognisable lifecycle. Knowing it explains why no single defence is enough and why defence-in-depth is the only coherent strategy.

The most widely used reference is the **MITRE ATT&CK** framework, which catalogues adversary tactics, techniques and procedures observed in the real world. For an introduction, a simpler linearisation is enough:

1. **Reconnaissance**: the attacker gathers information on the target: domain names, IP ranges, employee names and emails, technology stack, exposed services, social media presence. Much of it is passive and leaves no trace in your logs.

2. **Initial access**: the first foothold: a phishing email that delivers malware, an exploited vulnerability in a public-facing service, stolen credentials used on a login portal, or a supply chain compromise.
3. **Persistence**: the attacker makes sure they keep access even if the initial vector is closed: a backdoor, new user accounts, modified startup scripts, command-and-control (C2) infrastructure.
4. **Privilege escalation**: moving from a limited foothold to higher privileges, by exploiting a local vulnerability to become root or SYSTEM, abusing misconfigured permissions or stealing credentials from memory.
5. **Lateral movement**: moving through the network, compromising more systems and heading toward high-value targets.
6. **Exfiltration or impact**: the objective: stealing data, deploying ransomware, disrupting operations or establishing long-term espionage access.

> **THE DWELL TIME PROBLEM**
>
> The time between an attacker's initial access and detection, the dwell time, has fallen sharply: Mandiant's M-Trends reports put the global median at around 10 days for 2023 (down from 16 the year before), against more than 200 days a decade ago. Still, it means days in which the attacker moves through your systems before anyone notices. Most defences are built to prevent initial access; far fewer are built to detect and respond to an attacker who is already inside.

2 Social engineering

The best technical defences can be bypassed by phoning an employee and asking them to reset a password. Social engineering exploits human psychology rather than technical flaws, and it is consistently among the most common initial access vectors in real breaches.

2.1 Phishing

Phishing means sending fraudulent communications, most often emails, that look like they come from a trusted source, to trick the recipient into revealing credentials, clicking a malicious link or opening malware.

A typical phishing email impersonates a known service (Microsoft, Google, a bank, internal IT) and creates urgency: “Your account has been compromised. Verify your identity within 24 hours.” The link leads to a convincing copy of the real login page, on a domain that differs by one character or uses a homoglyph attack (a Latin “o” replaced by a visually identical Cyrillic “o”).

Modern phishing infrastructure is sophisticated. **Adversary-in-the-middle (AiTM) phishing kits**, such as Evilginx2 and Modlishka, work as a reverse proxy between victim and real site: the victim authenticates against the real service through the attacker's proxy, the attacker captures the session cookie in real time and multi-factor authentication is bypassed entirely, because the session is already authenticated.

2.2 Spear phishing

Where standard phishing is a mass campaign, spear phishing is targeted. The attacker studies a specific person (role, colleagues, recent projects, writing style) and writes a highly personalised message. An email that seems to come from the CFO asking the accounts payable manager to process an urgent wire transfer is far more convincing than a generic “your account is locked”. Spear phishing aimed at CEOs, CFOs and board members is called **whaling**: the effort is higher, and so is the payoff.

2.3 Vishing and smishing

Vishing (voice phishing) uses phone calls: attackers pose as IT support, a bank's fraud department, a public agency or a colleague. **Smishing** (SMS phishing) exploits the higher trust people tend to give text messages than email: a text from a “courier” with a tracking link, or from a “bank” about a suspicious transaction.

2.4 Business Email Compromise (BEC)

BEC is a sophisticated social engineering fraud that has cost organisations more than any other category of cybercrime: the FBI reports over $55 billion in losses worldwide between 2013 and 2023. The attacker either takes over a legitimate email account or convincingly spoofs one, then uses it to authorise fraudulent transfers or divert legitimate payments.

> **CASE STUDY: THE TWITTER HACK (2020), SOCIAL ENGINEERING AT SCALE**
>
> In July 2020 attackers took control of high-profile Twitter accounts, among them Barack Obama, Joe Biden, Elon Musk, Apple and Uber. It started not with a technical exploit but with phone calls to employees: the attackers impersonated internal IT support, obtained credentials and used them to reach Twitter's internal admin tools. From there they took over the target accounts and posted a cryptocurrency scam that collected roughly $120,000 in Bitcoin before it was shut down. Three people were charged, and the alleged mastermind was 17 at the time. The lesson: no technical hardening protects you from an employee who has been talked into handing over their access.

3 Credential attacks

Usernames and passwords are still the main way to authenticate almost everywhere, and the main target of attackers, because stealing valid credentials is easier and more reliable than exploiting vulnerabilities.

3.1 Credential stuffing

It exploits a simple and persistent human habit: password reuse. When a large database of username and password pairs leaks from one service (and billions are circulating on underground markets) attackers use automated tools to test the same pairs against other services at scale. A meaningful share of users reuse passwords, and distributed infrastructure lets attackers test millions of pairs in hours while evading rate limiting and IP blocking. Tools such as Sentry MBA, SNIPR and OpenBullet are commodity software on criminal forums.

3.2 Password spraying

Where credential stuffing tries many passwords against many accounts, password spraying tries a few common ones (“Password1”, “Welcome123”, “CompanyName2024”) against a large number of accounts. It deliberately stays below lockout thresholds: instead of hammering one account, it makes one guess on thousands of accounts and then waits before the next. It is particularly effective in Active Directory environments with per-account lockout policies: one guess per account per hour across ten thousand accounts produces plenty of attempts without tripping any individual lockout.

3.3 Pass-the-hash and pass-the-ticket

These are post-exploitation attacks at the Windows authentication layer. When a user authenticates, the password hash or Kerberos ticket is kept in memory by the LSASS process. An attacker with SYSTEM privileges can extract it with tools such as Mimikatz and use it to authenticate as that user on other systems, without ever knowing the plaintext password. **Pass-the-hash** works against NTLM authentication, **pass-the-ticket** against Kerberos. Both are standard in lateral movement in Windows environments.

> **WHY MFA IS NOT A SILVER BULLET**
>
> Multi-factor authentication raises the cost of credential attacks a lot, but doesn't eliminate them. AiTM phishing bypasses it by stealing post-authentication session cookies. SIM swapping lets attackers receive SMS codes. Push-fatigue attacks flood users with approval requests until they approve one by mistake or out of exhaustion. MFA is essential, and it is not sufficient on its own.

4 Network attacks

Network-layer attacks target the infrastructure data travels through, rather than the applications or users at either end.

4.1 Man-in-the-Middle (MitM)

A MitM attack puts the attacker between two communicating parties so that they can intercept, read and possibly modify traffic without either noticing. Several techniques achieve that position:

- **ARP poisoning** abuses the Address Resolution Protocol, which maps IP addresses to MAC addresses on a local network. With forged ARP replies the attacker makes other hosts associate the attacker's MAC with a legitimate IP, routing traffic through their machine.
- **DNS spoofing** corrupts DNS cache entries to redirect lookups to attacker-controlled addresses: a user typing “bank.com” lands on the attacker's server.
- **SSL stripping** downgrades HTTPS to HTTP. The attacker intercepts the redirect from HTTP to HTTPS and keeps an HTTP connection with the user while proxying HTTPS to the real server. HSTS (HTTP Strict Transport Security) is the main defence.

4.2 Denial of Service (DoS) and Distributed DoS (DDoS)

A DoS attack overwhelms a system with traffic or requests until it can't serve legitimate users. DDoS spreads the attack across thousands or millions of compromised machines (a botnet), so it can't be stopped by filtering a single source IP. Modern DDoS works at several layers:

Table 1

Type	Layer	Mechanism	Examples
Volumetric	Network (L3/L4)	Floods the bandwidth with raw traffic	UDP flood, ICMP flood, DNS amplification
Protocol	Transport (L4)	Exhausts the server's connection state tables	SYN flood, Ping of Death, Smurf
Application	Application (L7)	Exhausts application resources with legitimate-looking requests	HTTP flood, Slowloris, ReDoS

Amplification attacks deserve a mention. By spoofing the source IP of a small request to a service that returns a much larger response (DNS, NTP, memcached), an attacker can direct huge volumes of traffic at a target with very little outbound bandwidth. The 2018 attack on GitHub peaked at 1.35 Tbps through memcached amplification, the largest ever recorded at the time.

4.3 Port scanning and enumeration

Before attacking a system, an attacker needs to know what runs on it. Port scanners, **Nmap** above all, send probes to a range of ports and analyse the answers to find which are open, which services listen and which software versions run. It is active probing, but it's so routine that on an internet-facing host it disappears into the background noise of constant scanning, and what it reveals directly decides which exploits to try.

5 Web application attacks

Web applications are among the most targeted surfaces: public, often complex and frequently carrying flaws introduced during development. The **OWASP Top 10** is the industry-standard reference for their risks. These are the technically most significant.

5.1 SQL injection (SQLi)

SQL injection happens when user input ends up in a database query without proper handling, letting the attacker change the query's logic. Take a login form that builds this query:

```sql
SELECT * FROM users WHERE username = '[input]' AND password = '[input]'
```

If the attacker enters `' OR '1'='1` as the username, the query becomes:

```sql
SELECT * FROM users WHERE username = '' OR '1'='1' AND password = ''
```

Since `'1'='1'` is always true, the query returns users and authentication is bypassed. More sophisticated SQLi can extract an entire database, write files on the server or, in some configurations, run operating system commands. Variants include **blind SQLi** (no direct output: the attacker infers information from the application's behaviour or timing), **time-based blind SQLi** (deliberate delays as a binary signal) and **out-of-band SQLi** (data exfiltrated through DNS or HTTP requests triggered by the database server). The defence is the same for all: parameterised queries, never string concatenation.

5.2 Cross-Site Scripting (XSS)

XSS happens when an attacker injects malicious scripts into pages viewed by other users. The browser treats the injected script as legitimate and gives it access to everything legitimate scripts can reach: cookies, session tokens, local storage, the DOM and the ability to make authenticated requests on the user's behalf.

Reflected XSS arrives through a crafted URL: the script is in the request and reflected in the response, so the victim must click the link. **Stored XSS** (persistent) is injected into the application's storage (a comment, a profile name, a forum post) and runs for everyone who views that content, which makes it much more dangerous, since it needs no targeted delivery. **DOM-based XSS** happens entirely in the browser, when client-side JavaScript processes attacker-controlled data unsafely.

5.3 Cross-Site Request Forgery (CSRF)

CSRF exploits the fact that browsers automatically attach credentials (cookies) to requests to a site, wherever the request comes from. If an attacker gets a logged-in user to load a page they control, that page can silently make authenticated requests to the target site (submitting forms, changing settings, starting transfers) in the victim's name.

The classic defence is a CSRF token: a secret, unpredictable value included in forms and validated by the server. A cross-origin attacker can't read it because of the Same-Origin Policy, so they can't put it in forged requests. Modern defences also include `SameSite` cookies.

5.4 Server-Side Request Forgery (SSRF)

SSRF is among the most dangerous modern web vulnerabilities. It occurs when an application fetches a remote resource based on user-supplied input without proper validation. The attacker can supply an internal URL, such as `http://169.254.169.254/latest/meta-data/` on AWS, and make the server query internal services that aren't reachable from the internet. In cloud environments, SSRF against the instance metadata service can yield temporary credentials with significant IAM permissions.

> **CASE STUDY: CAPITAL ONE (2019), SSRF IN THE CLOUD**
>
> In March 2019 a former AWS employee exploited a misconfigured web application firewall running on a Capital One EC2 instance. The firewall could be made to issue server-side requests, which the attacker used to query the instance metadata service and obtain temporary AWS credentials tied to

> an overly permissive IAM role. With them they listed and downloaded S3 buckets holding the data of roughly 100 million people in the US and 6 million in Canada, including names, addresses and credit-related information, and some 140,000 Social Security numbers. The attacker was identified because they boasted about it online. The technical failure was the combination of an SSRF-prone firewall and an IAM role with excessive S3 permissions: least privilege, giving services only what they need, would have limited the damage considerably.

5.5 Insecure deserialisation

Serialisation converts an object into a format that can be stored or transmitted (JSON, XML, binary); deserialisation rebuilds the object. If an application deserialises user-supplied data without validation, an attacker can craft a malicious serialised object that executes arbitrary code on the server when deserialised. Java deserialisation flaws have been particularly severe: the Apache Commons Collections gadget chain, published in 2015, enabled remote code execution against any Java application using that library, including enterprise products such as WebLogic, WebSphere and Jenkins.

6 Malware

Malicious software is the main tool attackers use to establish and keep access to compromised systems. Knowing the main categories helps explain which defences are needed and why endpoint security is so hard.

6.1 Viruses and worms

A **virus** attaches itself to a legitimate program and needs a human action to spread (opening an infected file, running an infected executable). A **worm** spreads on its own across networks by exploiting vulnerabilities. WannaCry, which in 2017 infected more than 200,000 systems in about 150 countries, was a worm that used the EternalBlue exploit (developed by the NSA and leaked by the Shadow Brokers) to spread through unpatched SMB services.

6.2 Trojans

A Trojan masquerades as legitimate software while carrying out malicious functions in the background. Unlike viruses, Trojans don't replicate: they rely on social engineering to get installed (a cracked software download, a fake system utility, a malicious attachment). Once in, a Trojan can install a backdoor, steal credentials, join a botnet or download further payloads.

6.3 Ransomware

Ransomware encrypts the victim's files and demands payment for the key. It has gone from a consumer nuisance in the early 2010s to one of the main threats to enterprises, hospitals, governments and critical infrastructure.

Modern **ransomware-as-a-service (RaaS)** operations such as LockBit and BlackCat/ALPHV work like software businesses: a core team develops and maintains the ransomware and the negotiation infrastructure, while affiliates carry out the intrusions and pay the core team a share of the ransom.

That division of labour has multiplied the number and sophistication of attacks (even if law enforcement has since disrupted some of these groups, others take their place).

Today's attacks typically use **double extortion**: besides encrypting files, the attacker exfiltrates data first and threatens to publish it if the ransom isn't paid. This removes the old “we have backups, we won't pay” defence.

> **CASE STUDY: COLONIAL PIPELINE (2021)**
>
> In May 2021 the DarkSide group attacked Colonial Pipeline, operator of the largest fuel pipeline in the United States. The initial access was a compromised VPN account: the password of a legacy profile that was no longer in use but hadn't been disabled, and crucially had no multi-factor authentication. Colonial shut down operations as a precaution, causing fuel shortages on the East Coast and a brief spike in prices. The company paid about $4.4 million in ransom, and the FBI later recovered roughly $2.3 million of it. The attack showed that ransomware against critical infrastructure was not hypothetical, and that the exposure extends beyond operational systems to the IT that supports them.

6.4 Rootkits

A rootkit is malware built to hide its own presence and that of other malware. Kernel-level rootkits hook system calls to hide files, processes and network connections from the operating system and from the tools running on it. A system with a kernel rootkit can't be trusted to report its own state, because the rootkit intercepts the very calls security tools use to inspect it.

6.5 Spyware and keyloggers

Spyware silently monitors user activity and sends the data to the attacker: browsing history, keystrokes, screenshots, microphone and camera, clipboard contents. **Keyloggers** record keystrokes specifically, capturing passwords, card numbers and anything else typed. At the most sophisticated end are commercial products such as Pegasus, developed by NSO Group and sold to government clients, which use zero-day vulnerabilities to take over a device with no user interaction.

6.6 Botnets

A botnet is a network of compromised machines (bots or zombies) under the attacker's control through C2 infrastructure, used for DDoS, spam, credential stuffing, cryptocurrency mining and distributing more malware. The Mirai botnet, which in 2016 launched a DDoS of over a terabit per second against the DNS provider Dyn and disrupted services such as Twitter, Reddit and Netflix, was made up mostly of IoT devices (cameras, routers, DVRs) whose default credentials had never been changed.

7 Advanced Persistent Threats (APT)

APTs are sophisticated, long-running attacks, typically by state actors or highly organised criminal groups with significant resources. What defines them is persistence (access kept for months or years), stealth (avoiding detection while pursuing objectives) and targeting (going after specific high-value targets rather than any available system).

7.1 Supply chain attacks

Instead of attacking a well-defended target directly, a supply chain attack compromises a trusted third party (a software vendor, a managed service provider, a hardware manufacturer) and uses that relationship as the way in.

> **CASE STUDY: SOLARWINDS / SUNBURST (2020)**
>
> In December 2020, FireEye revealed that attackers had compromised the build pipeline of SolarWinds, an IT management software vendor, and inserted malicious code, dubbed SUNBURST, into a signed update of the Orion platform. The update reached roughly 18,000 SolarWinds customers, including US government departments (Treasury, Commerce, Homeland Security, State) and major technology companies. SUNBURST was extremely careful: it stayed dormant for up to two weeks after installation to evade sandbox analysis, communicated through DNS traffic that looked legitimate, imitated normal Orion traffic and checked for security tools to adapt its behaviour. The US government attributed the operation to Russian foreign intelligence (the APT29 group). The lesson: even organisations with mature security programmes can be compromised through trusted updates, and defending against supply chain attacks takes a different way of thinking than perimeter defence.

7.2 Zero-day exploits

A zero-day is a vulnerability unknown to the vendor, for which no patch exists; the vendor has had "zero days" to fix it. Reliable zero-days in widely deployed software are among the most valuable assets of the offensive ecosystem, and governments and intelligence agencies pay millions for them. The Stuxnet worm, discovered in 2010 and widely attributed to a US-Israeli operation (which neither government has officially confirmed), used several zero-day vulnerabilities at once, an unprecedented number, to compromise Iranian uranium enrichment facilities and physically damage centrifuges by manipulating their control systems while reporting normal operation to the operators.

8 Exploiting known vulnerabilities

Beyond zero-days, attackers routinely exploit known vulnerabilities for which patches exist but haven't been applied. The gap between disclosure and widespread patching is a primary attack surface for both opportunistic and targeted attackers.

Table 2

Class	Mechanism	Typical impact
Buffer overflow	Writing data beyond the bounds of an allocated buffer, overwriting adjacent memory, including return addresses	Arbitrary code execution, privilege escalation
Use-after-free	Accessing memory after it was freed and possibly reallocated to another object	Arbitrary code execution, information disclosure
Integer overflow	An arithmetic operation produces a value too large for the type and wraps to an unexpected one	Buffer overflows, logic errors, security bypasses

Path traversal	Using `../` sequences to leave the intended directory and reach arbitrary files	Unauthorised file access, credential theft
Command injection	User input inserted unsanitised into operating system commands run by the application	Remote code execution with the application's privileges
XML External Entity (XXE)	An XML parser processes external entity references in attacker-supplied XML	File reading, SSRF, denial of service

9 Insider threats

Not all threats come from outside. Insider threats are malicious or negligent actions by employees, contractors or other trusted people with legitimate access, and they are a significant and often underestimated category.

Malicious insiders abuse their access deliberately, for personal gain, revenge or ideology. Edward Snowden's exfiltration of NSA documents is the most famous case at scale, but most malicious insider incidents involve theft of data for competitive advantage, financial fraud or sabotage. **Negligent insiders** cause harm through carelessness: clicking phishing links, misconfiguring cloud storage, losing unencrypted devices, sharing credentials, skipping procedures. Roughly two thirds of breaches in Verizon's Data Breach Investigations Report involve a human element.

> **THE MISCONFIGURATION EPIDEMIC**
>
> Misconfigured cloud storage is behind an enormous number of data exposures: publicly accessible S3 buckets, exposed Elasticsearch indices, world-readable Firebase databases. These aren't sophisticated attacks. They are automated scans for known misconfiguration patterns followed by trivial exploitation, and the Verizon report regularly lists misconfiguration among the leading error types behind breaches.

10 Cryptographic attacks

Cryptography is the mathematical foundation of most security systems. Attacks on it target the algorithms, their implementations or the protocols that use them.

10.1 Downgrade attacks

A downgrade attack forces a connection to use an older, weaker protocol version or cipher suite that the attacker can then attack. POODLE (2014) exploited the possibility of forcing connections back to SSL 3.0, which had exploitable weaknesses. Other well-known TLS attacks work differently: BEAST targeted a weakness in CBC mode in old TLS versions, while CRIME and BREACH exploited information leaked by compression.

10.2 Hash collisions

Cryptographic hash functions must be collision-resistant: it should be computationally infeasible to find two different inputs with the same hash. When that fails, digital signatures and certificate

validation can be forged. MD5 has been broken in practice since 2004, and SHA-1 fell to Google's SHAttered attack in 2017, which produced two different PDF files with the same SHA-1 hash.

10.3 Brute force and rainbow tables

Password hashes stored in a database can be attacked offline: with the hash file, an attacker can try to recover the passwords without touching the live system. A brute force attack tries every combination, a dictionary attack tries common passwords and variations, and a rainbow table attack uses precomputed chains of hashes to recover passwords quickly, trading storage for computation time.

Salting, adding a unique random value to each password before hashing, defeats rainbow tables by making precomputation infeasible. Modern password hashing functions such as bcrypt, scrypt and Argon2 are deliberately expensive to compute, so brute force stays slow even on powerful hardware.

11 The defence framework: thinking in layers

Effective defence is never a single control. It is a layered architecture, defence-in-depth, in which several independent controls must all be bypassed for an attacker to succeed, and the failure of one layer doesn't mean total compromise.

Table 3

Layer	Controls	What it addresses
Perimeter	Firewalls, WAF, DDoS mitigation, network segmentation	Reduces the attack surface and blocks known-bad traffic
Identity	MFA, privileged access management, zero trust architecture	Makes stolen credentials less useful and limits lateral movement
Endpoint	EDR/XDR, application allowlisting, patch management	Detects and contains malware and reduces exposure to vulnerabilities
Application	Secure SDLC, SAST/DAST, dependency scanning, secrets management	Prevents vulnerabilities in code and dependencies
Data	Encryption at rest and in transit, DLP, access controls, backups	Limits the impact of a successful breach and enables recovery
Detection	SIEM, log aggregation, threat hunting, anomaly detection	Identifies attacks in progress and reduces dwell time
Response	Incident response plan, forensic capability, crisis communication	Contains and recovers from successful attacks

> **THE MOST IMPACTFUL CONTROLS**
>
> With a limited security budget, three controls give disproportionate value. (1) **Multi-factor authentication** on all accounts, especially privileged ones: it stops most automated credential attacks (preferably phishing-resistant methods such as FIDO2 keys, given AiTM). (2) **Patch management**: a large share of exploited vulnerabilities are known and already patchable when the attack happens. (3) **Regular, tested, offline backups**: the only reliable defence against ransomware that doesn't involve paying.

12 Conclusion

The threat landscape described here is not static. New techniques emerge continuously, existing ones get refined and automated, and the attack surface grows with every new service, every new integration and every new employee.

But the fundamentals of defence are more stable than the specifics of individual attacks. Understanding how attacks work, mechanically and not just by category, is the prerequisite for defences that hold. An attacker who has spent months studying your environment, your systems and your people won't be stopped by a compliance checkbox. They will be stopped by layered controls, active detection and a practised response.

Security is not a product you buy; it is an ongoing discipline of understanding, investment and continuous improvement. In a world where attackers optimise for the path of least resistance, being a meaningfully harder target often means being the one they move past.

This article reflects the state of the threat landscape as of early 2026. Vulnerabilities, tools and techniques evolve fast. For operational guidance, consult current advisories from CISA, NIST and the relevant sector ISACs, or from your national CSIRT.

HOW TO CITE THIS ARTICLE

Francesco Barbato (2026). Cybersecurity attacks explained: a technical guide to how attackers actually work. localhost:3000.

Online version: http://localhost:3000/en/blog/cybersecurity-attacks-technical-guide