

THE IN A HURRY SERIES

Cybersecurity *in a Hurry*

Stay safe without paranoia

Max Brocklesby

BOOKSINAHURRY.COM

Contents

The Whole Thing in One Page 4

Why You Should Care 5

The Core Ideas 7

How It Actually Works 21

What People Get Wrong 31

Use It 36

Terms 41

Go Deeper 45

Notes and Sources 47

Bibliography 56

The Whole Thing in One Page

Cybersecurity has been given a face, and the face is wearing a black hoodie. It sits in a dark room, types furiously and breaks through a glowing padlock. The picture is useful to people selling fear. It is almost useless to anyone trying to stay safe.

An attack need not involve a genius defeating advanced mathematics. An exposed service, an old account or a persuasive message may be enough to start it. Excessive permissions let it spread; a recovery process nobody tested makes the damage harder to undo. The attacker looks for a cheap path to something valuable. Cybersecurity is the work of making those paths harder, narrower, more visible and less damaging.

Begin with the possible loss. A teenager, a shop and a hospital do not need the same defences because they do not hold the same things, face the same attackers or suffer the same consequences. The useful question is not whether a device is secure. Nothing connected and useful deserves that absolute word. Ask what matters, who might want it, how they could reach it and what failure would cost. That is threat modelling, and it is the cure for both complacency and paranoia.

Then control authority. Modern life is run through accounts, and an account is a bundle of permissions. Protect the email account that can reset the others. Prefer passkeys where supported, or distinct passwords through a password manager with multi-factor authentication. The method matters: traditional codes can be phished; a passkey with a local PIN or biometric check already combines factors. Keep ordinary work away from administrator rights. Remove accounts, software and connections that no longer earn their risk.

Assume prevention will sometimes fail. Updates close known openings, but not every opening. Filters catch many malicious messages, but not every message. A careful person can still be rushed, deceived or handed a convincing request through a compromised account. The answer is layered defence: limit what one mistake can reach, record important activity, notice

abnormal use, and know who can stop the spread.

Recovery belongs inside security, not after it. A backup that shares the attacker's access may be deleted with the original. A copy never restored is a hope, not a recovery system. An incident plan that exists only in a folder may fail when email, identity services and laptops are unavailable. Protect clean copies, test restoration and rehearse the first decisions before panic makes them for you.

This is why staying safe does not require watching every packet, distrusting every stranger or buying every security product. It requires proportion. Spend attention where plausible loss is large. Prefer controls that work automatically. Give people clear ways to verify unusual requests. Reduce privileges and dependencies. Detect what prevention misses. Preserve a way to resume work without depending on the attacker.

The loop closes where it began. Your threat model sets the priorities. Incidents, near misses and changes in the system then reveal where that model was wrong, so the next round of protection is better aimed. Security is not a fortress completed once. It is disciplined adaptation under limited time, money and attention.

That is the book.

Why You Should Care

On 12 May 2017, staff across parts of the National Health Service found computers locked by WannaCry ransomware. NHS England later estimated that more than 19,000 appointments had been cancelled. Five accident and emergency departments had to divert patients. The malware did not need to alter a medical record to disrupt care. Making ordinary work unavailable was enough. The damage reached people who had never touched the affected computers, through a missing appointment or a journey to a different hospital.

That is the first correction. Cybersecurity is often described as the protection of data, which sounds like a concern for banks, governments and people

with secrets. In practice it protects action. Your email is how other accounts recognise you. Your phone may approve payments and hold the keys to your home, work and social life. A company's inventory, payroll, bookings, customer records and supplier instructions are parts of its ability to operate. Integrity matters because a false bank detail can be more costly than a leaked document. Availability matters because a hospital without systems, a warehouse without labels or a family without access to its photographs has lost something in the physical world.

The second reason is that the attack has industrialised. An attacker does not need to know you before testing whether your old credentials work somewhere else. Internet-facing systems can be scanned at scale. Stolen access can be traded. Fraud messages can be adapted cheaply, translated quickly and sent by the thousand. An ordinary account can attract an attacker because a route is available and the expected return looks positive. Fame is unnecessary. This does not make attackers omnipotent. It means obscurity is a weak defence, while common safeguards deserve more respect than dramatic countermeasures aimed at an opponent you are unlikely to meet.

This sounds bleak until the economics are turned around. Large-scale attack depends on cheap repetition, so small increases in attacker cost matter. One unique credential defeats reuse from another breach. A passkey binds the login to the legitimate service and resists common credential phishing. Automatic updates close many known openings before ordinary delay becomes an attacker's advantage. Least privilege turns a stolen account from a master key into a key for one room. A tested backup makes destructive extortion less effective. None is perfect. Perfection is not required for a control to change the attacker's calculation or contain the result.

The third reason is personal agency without personal blame. Security advice has often treated the user as the defective component: do not click, do not trust, remain alert. That asks a human being to perform flawless fraud detection while working, travelling, caring for children and answering

messages. Good security still needs judgement, especially around unusual requests, but it places the hardest work in design. Safe settings should be the default. High-risk actions should require stronger proof. Recovery should be possible without handing a stranger unlimited power. Reporting a suspected mistake should be quick and consequence-free enough that people do it early.

There are limits. A person facing stalking, a dissident, a senior executive and an ordinary consumer face different threats. Some people need specialist advice because their opponents have patience, money or privileged access. This book does not make everyone safe from a state intelligence service, and it does not pretend that one checklist fits a household and a power grid.

What it gives you is a way to tell a useful precaution from an expensive distraction. A warning about danger is easy to produce; an explanation of how a particular loss could happen is harder and more valuable. Follow that explanation and familiar objects change: the inbox becomes an authority, the update becomes a repair, and the spare copy becomes a decision about tomorrow. The subject gets interesting when the padlock disappears and the working parts come into view.

The Core Ideas

Start With the Loss, Not the Hacker

The wrong way to begin cybersecurity is with a list of frightening things an attacker might do. The list has no natural end. A criminal could guess a password, steal a laptop, corrupt a supplier, exploit an undisclosed flaw, bribe an employee, cut a cable or spend six months studying one target. Treat every imaginable path as equally urgent and you have built paranoia, not security.

Imagine losing your phone. Replacing the object is one loss. Losing access to your accounts is another; someone reading private messages is a third. Security professionals call the things at stake assets, but a list of devices

misses the distinction. The same phone can carry money, privacy and the ability to contact someone when you need help. At work, a readable database may still be dangerous if its payment details have been altered. A missing public brochure is an inconvenience; missing payroll on payday stops an operation. What matters is the consequence, not the price tag on the equipment. That is why a modest device can deserve more protection than an expensive one.

Next ask who might act and why. The plausible set may include opportunistic criminals, a dishonest insider, an abusive partner, a competitor, activists or a state. Their resources, access and appetite differ. A criminal crew may automate credential testing because each attempt is cheap. An insider may need no technical exploit because the system already trusts the account. A state may spend time on a target that would never repay an ordinary criminal. Calling all of them hackers erases the differences that determine sensible defence.

Then describe a path. What must be true for the harm to occur? Perhaps an attacker needs the email account, then a password reset, then the ability to change bank details. Perhaps ransomware must reach both the live files and the backups. Perhaps a fraudster needs an employee to accept a new payment instruction without independent confirmation. This is threat modelling: turning a cloud of anxiety into a chain of conditions that can be broken.

Risk is often described as likelihood multiplied by impact. That is a useful reminder, not a precision instrument. Neither number is known cleanly, and multiplying two guesses does not produce knowledge. The practical aim is ranking. A common, moderate loss may deserve more attention than a spectacular event with no plausible route. A rare event may dominate where the consequence is fatal or irreversible. Controls should follow those judgements, with uncertainty stated rather than hidden behind a score.

Absolute labels such as secure and insecure weaken that ranking. A system can resist one attacker and fail against another; protect confidentiality while losing availability; or be well defended today and exposed after a new

connection tomorrow. Security is always security of something, against something, for some period and at some cost. Saying that aloud prevents assurance from turning into a status badge.

Threat models also have boundaries. A household can reasonably prepare for account theft, fraud, device loss and ransomware without designing against a tailored hardware implant. A small firm should care about supplier access and payment fraud, but it cannot reproduce a national intelligence agency. The test is whether the model captures the attacks that would change the decision, not whether it contains every attack ever demonstrated.

Proportion is a decision about what not to worry about as well as what to protect. Once a loss has a plausible route and a serious consequence, give it attention. Possibility alone cannot make the same claim on your day.

Accounts Decide Who Can Act

Most digital systems do not know you. They know that somebody presented an acceptable credential and completed the required checks. From that moment, the account can read, change, approve or delete whatever its permissions allow. The account, rather than the person at the keyboard, is what the system has agreed to trust.

This is why email deserves unusual protection. It is often the recovery route for shopping, social media, cloud storage and financial services. An attacker who controls it may not need to break each account separately. They can ask the services to recognise a new password, approve a new device or send a recovery link. The master key may be the account that receives the spare keys.

Passwords fail in several distinct ways. A short or predictable one may be guessed. A reused one may be tried after it leaks elsewhere. A strong one may be entered into a convincing false site. A unique, randomly generated password defeats reuse and guessing far better than a memorable variation does, which is why a reputable password manager changes the problem. You protect one strong primary access route, and the manager creates and

stores different credentials for the rest. The manager becomes important infrastructure, so its recovery method and protected devices matter too.

Multi-factor authentication adds a second kind of proof. In the familiar formulation, the factors come from something you know, something you have or something you are. A text-message code can block an attacker holding only a password, yet fail against number takeover or a false login page relaying the code. An approval prompt can be accepted through fatigue or confusion. A time-based code can be phished in real time. More factors improve many situations, but the method and the attack determine the result.

Passkeys change the exchange. The service stores a public key. The user's device or credential provider proves possession of the matching private key without sending it. The login is bound to the legitimate service, so there is no reusable secret for a false site to collect in the usual way. A local PIN or biometric check can authorise use of that private key: possession of the credential plus verification of the user. That can provide multi-factor authentication in one brief action, without an extra text-message code. Correctly implemented passkeys resist common credential phishing and reuse better than passwords and traditional second factors.

They do not abolish identity risk. A compromised device, weak screen lock, malicious recovery process or stolen active session can still matter. Some passkeys stay on one device; others synchronise through a provider. The protections around that provider account and its recovery process also matter, though account takeover does not automatically reveal every synchronised key. Organisations also face legacy applications, shared devices and complicated enrolment. Prefer phishing resistance where the service supports it, while keeping recovery and device security part of the same system.

Sessions deserve separate thought. After a successful login, a service usually issues a token so the user need not authenticate for every click. Steal that token and the attacker may inherit accepted access without knowing the password or repeating the second factor. Password changes

should therefore be paired with review and revocation of active sessions, while sensitive changes may require fresh authentication. The login ceremony protects the door; session management governs how long the open door remains trusted.

Authentication proves control of a credential. Authorisation decides what the resulting account may do. Confusing them creates dangerous designs: a valid login is treated as permission for every sensitive action. Strong systems ask for more evidence when risk rises, restrict changes to recovery settings, notify the old channel when a new one is added and make high-value actions harder than reading ordinary information.

An account keeps changing after its first login. Devices are added, sessions continue, and lost credentials need replacing. Protecting identity means following those changes, especially the route back in when the normal proof is gone.

Exposure Grows Through Connections and Dependencies

A machine without connections is easier to secure. Each useful connection changes that: a web page accepts input, a phone installs applications, a company admits a supplier. Cloud software links services whose administrators may never meet. The attack surface is the set of places where an untrusted person or process can interact with the system, and modern systems create more of it because connection is their purpose.

The first defence is embarrassingly administrative: know what exists. Unknown laptops do not receive managed updates. Forgotten accounts remain usable. An internet-facing service inherited through an acquisition may be nobody's responsibility. Software can reach the end of support while still running a critical task. At home, this includes the router that keeps working after everyone has forgotten its settings. Its support life matters as much as its signal strength. An inventory turns invisible exposure into somebody's job.

A vulnerability is a weakness that could be used. An exploit is a method that

uses it. The term zero-day usually describes a flaw unknown to its supplier, or an attack exploiting a flaw before a fix is available. Usage varies, but the practical question is whether a defence exists and has been applied. Most organisations should not let the glamour of zero-days distract from known weaknesses. Once a flaw is public and exploitation becomes practical, attackers can scan for systems that remain exposed. Updating closes the specific route; automatic updating shortens the period during which ordinary delay becomes attacker opportunity.

Patching is still a risk decision. A hospital device, factory controller or old business application may fail if changed without testing. Some updates need a restart that interrupts service. The answer is not to choose between instant patching and permanent delay. Prioritise systems reachable from the internet, weaknesses known to be exploited, highly privileged components and assets with severe consequences. Test where failure matters, isolate what cannot be fixed and replace technology whose unsupported state has become the main risk.

Configuration can matter as much as code. A cloud storage area may be technically sound and publicly readable because somebody selected the wrong setting. Remote access may use a default credential. A security feature may exist but remain off because the vendor sells ease first and protection as an option. Secure-by-default design shifts work towards the party best placed to perform it, rather than requiring every buyer to become an expert before the product is safe enough to use.

Dependencies widen the surface beyond what you own. The SolarWinds compromise showed the power of trusted distribution: malicious code inserted into a signed software update reached customers through a route designed to receive legitimate improvements. A supplier with remote access, a library included in an application, a cloud administrator and a payment processor can each become part of your security boundary. Contracts and questionnaires help, but they do not erase technical dependence. Limit supplier permissions, know which services are critical, monitor unusual access and plan for the provider being unavailable or

compromised.

Cloud services change who performs tasks, not whether the tasks exist. A provider may secure physical facilities, infrastructure and parts of the platform while the customer still controls identities, permissions, data settings and application behaviour. The boundary varies by service. “In the cloud” describes an operating arrangement, not a proof of safety.

Reducing exposure can spare the work of detecting abuse later. An unused service or an obsolete account has little benefit to set against its risk. Retiring it removes something that would otherwise need patching, monitoring and explaining. Security sometimes improves because there is less system left to defend.

Privilege Sets the Blast Radius

An intrusion becomes a catastrophe through authority. The first compromised account may be ordinary; the damage depends on what that account can reach, what credentials it can collect and whether it can become something more powerful. Privilege is permission to perform sensitive actions. The blast radius is the amount of the system that one failure can affect.

Least privilege means giving a person, device or process only the access needed for its current job. It sounds obvious until convenience arrives. Administrator rights make installation easy. Shared accounts avoid licence costs and handovers. Broad cloud roles prevent support calls. Permanent supplier access saves arranging a temporary window. Each choice moves effort from today into an unknown future incident.

Privilege also belongs to machines. Applications, scripts and service accounts hold credentials so they can read databases, move files or manage infrastructure without a person present. These identities are easy to forget because they do not resign, complain or appear in staff lists. Their secrets may live in code or configuration for years. They need an owner, narrow permissions and a way to expire or replace credentials safely. This is not a reason to make employees change sound passwords on a calendar:

change those when compromise is suspected, and prevent reuse from the start.

Separation changes the shape of failure. An employee can see the records needed for a task without receiving a bulk-export permission. This limits scale; it cannot prevent someone copying information they are allowed to read. A finance worker can prepare a payment but another person must release it. The account used for email is different from the account used to administer servers. Backups accept new copies from production but production credentials cannot erase protected history. One stolen identity no longer supplies every condition the attacker needs.

Network segmentation applies the same idea to movement. Systems are divided so that access to one area does not confer access to all others. The boundary may use networks, identities, application rules or combinations of them. Good segmentation follows business need: a reception device has no reason to initiate connections to payroll; a contractor has no reason to browse an entire internal network; a test environment should not hold production secrets by habit.

Zero trust is the current label for a related model, and the label is easy to abuse. It does not mean trusting nobody, interrogating employees constantly or buying one product. It means that network location or device ownership does not grant broad, lasting trust by itself. Access is made to a resource under policy, using signals such as identity, device state and context, with permissions kept as narrow as the work allows. The useful part is the removal of the soft internal network where one successful entry becomes a tour pass.

Privilege must also expire. Temporary administrative access is safer than a standing right that may outlive the task, the employee or the supplier. Dormant accounts and old tokens are attractive because legitimate users no longer notice their use. Regular review matters most around powerful roles, recovery administrators, service accounts and machine credentials, where one overlooked secret can survive long after the person who created it has left.

There is a trade-off. Controls that are too restrictive drive work into spreadsheets, personal accounts and unofficial messaging. A system can reduce formal permissions while creating a shadow route with less oversight. Least privilege therefore needs usable request and approval paths. People must be able to get legitimate work done before the deadline that made the shortcut tempting.

The deepest security property here is containment. You may fail to prevent one phish, one stolen laptop or one vulnerable application. You can still stop that event from becoming control of the company. Good architecture does not assume the first line will hold. It decides in advance how far failure is allowed to travel.

Design for People Having an Ordinary Day

Imagine an accounts clerk paying a familiar supplier. The invoice is expected, the amount is right and the message continues a real conversation. Only the bank details have changed. Calling the resulting loss human error tells us almost nothing about why the payment was possible. The useful question is whether the organisation made that change depend on anything beyond a plausible email. One account may have been compromised; the process need not give it the power to redirect the money.

Social engineering works by borrowing trust. The message appears to come from a colleague, bank, supplier, courier, manager or government department. It may create urgency, secrecy, fear, curiosity or helpfulness. The technical content can be modest because the request fits an existing relationship. Compromising a real email account is especially powerful: the attacker inherits the address, signature, conversation history and expectation that the sender belongs there.

Phishing is one route. A link may collect credentials, an attachment may run malicious code, or a message may begin a longer fraud. Voice calls, text messages and support chats can do the same work. Generative systems can improve grammar, translation and variation, but fluency was never the main defence. A polished message can be false and a clumsy one can

come from a genuine colleague. The stronger question is whether the requested action deserves separate verification.

Design the check around consequence. A low-value newsletter subscription needs little friction. A new bank account, password reset, payroll diversion, security setting or release of sensitive data needs more. Verification should use a channel the requester does not control: call a known number already on file, open the service through a saved route, ask a second authorised person, or require an approval generated inside the trusted system. Replying to the suspicious message proves only that the same channel can answer.

The verification route must also be prepared before the request arrives. Searching the message for a phone number or following its link lets the sender choose both the claim and the supposed check. Stored contact details, known application routes and pre-agreed approval rules remove that choice. Good verification does not demand that a tired person become a detective. It gives the person a trusted second path.

Training can teach patterns and procedures. It cannot make attention permanent. Repeated simulations may improve reporting and expose weak processes, but a programme that celebrates low click rates can mistake a test for the real threat. The important measures are whether high-risk actions have safeguards, whether people report quickly, whether the response is useful and whether one error can become a large loss.

Help desks and account-recovery teams deserve special attention because their job is to restore access to people who lack the normal proof. An attacker can exploit politeness, authority cues or knowledge taken from public records. Staff need defined evidence, escalation routes and permission to delay an unusual request. "Use your judgement" is weak control where the caller's goal is to manipulate that judgement.

People can also detect what tools miss. An employee may notice that a request is out of character, a supplier's bank details have changed unexpectedly or a shared folder contains a file that does not belong there.

Reporting turns local suspicion into organisational visibility. Punishing honest mistakes teaches silence, which gives the attacker time.

The aim is not to remove people from security. It is to put human judgement where context matters and remove it where repetition makes error likely. Let software generate credentials, apply routine updates and block known malicious files. Let people question an unusual transfer, assess a business exception and decide whether shutting down a service is worth the cost. Security improves when the task matches the decision-maker.

Prevention Is Incomplete Without Visibility and Response

A prevented attack leaves little evidence of the disaster that did not happen. An undetected attack can look like normal operation. Accounts still log in, files still open and invoices still move while data is copied or access is prepared for later use. Detection is the work of noticing that the system's behaviour no longer fits the authorised story.

Logs are records of events: a login, permission change, process start, file access, network connection or administrative action. They are useful only if the right events are recorded, protected, retained and examined in time. A million lines stored for compliance do not create awareness. A small set of well-chosen signals around identity, privilege, critical data and backup administration may be more valuable than exhaustive noise nobody can interpret.

The records need their own integrity. If an attacker with administrator rights can erase or alter every local log, the history disappears with the intrusion. Central collection, restricted access and reliable time help investigators join events across systems. Retention should match the period in which an attack may remain unnoticed and the legal or operational need, rather than a round number copied from another organisation.

Context separates an event from an incident. A login from a new country might be travel, a privacy service or theft. A large data transfer might be a backup or exfiltration. One failed password is ordinary; thousands across

many accounts may be automated guessing. Detection combines rules, baselines, threat information and human investigation. For a personal account, an unfamiliar login deserves attention, but an unfamiliar login followed by changed recovery details is more revealing. False positives consume attention, while thresholds set too high preserve quiet by hiding attacks. Tuning is part of defence, not an admission that the tool failed.

Speed matters because attackers use time to enlarge authority, learn the network, collect data and damage recovery. Early containment may mean disabling an account, isolating a device, blocking a route or pausing a service. Each action has a cost. Shut down too little and the attacker continues. Shut down too much and the response creates its own outage. The decision is easier when critical services, dependencies and acceptable interruption were mapped before the incident.

Incident response therefore begins before the alert. Roles must be assigned. Contact details must work when normal systems do not. Someone needs authority to contain; someone preserves evidence; someone manages operations, legal duties, customers and public communication. External specialists, insurers, regulators and law enforcement may need notice. A plan should state triggers and decision rights rather than attempt to predict every technical detail.

The operating sequence is usually preparation, detection and analysis, containment, removal, restoration and learning, with overlap and return between them. Removal without understanding can leave another route open. Restoration from an infected image can reintroduce the problem. Waiting for perfect certainty can give the attacker days. Response is managed uncertainty under pressure.

Visibility must respect privacy and proportionality. Monitoring employees or customers without a clear purpose can create legal, ethical and security risks of its own. Collected logs become sensitive assets. Record what supports a defined threat model, restrict access, set retention deliberately and explain the monitoring where appropriate. More surveillance is not the same as more security.

A barrier that fails need not decide the outcome. Visibility can expose the failure while there is still time to contain it. Recovery begins from what those defences preserved.

Recovery Is a Security Control

Security advice often ends at prevention because prevention is emotionally clean. Nothing bad happened. Recovery begins from the less flattering assumption that an account will be taken, a device will fail, a supplier will go offline or malicious code will run. That assumption is not defeatism. It is what makes survival design possible.

Backups are the obvious example and one of the most misunderstood. A backup is another copy, not an automatically safe copy. If the same administrator credential can alter production and delete every backup, ransomware has one job. If old versions are overwritten immediately, corruption may be faithfully copied. Synchronisation can do the same with a deletion: remove a photograph on one device and the service may remove it elsewhere too. Several matching copies are not necessarily an independent way back. If restoration has never been tested, the organisation does not know whether the data, applications, keys and instructions needed to resume work are present.

Useful backup design creates separation. Some copies are offline, immutable or protected through a different administrative boundary. Version history preserves a point before corruption. Backup accounts use strong, distinct authentication. Deletion and policy changes generate alerts. Restores are tested at the scale and speed the business needs, not by opening one sample file. The recovery target includes time: data that returns after three weeks may arrive too late for payroll, treatment or trade.

Backups do not solve every ransomware problem. Attackers may steal information before encrypting systems and threaten publication. They may disrupt identity, communications, virtual infrastructure or physical operations. Clean data does not tell staff which machines can be trusted, which credentials must change or what customers need to know. Recovery

combines data, technology, people, premises, suppliers and decisions.

This is why continuity belongs with cyber defence. Which services must return first? What manual process can run safely for a day? Which dependencies have no substitute? Can staff communicate if corporate email and messaging are unavailable? Who can authorise emergency spending? A rehearsed answer shortens the period in which uncertainty becomes the main damage.

Exercises expose false assumptions cheaply. A discussion around a plausible scenario can reveal that the crisis list sits inside the unavailable network, that only one person knows how to restore the database, or that the supplier's promised response time does not cover a widespread attack. More technical exercises can test detection and containment. The goal is not theatre for auditors. It is discovering before the incident which decisions, access routes and facts are missing.

Governance decides whether any of this remains alive. Cyber risk belongs to the people who own the service and consequence, not to an isolated security team. Leaders choose how much interruption to accept for patching, which supplier concentration to tolerate, how much data to keep and which recovery time to fund. Security specialists provide evidence and challenge. They cannot decide the organisation's appetite for loss on its behalf.

The first account of what could go wrong now meets what did. An exercise may reveal a forgotten dependency; an incident may show that limited permissions contained the damage. Both change where the next pound and hour should go. A threat model earns its keep by being corrected, not by surviving every meeting unchanged.

A useful security promise can be tested: the essential work will return through a route the attacker has not taken away. An exercise that disproves it is a chance to repair the promise before someone depends on it.

How It Actually Works

The night connection became the problem

At about 8.30 on the evening of 2 November 1988, a program began spreading across the young Internet. It used several routes into a particular family of Unix systems and copied itself from machine to machine. A mistake in its design caused some computers to run repeated copies until they slowed or stopped. Within a day, an estimated 6,000 of the roughly 60,000 connected computers had been hit.

The Morris Worm did not erase files. It did not need to. Universities, laboratories and government sites disconnected systems, delayed work and spent days understanding and removing it. The network community discovered a property that now defines cybersecurity: connection turns another machine's weakness into your operating condition. A program released in one place could consume resources across institutions that had never heard of its author.

The incident also exposed the shape of defence. Administrators needed to know which systems were affected, share information, analyse the code, distribute fixes and restore operation without letting the worm return. Days later, the United States created a national computer emergency response team. Cybersecurity began to look less like protecting one locked machine and more like coordinating a society of fallible, dependent systems.

Finding a route

A modern attack often starts before any human at the target notices. Internet addresses and exposed services can be scanned. Public websites reveal staff, suppliers and technology. Leaked credential collections can be tested automatically against other services. Criminal markets sell access already obtained by somebody else. The first attacker may be a specialist in entry, while another performs fraud or ransomware later.

This changes selection. Some attacks are tailored because the target is valuable. Many are broad searches for cheap opportunity. A vulnerable

remote service, an account without an additional factor or a publicly exposed storage system may attract attention because it is visible, not because anyone formed an opinion about the owner. The attacker can reject difficult targets and move to the next address.

Defenders meet this stage through inventory and exposure management. Which services are reachable? Which software versions are running? Which accounts exist? Which suppliers can connect? A vulnerability scanner can identify known weaknesses, while attack-surface monitoring can show unexpected public systems. Neither knows the full business meaning. A forgotten test server may hold nothing, or it may contain a credential that opens production. The technical finding becomes useful when ownership and consequence are attached.

Threat intelligence adds information about current attacker behaviour, exploited weaknesses and malicious infrastructure. Its value is selective. A small retailer does not need a daily catalogue of every state group's tools. It needs timely knowledge that changes patching, monitoring or response. Information that produces no decision is an expensive form of weather reporting.

Attackers make resource choices too. A route is attractive when access is cheap, repeatable and likely to reach something saleable or coercive. Defenders often hear that the attacker needs to succeed once while they must succeed every time. The line is memorable and incomplete. A serious attack usually requires a chain of successes: find exposure, gain access, preserve it, increase authority, reach the asset and complete the objective before being stopped. The defender can break or expose the chain at several points. The asymmetry is real because attackers choose the time and can test many targets, but they do not receive unlimited attempts against one unchanged system. Friction, containment and detection accumulate. Security is the deliberate arrangement of those costs. The useful defender does not need to predict the whole campaign. Breaking one necessary link can be enough.

Initial access

The attacker now needs a foothold. Three routes show how different kinds of trust can supply one.

The first is identity. A password may be guessed, reused from another breach, stolen by malware or entered into a false service. An active session token can sometimes be stolen after login, bypassing the need to repeat authentication. Support and recovery processes can be manipulated when the normal credential is unavailable. Identity attacks are attractive because a successful login enters through the front door and may resemble authorised work.

The second is software. A reachable application accepts untrusted input and contains a weakness that lets the attacker perform an unintended action. The flaw may expose information, run code or change permissions. Public knowledge of a vulnerability can turn patch delay into a race. In the 2017 Equifax breach, attackers entered through an online dispute portal. The US Government Accountability Office's 2018 account put the exposed population at at least 145.5 million people. Equifax's investigation identified failures involving identification, detection, segmentation and data governance. The breach was not one bug enlarged to national scale by magic. The bug opened a route; the surrounding system determined how long the route remained available and how far it reached.

The third is trusted supply. Software updates, managed service providers, libraries and remote-support tools are designed to cross boundaries. If an attacker compromises that route, malicious activity can arrive wrapped in an expected relationship. Code signing proves that a package passed through a particular signing process. It cannot prove that every contributor, build system and decision before signing was sound.

These routes overlap. A malicious document may exploit software and steal credentials. A compromised supplier account may use a legitimate remote tool. An attacker may phish an administrator because exploiting the platform would cost more. The categories help defenders assign controls, but the

incident will not respect an organisational chart.

Initial access also says little by itself about severity. A phished newsletter account and a phished recovery administrator are technically the same class of event and operationally different worlds. The useful response connects the entry to current permission, possible escalation, data and business function. This is why alerts need asset and identity context. Without it, a security team can spend an hour on a noisy low-value device while a quiet privileged session changes the organisation's recovery settings.

Turning access into control

Initial access is often fragile. The attacker may control one browser session, one laptop or one low-privilege account. To retain access, they may create or alter credentials, misuse remote-management features, arrange for code to run again or hide inside legitimate services. Defenders call this persistence. The details vary, but the aim is stable return.

The next task is discovery. What system is this? Which users are active? Where are the valuable files? Which services trust one another? What security tools are present? Attackers learn before they act because noisy damage can close the route too early. Ordinary administrative tools can supply much of the information, which makes intent harder to infer from a command alone.

Privilege escalation seeks greater authority. A local account may exploit a flaw, obtain an administrator credential or find an overpowered service token. Once privileged, the attacker can disable controls, read protected material, create accounts or reach new systems. Credential stores and authentication infrastructure become high-value targets because they convert one foothold into many identities.

Defence at this stage is containment by design. Separate administrator accounts reduce the chance that ordinary email work exposes privileged credentials. Device controls restrict unauthorised code. Secrets are stored away from application files. Strong authentication protects remote and

administrative access. Permissions and network paths are limited. Alerts watch for new administrators, unusual authentication and security-control changes. Each measure removes a condition from the attacker's chain or makes the condition visible.

Machine identities complicate the picture. Applications need tokens and keys to call other services, often without a person present. Those credentials may be broad, long-lived and copied across environments because automation rewards convenience. A leaked token can therefore provide clean access that never triggers a failed login. Modern identity systems also federate trust: one provider tells many applications that a user has authenticated. Federation reduces separate passwords and can strengthen central control, while making the identity provider and its administrators more consequential. Concentrating trust can improve defence and increase the cost of its failure at the same time.

Moving, collecting and preparing

A flat organisation gives a compromised account too much to discover. Shared drives, broad internal trust and reusable administrator credentials let the attacker move laterally from the first system towards more valuable ones. Segmentation and resource-level access checks force repeated decisions. The attacker may still progress, but each boundary creates delay, evidence and another chance to stop them.

The map is rarely only a network diagram. Trust flows through single sign-on, shared administrator groups, software deployment, backup agents, support platforms and application programming interfaces. A connection may be closed at the firewall while an authorised cloud service still moves data between the same systems. Defenders therefore follow identities and permitted actions as well as cables and addresses. The question is which principal can cause which consequential change, through any route the architecture accepts.

Data collection has its own sequence. The attacker identifies material, stages it, compresses or packages it and transfers it through an available

channel. A sudden, enormous transfer is easy to imagine and sometimes easy to block. Slow use of approved services can look more normal. Defenders therefore combine content sensitivity, user role, destination, volume and timing. The aim is not to classify every byte perfectly. It is to notice when behaviour no longer matches the account's legitimate purpose.

Preparation for disruption can occur in parallel. Ransomware operators may seek backup consoles, virtual infrastructure and domain-wide administration before encrypting anything. The visible event is late in the intrusion. By then, the attacker may understand which systems restore the organisation and which people can approve payment. A ransom screen is not the beginning of the attack. It is the moment the attacker chooses to announce accumulated control.

This explains why one security product rarely defines the outcome. Antivirus might stop a known malicious file. It will not correct an overpowered account or a payment process with no verification. A firewall limits certain network paths. It cannot distinguish a legitimate user from an attacker operating the user's session with permitted tools. Controls work as a system because attackers switch routes when one becomes expensive.

Acting on the objective

Cyber incidents are grouped by technical method, but attackers care about objectives. Theft seeks information, money or access that can be sold. Fraud changes a trusted instruction. Espionage values continued secrecy and may avoid disruption. Extortion creates a cost, then offers to stop or reverse it for payment. Sabotage seeks interruption or destruction. Harassment may expose, alter or deny access because the personal effect is the objective.

An attacker can also deny a service without getting inside it. A flood of requests may consume the connection or the system's capacity until legitimate users cannot get through. When many sources contribute, this is a distributed denial-of-service attack, usually shortened to DDoS. A strong password does not create more capacity, and a backup does not clear a

saturated connection. Defences may need to work upstream, through an internet or service provider, before the traffic reaches the victim. The loss model still applies, but the intrusion sequence does not: sometimes the attacker's aim is to stop anyone entering at all.

The same access, once obtained, can support several outcomes. An email account can reveal confidential messages, reset other accounts, impersonate the owner and redirect invoices. A hospital system can be valuable for patient information and for the operational pressure created when treatment is disrupted. A factory control system can expose engineering data or stop physical production. The defender's asset map must therefore include the actions that data and systems enable.

WannaCry made this visible. The ransomware spread in May 2017 through systems that shared a known weakness. Within the English NHS, organisations lost access to devices and information; appointments were cancelled and patients were diverted. The National Audit Office found unpatched or unsupported systems among affected organisations and reported that a national response plan had not been rehearsed locally. Technical exposure and response preparation met in the same incident.

Attackers also exploit the victim's uncertainty. Is data merely encrypted or also stolen? Does the attacker still have access? Will a decryption tool work? Which systems can be trusted? Public claims by criminals are strategic messages, not reliable incident reports. Payment may buy a tool, delay publication, encourage further demands or achieve nothing. The decision depends on law, safety, evidence and business circumstances, and it should not be improvised by one frightened person over an anonymous chat.

The defender's race

Detection begins when a signal acquires meaning. A user reports an unexpected approval prompt. A monitoring service sees a new administrator. A supplier warns that its account was compromised. Files change at unusual speed. One clue may be weak; several can establish an incident.

Triage asks what happened, what may still be happening and what matters first. The defender needs a reliable timeline, affected identities, systems, data and dependencies. Evidence is preserved where possible, but containment may outrank perfect collection when harm is continuing. Isolating a device from the network can interrupt communication without switching it off. Powering it down is different: evidence held in volatile memory can disappear. Isolation can also change connection evidence, so neither action is forensically neutral. On safety-critical systems, involve the people responsible for safe operation.

Investigation works from traces that were never created for a perfect reconstruction. Logs can be missing, clocks can disagree and the attacker can use legitimate tools. Preserving a running device may help the investigation, but stopping ongoing harm can take priority. Experienced responders state competing explanations and seek observations that separate them. They do not wait for certainty before every protective action, but they record what was assumed so that containment can change when the evidence changes. The evidence is changing while the team tries to understand it.

Containment narrows the event. Accounts are disabled or reauthenticated. Devices are isolated. Malicious routes are blocked. Sensitive operations may be paused. The team must consider whether the attacker controls the systems used to coordinate the response. If corporate identity, email and messaging are suspect, an independent channel becomes necessary. Contact lists and clean devices prepared in advance now earn their cost.

Eradication removes the attacker's access and the weaknesses used.

Password resets alone fail if session tokens, malicious software or recovery settings remain. Rebuilding a machine fails if the same vulnerable service is restored unchanged. Closing one entry point fails if another administrator account survives. The team needs enough causal understanding to avoid declaring victory over the symptom.

Communication runs throughout. Staff need instructions they can follow. Customers, partners, insurers, regulators and law enforcement may require notice at different times and under different rules. Premature certainty can mislead; silence can deepen harm. Good communication says what is known, what is uncertain, what people should do and when the next update will come. It does not turn the investigation into theatre.

Returning to work

Recovery is prioritised restoration, not a ceremonial switch back on. Identity services, communications, core applications and data may depend on one another. Clean backups must be located and tested. Credentials and keys may need rotation. Systems are restored in an order that supports critical work without reopening the route. Temporary manual processes require their own controls so that urgency does not create fraud, safety failures or unreconciled records.

Recovery point and recovery time are different constraints. The newest clean copy determines how much recent work may be lost. The speed of rebuilding determines how long the service remains unavailable. A business can preserve every transaction and still fail because restoration takes longer than customers, patients or cash flow can tolerate. Both targets need tests against the full dependency chain.

The organisation then watches for recurrence. Logs, indicators and behaviour are checked across the environment. Restored systems may be monitored more closely. People who know the business confirm that transactions, balances, schedules and production make sense. Technical health is necessary, but an application can run while its data or business process remains wrong.

Learning closes the cycle. The useful review is not a hunt for the person who made the first mistake. It reconstructs the chain: which asset mattered, which path opened, which control failed, which control limited damage, which signal arrived and which recovery assumption proved false. Actions receive owners, dates and evidence of completion. Otherwise the post-incident report becomes a polished account of why the next incident will look familiar.

How we know

Cybersecurity evidence is uneven. Public incident reports overrepresent large organisations, regulated sectors and events that became visible. Victims may not know the full intrusion, may withhold detail during litigation or investigation, and may describe the event through the controls they later changed. Vendor reports contain useful telemetry but reflect their customers, products and definitions rather than the whole world.

This account relies on mechanisms that recur across official frameworks, incident investigations and security engineering, while keeping case claims within their documented scope. The Morris Worm figures come from the FBI's historical account. The Equifax sequence and affected population come from the US Government Accountability Office. The WannaCry effects and preparedness findings come from the UK National Audit Office. Current guidance on identity, incident response and ransomware recovery comes principally from NIST and the UK National Cyber Security Centre.

Attack frequencies change with reporting populations, technology and definitions; this book does not turn them into a personal probability of compromise. A framework can organise work without proving that one control caused a particular outcome. The strongest evidence here concerns observable attack paths, permissions, recorded events and recovery tests. Claims about deterrence, attacker choice and organisational behaviour remain conditional on setting.

What People Get Wrong

“Only important people are targeted”

Targeted attacks exist, and some people face opponents willing to study them for months. That fact has produced the comforting inverse: if nobody knows your name, nobody will bother.

Automation breaks the logic. Internet services can be scanned without caring who owns them. Credentials leaked from one site can be tested against thousands of others. Fraud messages can be sent until the small fraction that arrive at the right moment repay the campaign. The target may be selected after the weakness is found.

Scale also changes what targeting means. A campaign may begin with a software version, exposed service or leaked credential set, then discover the organisation behind it. The victim is not chosen first. The vulnerable route is, and whoever owns it becomes the target.

Ordinary accounts also have value. Email can reset other services. A social account can impersonate its owner. A business mailbox can redirect an invoice. A home computer can supply stored credentials or a route into work. The attacker does not need your biography; the account's powers are enough.

The correction matters because sensible basic controls are aimed at scale. Unique credentials, passkeys or additional factors, updates and protected recovery routes make cheap repetition less productive. Being unimportant may reduce tailored attention. It does not make exposed access worthless.

“Hackers usually break the encryption”

The padlock dominates security imagery because it turns protection into a visible contest of mathematics. Modern encryption can fail through bad algorithms, weak implementation, exposed keys or outdated settings. Yet direct defeat of sound cryptography is often the expensive route.

Attackers prefer the ends. They steal the credential before encryption

begins, take the data after an authorised user opens it, compromise the device holding the key, abuse a valid session or persuade the owner to send the information voluntarily. A perfectly encrypted backup is useless if the attacker can authenticate to the console and delete it. An encrypted connection to a fraudulent website protects the fraudster from eavesdroppers too.

This does not make cryptography unimportant. It makes its boundary clear. Encryption protects specified data against specified access while the keys and endpoints remain controlled. It does not decide whether the person at either end should be trusted.

The correction redirects effort. Use strong, maintained cryptography rather than inventing it, then protect identity, devices, permissions and recovery. The strongest lock still depends on who holds the key and what waits behind the door.

“Multi-factor authentication makes an account safe”

Adding a second factor blocks many attacks that stop at a stolen password. It became popular because the improvement is large and easy to explain. The slogan hardened into an absolute claim the technology never made.

Traditional methods can still be phished. A false site may relay a time-based code. A push request may be approved through confusion or repeated prompts. Text messages depend on control of a phone number. Malware may steal an authenticated session after the factors have done their job. Recovery staff may accept weaker evidence than the normal login.

Phishing-resistant methods such as correctly implemented passkeys improve the exchange because a false service cannot collect a reusable credential in the usual way. They still depend on device security, enrolment, synchronisation and recovery. No authenticator protects an action that the authorised user is persuaded to perform.

The correction is not to abandon multi-factor authentication. Prefer

phishing-resistant options where available, use other second factors where they are not, and inspect the whole account lifecycle. The weakest route may be the one labelled “I cannot log in”. Review registered authenticators, active sessions and fallback methods after any suspected compromise, because the attacker may have added a way back.

“A careful person can spot every phish”

Many fraudulent messages contain clues, so advice naturally teaches people to inspect senders, links, spelling and urgency. The danger is turning useful suspicion into a promise of perfect detection.

A message can come from a compromised real account, continue an existing conversation and use information only the participants would expect. A false login page can copy the legitimate design. A voice caller can know names and recent transactions. Pressure narrows attention, while ordinary work trains people to respond quickly to colleagues, customers and systems.

Evidence from real incidents and exercises shows that phishing remains a recurring entry route despite years of awareness campaigns. That does not prove people are foolish. It proves that judging authenticity from presentation alone is an adversarial task with uneven information.

The stronger defence attaches checks to consequential actions. Open the service through a known route. Confirm changed payment details using a stored number. Require another approval. Make reporting quick and non-punitive. Filters, secure authentication and limited permissions reduce what one judgement must carry. Care matters. A process that survives ordinary human limits matters more.

“Public Wi-Fi is the main danger”

Early web traffic often crossed local networks without encryption, so the stranger in the café became the standard warning. The image survived after the web changed. Widespread HTTPS now encrypts most ordinary connections between a device and the service, which makes passive local snooping far less useful than it once was.

Public networks can still be malicious or badly configured. A false hotspot can misdirect users, devices can expose local services, and an attacker may exploit unpatched software. Yet the lock symbol proves only an encrypted connection to the site reached. A fraudulent site can use HTTPS. Logging into it remains a secure delivery of credentials to the wrong party.

A consumer VPN protects the path to the VPN provider and can hide traffic from the local network. It does not create anonymity. It moves visibility and trust towards the provider, while websites can still recognise logins, cookies and information supplied by the user.

The correction changes priorities. Keep devices updated, verify the service, use strong account protection and distrust unexpected requests on every network. Do not treat mobile data as purity or a VPN subscription as absolution.

“Backups solve ransomware”

Backups are among the strongest defences against destructive loss, which is why attackers search for them. The myth appears when making a copy is confused with having a recoverable system.

A backup may use the same administrative identity as production, remain permanently reachable, overwrite clean history or contain malware and corrupted data. Restoration may depend on unavailable keys, software, documentation or specialists. A small test file proves little about rebuilding an identity platform, database and working application within the time the organisation can survive.

Ransomware has also become an extortion model that may include stolen

information. A clean copy can reduce pressure created by encryption. It cannot retrieve confidentiality once data have left, stop publication or prove that the attacker has gone.

The correction is to protect backups as a separate security domain. Preserve versions, restrict and monitor deletion, use distinct strong authentication, keep resistant copies and test restoration. Then plan identity, communications and operations around them. A backup is material for recovery. Recovery is the demonstrated ability to resume trusted work.

“Cybersecurity belongs to the IT department”

The technology team operates many controls, so handing it the problem feels efficient. The arrangement fails when security choices are business choices wearing technical clothes.

Only service owners know which outage is tolerable, which data are necessary, which supplier is replaceable and which manual process is safe. Finance designs payment approval. Human resources controls joining and leaving. Procurement grants dependencies. Leaders decide whether patching may interrupt revenue, whether recovery deserves funding and how much concentration to accept. IT cannot settle those trade-offs alone.

This is also why buying a tool cannot transfer accountability. A security service can monitor events, but somebody must decide what to contain. An insurer can absorb some financial loss, but it cannot restore trust, recreate unavailable records or run the organisation during an outage. A supplier can operate infrastructure while the customer retains decisions about identities and data.

The correction assigns cyber risk to the owners of consequence, supported and challenged by specialists. When everyone owns an undefined share, nobody owns the decision. Clear authority before the incident is itself a control.

Use It

Price the loss before buying the control

Security shopping reverses the order of thought. A product names a threat, demonstrates a frightening possibility and offers relief. Start instead with the event that would hurt.

For a person, the loss may be the primary email account, money, private photographs or family files. For a small company it may be redirected supplier payments, unavailable order systems, stolen customer data or a week without dispatch. State the consequence, then describe the shortest plausible path to it.

This exposes misplaced effort. A household may buy a consumer VPN while reusing its email password. A company may buy advanced monitoring while former staff retain accounts. The visible threat is addressed; the cheap path remains.

The comparison becomes useful when it changes the next task. For the household, securing email may close several recovery routes at once. For the company, withdrawing former staff access may matter more than another dashboard. The decision is now about a specific loss and an affordable interruption to its path, rather than whichever threat had the best advertisement.

Protect the account that can recover the others

Draw the recovery chain. Which account receives reset messages? Which device approves the login? Which phone number or support process restores access? Many plans protect each service while leaving the chain that controls them weak.

Start with primary email and the password manager. Give each a distinct, strong access method. Prefer passkeys where supported and the device or synchronisation account is protected. Otherwise use generated passwords and an additional factor. Store recovery codes away from the device they

recover. Review sessions and recovery details when devices or authorised users change. Where abuse is a concern, get safety advice before making changes that could alert the other person.

Businesses should apply the same logic to identity administrators, domain registration, cloud control, backups and finance. A recovery administrator may be more powerful than an ordinary one because it decides who is recognised after normal proof fails. Avoid making that recovery depend on one unavailable person.

Protect the route that creates new keys before polishing every door.

Make one mistake small

Assume a message will be believed, a laptop will be lost and one application will be compromised. Then ask what the event can reach without another decision.

Use ordinary accounts for ordinary work. Keep administrative access separate and temporary. Restrict shared folders. Give suppliers access to the service they support, not the whole network. Limit bulk exports where individual records are enough, while remembering that readable data can still be copied. Require a second person for large payments or bank-detail changes. Keep backups behind another administrative boundary.

This is defence in depth without the product catalogue. Each boundary contains a known consequence: an identity check, permission, network rule, approval or recovery copy. Strong boundaries force the attacker across different barriers. Repeating one stolen credential against five screens is one barrier drawn five times.

Containment also limits ordinary mistakes. A deleted folder, misdirected email or bad update affects less. Security and reliability meet because both care how far one failure travels. Ask what remains possible after the control fails.

Prefer controls that keep working

Attention is scarce. A plan that needs everyone to remember a warning at the right second will decay under pressure. Prefer controls that remain active while people are busy.

On a personal device, automatic updates and a screen lock carry protection through an ordinary distracted day. A password manager supplies different credentials without asking you to remember them all. These controls do different jobs: the screen lock helps when the device leaves your hands; the manager helps when a password leaks elsewhere. Neither depends on noticing the right clue in an incoming message.

Automation needs ownership. Updates can break old applications, filters can block mail and expiring privilege can interrupt work. Use feedback and an exception route. Monitor failure, make legitimate access attainable and remove exceptions when the need ends.

Use the same lens on vendors. Does the product arrive safely configured? Are updates supported? Can administrators use strong authentication? Are useful logs available? Can data and credentials be removed at exit? The strongest feature may be the one that removes security work from memory.

Ask what would tell you

For each serious loss, imagine prevention failed. What signal would show the path while there was still time to act?

An individual might receive a login alert, reset message, new-device notice or bank notification. These matter only if the channel is trusted and the response is clear. Open the service through its known app or address, not a link in the alert. Use a device you have reason to trust. Follow the provider's recovery process, end unfamiliar sessions and check recovery details and email-forwarding rules as well as the password. A login alert is a decision point, not decoration.

A company should identify high-value events: new administrators, changed authentication, unusual exports, disabled security tools, backup deletion,

mass file changes and unexpected access. The list depends on the threat model. Somebody must receive the signal, understand it and have authority to act.

Test detection with evidence. Can the team find who changed supplier bank details, trace a privileged login or notice backup deletion? If not, the organisation has accepted blindness. That may be tolerable for low-consequence activity, but it should be conscious.

Rehearse the first hour

At home, rehearse losing the phone that approves your important logins. For a business, imagine several devices show ransom notes. Do not solve the entire incident. Work through the first hour.

Who is told? How do they communicate if normal systems are suspect? Who can isolate accounts or devices? Which services must continue? Who records decisions? Which outside contacts are needed? A short discussion exposes dependencies a written plan conceals.

Keep the result usable. Store contacts and initial instructions outside the affected network. Define who may order containment and emergency spending. Give staff a reporting route and fallback. Record where recovery credentials are protected. Test a restore rather than trusting a green dashboard.

Repeat after major changes, incidents and supplier transitions. The exercise prepares decisions and communication, not a prediction of the attack. Rehearsal cannot remove pressure, but it can stop the first hour being spent discovering who may act.

The limits

No general book can assign your threat model. A person facing abuse may need to treat shared devices, location services and recovery as immediate safety issues. Changing settings or removing monitoring software can alert the abuser. A safer device and a specialist safety plan may need to come before ordinary account housekeeping. Journalists, activists, political figures, administrators and people controlling money or sensitive systems may face tailored attacks that justify specialist help. A small business and a power network need different depths of control.

Security competes with usability, privacy, cost and openness. Monitoring can become surveillance. Tight access can drive staff towards unofficial tools. Rapid patching can disrupt critical equipment. Identity checks can exclude legitimate users. Every control reallocates power and friction. Making security supreme produces systems nobody can use or trust.

Some losses remain irreversible. Published information cannot be made secret again. A fraudulent payment may move before it is understood. Recovery may restore operation without confidence. Insurance moves part of the financial cost and leaves other consequences behind.

The honest promise is reduction and resilience, not immunity. When the stakes or opponent exceed your competence, escalate early rather than decorating uncertainty with confidence.

The one thing to keep

The useful difference is between living with risk and giving it the whole day.

Return to the lost phone. At first it looked like a missing object. Now you can see the accounts it approves, the information it holds and the recovery arrangements that might depend on it. That extra understanding need not make the loss more frightening. It gives you places to act before it happens. A way back that does not depend on the missing device can turn a crisis into an inconvenience.

The same change in sight applies to a company. An inbox is an authority. A

supplier connection is a permission. A backup is a claim about what will still be possible after failure. Once those claims are visible, they can be tested. You no longer have to choose between trusting a reassuring logo and suspecting everything that moves.

There will still be alarming headlines. Some should change what you do; others describe a threat with no useful bearing on your circumstances. The distinction is earned by knowing what matters, not by knowing every attack. After a sensible precaution is in place, checking the same fear repeatedly does not necessarily add protection.

The point of securing an account is to use it. The point of preserving photographs is to keep them, not to spend every evening worrying about their disappearance. Good security leaves enough of your attention available for the life it is meant to protect.

Terms

Asset. Anything whose loss, alteration, exposure or unavailability would matter, including data, accounts, devices, money, reputation, personal safety and the ability to provide a service. Consequence determines priority.

Threat. A circumstance or actor capable of causing harm. Useful planning distinguishes motives, capabilities and routes rather than treating criminals, insiders, stalkers and states as one class.

Vulnerability. A weakness that could permit unintended access or action. It may exist in software, configuration, process, physical protection or human interaction, and matters in relation to exposure and consequence.

Exploit. A method that uses a vulnerability to produce an unintended result. It may be code, crafted input or a repeatable sequence of actions. A weakness and a practical exploit are different conditions.

Risk. The possibility of harm, considered through likelihood, consequence and uncertainty. Security decisions usually compare plausible losses rather than claim clean numerical precision. The comparison should still support

proportionate action.

Attack surface. The exposed points through which outside or untrusted activity can reach a system. Accounts, applications, devices, interfaces, suppliers and recovery routes all add surface. Removing unused exposure reduces both attack opportunity and defensive work.

Threat model. A structured account of what matters, who might attack it, how the attack could work and which controls change the path. It sets priorities and boundaries rather than predicting every incident.

Control. A safeguard that reduces likelihood, consequence or uncertainty. Controls may prevent, detect, contain or support recovery. A written policy counts only when it changes behaviour and its operation can be shown.

Defence in depth. Different protective layers arranged so one failure does not decide the outcome. Strong layers address separate conditions such as identity, permission, movement, detection and recovery.

CIA triad. Confidentiality, integrity and availability. The model asks whether information is kept from unauthorised disclosure, protected from improper change and usable when required. Different systems need the three in different proportions.

Authentication. Establishing that a claimant controls an accepted credential to the required confidence. It answers who or what is logging in, not what that identity may do.

Authorisation. Deciding which actions an authenticated identity may perform. Roles, policies and permissions implement it. Strong login with broad authorisation can still create wide harm.

Credential. Evidence used to authenticate, such as a password, key, device or token. Creation, storage, use, recovery and revocation can each become the weakest stage.

Session. A period of accepted access after authentication. Session tokens spare repeated login, but stealing one may let an attacker bypass the

original challenge.

Password manager. Software that creates and stores distinct credentials. It reduces reuse and memory pressure, while making its primary access, devices and recovery method especially important.

Passkey. A FIDO credential bound cryptographically to the legitimate service. It removes a reusable password from the login exchange and resists common phishing, while device, synchronisation, enrolment and recovery security still matter.

Multi-factor authentication. Authentication using more than one factor category, commonly knowledge, possession or inherence. It blocks many password-only attacks, although traditional codes and prompts may remain phishable.

Phishing. Deceptive communication intended to obtain credentials, run code, transfer money or trigger another unsafe action. It can use email, text, voice, messaging or a compromised genuine account.

Social engineering. Manipulating people or processes to obtain access, information or action. It exploits trust, authority, urgency and helpfulness, so consequential requests need independent checks.

Malware. Software designed for harmful or unauthorised action, including stealing information, obtaining credentials, disrupting devices or creating remote control. The label does not reveal the final objective.

Ransomware. Malware that denies access to systems or data, often through encryption, to demand payment. Attacks may also involve data theft and extortion. Backups cannot reverse disclosure.

Patch. A software change intended to correct a flaw or improve operation. Security patching closes known routes, with priorities shaped by exposure, active exploitation and consequence.

Zero-day. Usually a flaw unknown to its supplier, or an attack using a flaw before a fix is available. Usage varies. Novelty alone does not determine

severity; a known weakness left exposed can be equally damaging.

Least privilege. Granting only the access required for a defined task and duration. It reduces damage from compromise and mistakes, provided legitimate access remains attainable.

Privilege escalation. Gaining authority beyond the initial account or process through software flaws, stolen credentials or excessive permissions. It helps attackers disable controls, reach assets or persist.

Segmentation. Dividing systems and access paths so entry to one area does not confer broad movement. Effective segments contain failure and create further decisions and evidence.

Zero trust. An access model that rejects broad trust based on network location or ownership alone. Requests are evaluated for a resource using identity, device, context and narrow permissions.

Log. A record of an event such as a login, permission change or process start. Logs help only when relevant events are captured, protected, retained and reviewed in time.

Backup. A separate copy used to restore lost or damaged data and systems. A useful backup preserves clean history, resists the production incident and passes realistic restoration tests.

Incident response. Organised preparation, detection, analysis, containment, removal, recovery and learning after a security event. It combines technical action, business decisions and communication under incomplete information, which makes preassigned authority part of the control.

Go Deeper

The human investigation

Clifford Stoll, *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage* (1989; Gallery Books reissue, 2024). A small accounting discrepancy leads Stoll through a long investigation of unauthorised access at Lawrence Berkeley Laboratory. The machines, modems and institutional setting are historical. The habits are not: follow weak signals, preserve evidence, understand normal behaviour, coordinate across reluctant organisations and resist the urge to confuse a plausible story with proof. It is the most inviting book here and shows one defender learning the work while doing it, with curiosity carrying him beyond his original job description. Read it for the investigation's patience rather than for current technical procedure.

The complete systems view

Ross Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, third edition (2020). This is the large book behind the small one: a wide treatment of protocols, access control, fraud, privacy, security economics, psychology, industry systems and sustainable defence. Anderson keeps returning to the incentives and operating conditions that make technically respectable designs fail. It is over a thousand pages and works better as a reference or a sequence of chosen chapters than as a first weekend read. Use it when a security question looks technical but the failure sits in people, markets or institutions. The case studies reward selective reading because each exposes a different boundary between mechanism and governance.

The design method

Adam Shostack, *Threat Modeling: Designing for Security* (2014). Read this to turn “something bad might happen” into a structured design conversation. Shostack explains ways to model systems, identify threats and choose mitigations without requiring the exercise to become a specialist ritual. Its software-development setting is narrower than this book’s household and organisational scope, and some examples reflect the technology of its date. The durable value is procedural: make the system visible, ask what can go wrong, decide what to do and check whether the work was done. Teams can use that sequence long after the named tools have changed.

The destructive edge

Andy Greenberg, *Sandworm: A New Era of Cyberwar and the Hunt for the Kremlin's Most Dangerous Hackers* (2019). Greenberg follows researchers and victims through attacks linked to a Russian military intelligence unit, including disruption in Ukraine and the global damage caused by NotPetya. Read it for the point where code, physical infrastructure, geopolitics and commercial dependence meet. It is narrative journalism rather than a general security manual, and attribution in cyber conflict always deserves careful handling. Its strength is consequence: systems built for ordinary efficiency can carry destructive effects far beyond the intended target.

Notes and Sources

Scope, model and terminology

The book treats cybersecurity as risk management for connected systems rather than as a synonym for secrecy or technical intrusion. The confidentiality, integrity and availability triad supplies the basic categories of harm. The broader structure draws on the NIST Cybersecurity Framework 2.0, NIST incident-response guidance, NIST digital-identity guidance, NIST zero-trust guidance, the design principles collected by Saltzer and Schroeder, and the systems approach in Ross Anderson's *Security Engineering*. Those sources organise security differently and are not presented as one settled formula. The manuscript's sequence of loss, route, authority, visibility and recovery is an explanatory synthesis for a general reader.

Terms such as threat, vulnerability, exploit, risk and attack surface vary across standards and professional communities. The definitions here preserve the distinctions needed by the book without claiming one universal vocabulary. Risk is deliberately treated as a ranked judgement under uncertainty rather than a precise product of two known numbers.

The Whole Thing in One Page and Why You Should Care

The WannaCry account follows the UK National Audit Office's 2017 investigation. It records that at least 81 of 236 English NHS trusts were affected, alongside 603 primary-care and other NHS organisations; NHS England identified 6,912 cancelled appointments and estimated that more than 19,000 were cancelled in total; five accident and emergency departments were unable to treat some patients. The report also found unpatched or unsupported systems among affected organisations and stated that a national response plan had not been tested at local level. The text does not claim that WannaCry altered patient data, that every cancellation was observed directly or that the appointment tally measures clinical harm. The five accident and emergency departments diverted patients.

The claim that many attacks can be repeated cheaply is supported by current government and agency threat reporting, but no incident dataset measures the whole population of attempts. The UK *Cyber Security Breaches Survey 2025/2026* covers attacks recognised and reported by sampled organisations, with fieldwork from August to December 2025. ENISA's *Threat Landscape 2025*, published in October 2025 and updated to version 1.2 in January 2026, analyses 4,875 incidents reported or otherwise collected for the period 1 July 2024 to 30 June 2025. Both are bounded checks on threat reporting, not complete measures of attack frequency or universal sector risk.

The passkey comparison follows the UK National Cyber Security Centre's April 2026 paper on traditional credentials and FIDO2 credentials for personal use. It supports the narrower claim that correctly implemented FIDO2 credentials resist common credential phishing and reuse better than passwords and traditional phishable second factors. The manuscript keeps the paper's limits visible: device access, synchronisation, enrolment, recovery and active sessions remain relevant, and consumer use does not represent every enterprise deployment.

Core Idea 1: loss, threat models and proportion

Adam Shostack's *Threat Modeling* supplies the durable design sequence of making a system visible, asking what can go wrong, deciding what to do and checking that the work was completed. NIST CSF 2.0 provides a non-prescriptive taxonomy for governing, identifying, protecting, detecting, responding and recovering. Neither source establishes a universal numerical method for risk. The book therefore rejects false precision while retaining the practical need to rank plausible harm.

The examples of households, small firms, hospitals, insiders, abusive partners, criminals and states are scope examples rather than claims that each group faces one standard threat profile. People at heightened personal or political risk require tailored advice. The text avoids turning a population-level guide into a promise against a patient and well-resourced opponent.

Core Idea 2: identity, authenticators and sessions

NIST SP 800-63B-4 is the principal standards source for authentication and authenticator management. It distinguishes authenticator assurance, phishing resistance, lifecycle controls and recovery. It supports the separation among authentication, authorisation and session management used in the text. The manuscript does not equate a biometric with a remotely stored reusable secret: in common passkey designs, a local biometric or device PIN permits use of the credential held by the device or credential provider.

The NCSC FIDO2 comparison supports the claims about password reuse, real-time phishing of traditional second factors, phishing resistance and the remaining importance of endpoint and recovery security. Text-message codes, time-based codes and approval prompts differ in their weaknesses. The book groups them only where the shared point is that an extra factor can block password-only attacks without making every login exchange phishing-resistant.

The statement that email often acts as a recovery hub is an operating

observation, not a rule for every service. Financial and high-assurance systems may use separate recovery controls. The practical instruction is to identify whichever account or process can re-establish access to the rest.

The passkey description uses the broader distinction between synchronised and device-bound credentials. FIDO2 with user verification can combine possession and a local activation factor in one action. It does not inherently require a further SMS challenge. Providers differ in how synchronised credentials are protected and recovered; compromise of an ordinary account password is not treated as automatic possession of every passkey.

Core Idea 3: exposure, patching and dependencies

The distinction between a vulnerability and an exploit follows standard security usage. The treatment of patching is intentionally risk-based. An available correction can close a known path, but operational technology, medical devices and fragile legacy applications may require testing, isolation or replacement rather than an unexamined immediate change. The text does not use this constraint to excuse indefinite exposure.

The SolarWinds example follows the original FireEye/Mandiant investigation published on 13 December 2020, now hosted by Google Cloud. Malicious code was distributed through a trusted software-update route, showing that signed delivery can authenticate the source and integrity of a package as issued while the issuer's build or distribution process is itself compromised. The example establishes dependency risk; it does not imply that every affected customer was exploited further or suffered the same consequence.

Inventory, support status, configuration, supplier permissions and concentration are treated as parts of the attack surface because they affect reachable paths and the defender's ability to act. No claim is made that one asset inventory or supplier questionnaire proves control of the underlying system.

Core Idea 4: privilege and containment

Least privilege and separation of privilege are among the design principles in Saltzer and Schroeder's 1975 paper and remain central to Anderson's systems treatment. NIST SP 800-207 rejects implicit trust based solely on network location or device ownership. The book uses these principles to explain containment, not to claim that segmentation or a zero-trust product guarantees it. Restricting bulk export can limit scale but cannot make information impossible to copy once a user is allowed to read it.

Machine identities need ownership, limited permissions and controlled credential replacement or expiry. This is distinct from forcing routine password changes on human users. NIST SP 800-63B-4 prohibits verifiers from requiring periodic password changes and requires a change when there is evidence of compromise. In a confirmed incident, revocation and replacement also depend on which credentials and sessions were exposed.

Core Idea 5: human decisions and verification

The human discussion draws on security engineering, fraud controls and current identity guidance rather than on a claim that one psychological trait explains deception. Urgency, authority, familiarity and helpfulness can influence action, but attacks vary and people vary. Generative tools can reduce the cost of producing convincing text, voice or imagery, yet the practical defence remains grounded in the action requested: use a trusted route for consequential changes.

Independent verification means that the requester does not control both the demand and the checking channel. A number supplied inside the suspicious message is not independent. The text does not claim that calling always solves impersonation; stored details, known application routes, dual approval and delay rules each address different processes.

Phishing simulations and training can provide useful practice, but click rate alone is an incomplete outcome measure. Reporting speed, the quality of the response, protection around high-risk actions and the reach of one error matter to the resulting loss. This is a design judgement supported by the

wider systems literature rather than a universal experimental effect size.

Core Idea 6: detection and incident response

NIST SP 800-61 Rev. 3 is the main current source for integrating incident response with risk management. It supports preparation, detection, analysis, containment, removal, recovery and learning as overlapping work rather than as a perfectly linear checklist. The manuscript's exact wording is its own synthesis and does not imply that all organisations use the same team structure.

Logs are described as evidence only when relevant events are captured, protected, retained and reviewed. Central collection and restricted access can preserve useful history against local compromise, but no logging design guarantees detection. Baselines and alerts are context-dependent, and increased monitoring can create privacy, legal, cost and security burdens. The text therefore links collection to a defined threat model rather than treating surveillance volume as protection.

The distinction between isolating a network connection and removing power follows NIST SP 800-86, particularly its incident-response and volatile-data discussions. Both interventions can alter evidence; powering down additionally loses volatile memory. This older source is used for that stable forensic distinction, not as a current list of tools or a universal shutdown instruction. Operational safety and continuing harm determine the decision.

Core Idea 7: backups, continuity and recovery

The NCSC's *Ransomware-resistant Backups* supports the claims that backup data can be targeted, that separation and administrative controls matter, and that organisations must test and monitor the health of the backup regime. A ransomware-resistant backup design does not prevent data theft, protect every surrounding service or determine the order in which operations should return.

Recovery time and recovery point are explained without imposing one target. The necessary values depend on the service and consequence. A

restore test that proves one file can be opened does not establish that identities, applications, keys, configurations, suppliers and staff procedures can restore an operating service at the required scale.

Operating sequence and documented cases

The Morris Worm chronology and scale follow the FBI's historical account: around 8.30 p.m. on 2 November 1988, the program began spreading; within twenty-four hours an estimated 6,000 of roughly 60,000 Internet-connected computers had been affected. The figure is an attributed historical estimate, not a modern census. The worm targeted particular Unix systems through several routes and caused repeated copies to consume resources. It did not need to destroy files to disrupt work.

The Equifax account follows the US Government Accountability Office report GAO-18-559. Attackers gained access through the online dispute portal and reached personal information belonging to at least 145.5 million people. Equifax's own investigation identified four major factors: identification, detection, segmentation of database access and data governance. The book uses the case to show how an opening and its surrounding controls jointly determine consequence; it does not assign equal causal weight to each factor.

The WannaCry operating sequence and NHS consequences follow the National Audit Office investigation described above. The attack is used as a bounded example of known exposure, propagation, interruption and response preparation in English health organisations. It is not presented as a universal model of ransomware or healthcare security.

The generic attack path combines recurring stages found across incident-response guidance and documented cases: reconnaissance or route discovery, initial access, privilege expansion, movement, collection, objective and defender response. Real incidents can omit, repeat or reorder these stages. The sequence explains dependencies rather than promising that every attacker follows one named framework.

The denial-of-service counterexample follows the NCSC collection's

explanation of exhausted bandwidth, processing and other service resources, together with its guidance on upstream defences. The example is intentionally outside the intrusion sequence: loss of availability need not imply stolen credentials or access to internal systems.

What People Get Wrong

The encryption correction preserves the boundary with *Cryptography in a Hurry*. Modern cryptographic mechanisms are often strong enough that attackers choose credentials, endpoints, sessions, recovery processes or authorised human actions instead. This does not mean cryptographic failure is unimportant or impossible. It means confidentiality in transit does not establish trust in the destination, integrity of an endpoint or legitimacy of the action.

The public Wi-Fi correction follows the US Federal Trade Commission's consumer guidance: widespread HTTPS makes ordinary use of public networks safer than the older warning implies, while an encrypted connection to a fraudulent site remains harmful. The VPN correction follows FTC guidance that consumer VPN applications route traffic through provider-controlled servers and therefore transfer visibility and trust rather than create anonymity.

The backup correction follows NCSC guidance and current ransomware practice. Protected copies can reduce destructive leverage. They cannot reverse prior disclosure, remove malicious access, restore trust in every system or decide whether extortion payment is lawful, safe or effective.

Use It: personal recovery and safety

The FTC's account-recovery guidance supports securing the device first, using the provider's recovery process, ending sessions and checking recovery details and forwarding rules. The FTC's June 2026 stalkerware guidance warns that attempts to cut off access may alert an abuser and recommends a safer device and specialist safety planning. The book borrows that safety principle, not US reporting procedures or a promise that any one reset makes a person safe.

The lost phone, accounts clerk and short examples of business permissions are illustrative, not reported cases. They explain mechanisms rather than prove attack frequency. Synchronisation can propagate deletions; whether earlier versions remain recoverable depends on the service and its retention settings. Apple documents this behaviour for iCloud Photos, including its time-limited recovery facility.

Current evidence and its limits

Current guidance and consequential claims were checked for this edition on 5 September 2026. The newest important UK reference used is the *Cyber Security Breaches Survey 2025/2026*, published on 30 April 2026 and based on fieldwork from August to December 2025 about the preceding twelve months. The newest authentication source used is the NCSC FIDO2 comparison published on 23 April 2026. NIST SP 800-63B-4 was published in July 2025, with the final publication history dated 31 July; NIST SP 800-61 Rev. 3 appeared in April 2025.

Security incident evidence is selection-biased. Undetected events are absent, private organisations disclose unevenly, definitions differ and geopolitical reporting is incomplete. The manuscript therefore avoids a global probability that an individual or firm will be attacked, and it does not rank threats from incompatible datasets.

Go Deeper verification

Ross Anderson's third edition of *Security Engineering* was published by Wiley in 2020. Adam Shostack's *Threat Modeling* was published by Wiley in 2014. Clifford Stoll's *The Cuckoo's Egg* dates from 1989; the cited Gallery Books reissue appeared in July 2024. Andy Greenberg's *Sandworm* was first published in hardback by Doubleday in 2019. The recommendations have distinct purposes: systems reference, design method, defender investigation and destructive state-linked operations.

Bibliography

Standards, guidance and official investigations

Apple. *Delete Photos on Your iPhone or iPad*. Apple Support, 3 April 2026.

Department for Science, Innovation and Technology and Home Office. *Cyber Security Breaches Survey 2025/2026*. London: UK Government, 2026.

European Union Agency for Cybersecurity. *ENISA Threat Landscape 2025*. ENISA, 2025. Version 1.2, updated 9 January 2026.

Federal Bureau of Investigation. *Morris Worm*. FBI historical case account.

Federal Trade Commission. *Are Public Wi-Fi Networks Safe? What You Need to Know*. Washington, DC: Federal Trade Commission, 2023.

Federal Trade Commission. *How to Recover Your Hacked Email or Social Media Account*. August 2023.

Federal Trade Commission. *In the Market for a VPN App?* Federal Trade Commission, 2018.

Federal Trade Commission. *Stalkerware: What to Know*. June 2026.

FireEye/Mandiant. *Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor*. 13 December 2020. Original investigation, now hosted by Google Cloud.

National Audit Office. *Investigation: WannaCry Cyber Attack and the NHS*. HC 414, Session 2017-19. London: National Audit Office, 2017.

National Cyber Security Centre. *Comparing the Security Properties of Traditional User Credentials and FIDO2 Credentials for Personal Use*. London: National Cyber Security Centre, 2026.

National Cyber Security Centre. *Denial of Service (DoS) Guidance: “Understand Your Service” and “Upstream Defences.”* Published 2019; reviewed 25 March 2024.

National Cyber Security Centre. *Ransomware-resistant Backups*, including the on-premises and cloud principles. Version 1.0, 22 November 2024.

National Institute of Standards and Technology. *Digital Identity Guidelines: Authentication and Authenticator Management*. NIST SP 800-63B-4. Gaithersburg, MD: National Institute of Standards and Technology, 2025. doi: 10.6028/NIST.SP.800-63B-4.

National Institute of Standards and Technology. *Guide to Integrating Forensic Techniques into Incident Response*. NIST SP 800-86. 2006. doi: 10.6028/NIST.SP.800-86.

National Institute of Standards and Technology. *Incident Response Recommendations and Considerations for Cybersecurity Risk Management: A CSF 2.0 Community Profile*. NIST SP 800-61 Rev. 3. Gaithersburg, MD: National Institute of Standards and Technology, 2025. doi: 10.6028/NIST.SP.800-61r3.

National Institute of Standards and Technology. *The NIST Cybersecurity Framework (CSF) 2.0*. NIST CSWP 29. Gaithersburg, MD: National Institute of Standards and Technology, 2024. doi: 10.6028/NIST.CSWP.29.

Rose, Scott, Oliver Borchert, Stu Mitchell and Sean Connelly. *Zero Trust Architecture*. NIST SP 800-207. Gaithersburg, MD: National Institute of Standards and Technology, 2020. doi: 10.6028/NIST.SP.800-207.

United States Government Accountability Office. *Data Protection: Actions Taken by Equifax and Federal Agencies in Response to the 2017 Breach*. GAO-18-559. Washington, DC: US Government Accountability Office, 2018.

Research and books

Anderson, Ross. *Security Engineering: A Guide to Building Dependable Distributed Systems*. 3rd ed. Indianapolis: Wiley, 2020.

Greenberg, Andy. *Sandworm: A New Era of Cyberwar and the Hunt for the Kremlin's Most Dangerous Hackers*. New York: Doubleday, 2019.

Saltzer, Jerome H., and Michael D. Schroeder. "The Protection of

Information in Computer Systems.” *Proceedings of the IEEE* 63, no. 9 (1975): 1278-1308. doi: 10.1109/PROC.1975.9939.

Shostack, Adam. *Threat Modeling: Designing for Security*. Indianapolis: Wiley, 2014.

Stoll, Clifford. *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*. Gallery Books, 2024 reissue of the 1989 work.