Security 101

A Beginner’s Guide to Cybersecurity and Zero Trust for Developers

Security 101

A Beginner’s Guide to Cybersecurity and Zero Trust for Developers

Agenda:
In this session, we will delve into fundamental cybersecurity concepts, explore the concept of zero trust, and discuss key themes across various security domains. This session is especially for developers and will focus on how they can improve their company's security posture without having security slowing them down. The session will cover the following topics:

Basic Cybersecurity Concepts:
- Understanding the CIA triad: Confidentiality, Integrity, and Availability.
- Differentiating between risks and threats.
- Exploring the role of security controls in safeguarding systems and data.

Zero Trust: A Paradigm Shift
- Defining zero trust and its principles.
- Why zero trust matters in today’s interconnected and dynamic digital landscape.
- How to implement zero trust strategies effectively.

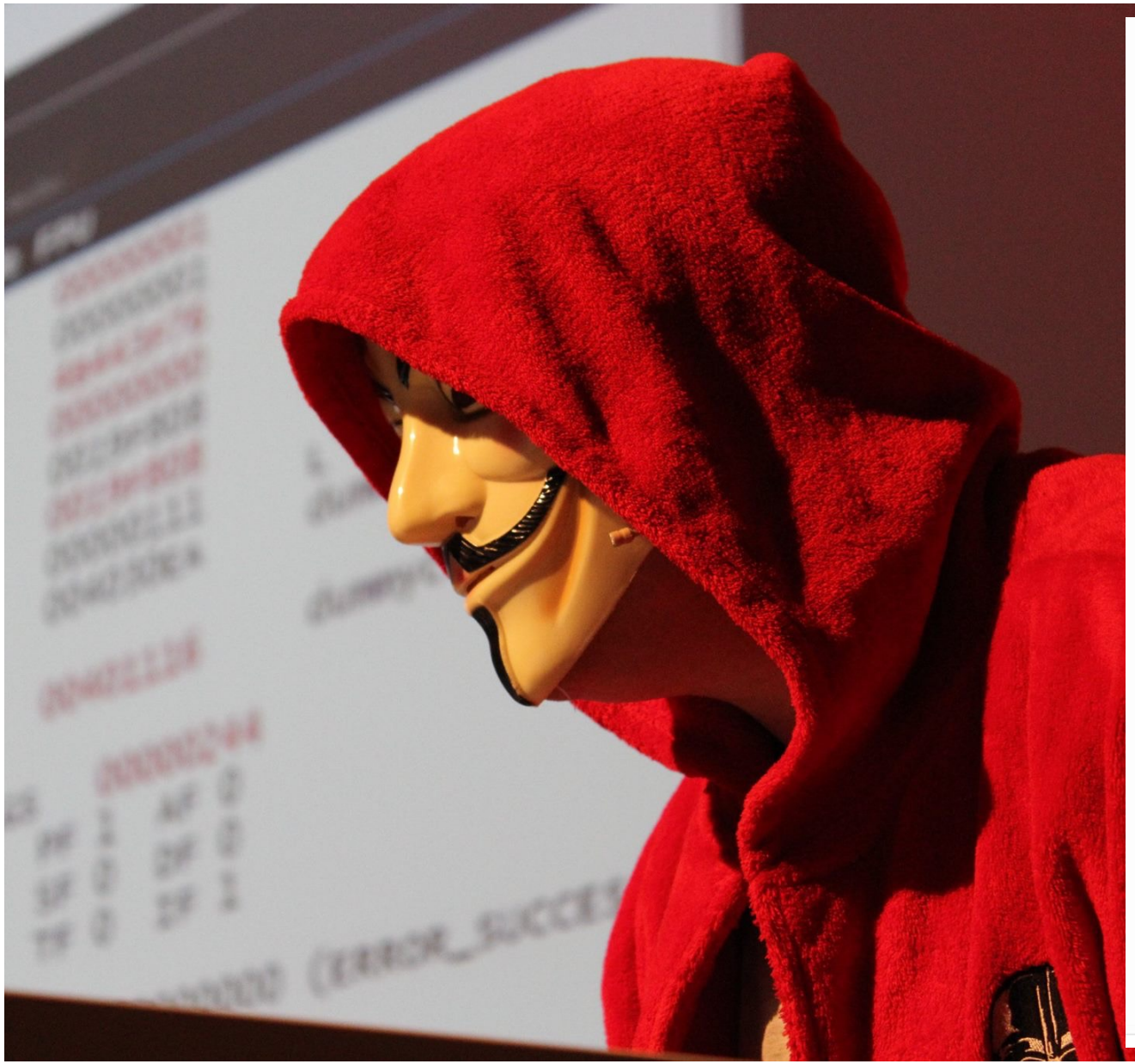
Key Concepts Across Security Domains
- Identity and Access Management (IAM): Managing user identities and permissions.
- Networking Security: Protecting networks from threats.
- Security Operations (SecOps): Incident response, monitoring, and threat hunting.
- Infrastructure Security: Securing servers, cloud resources, and endpoints.
- Data Security: Safeguarding sensitive information.

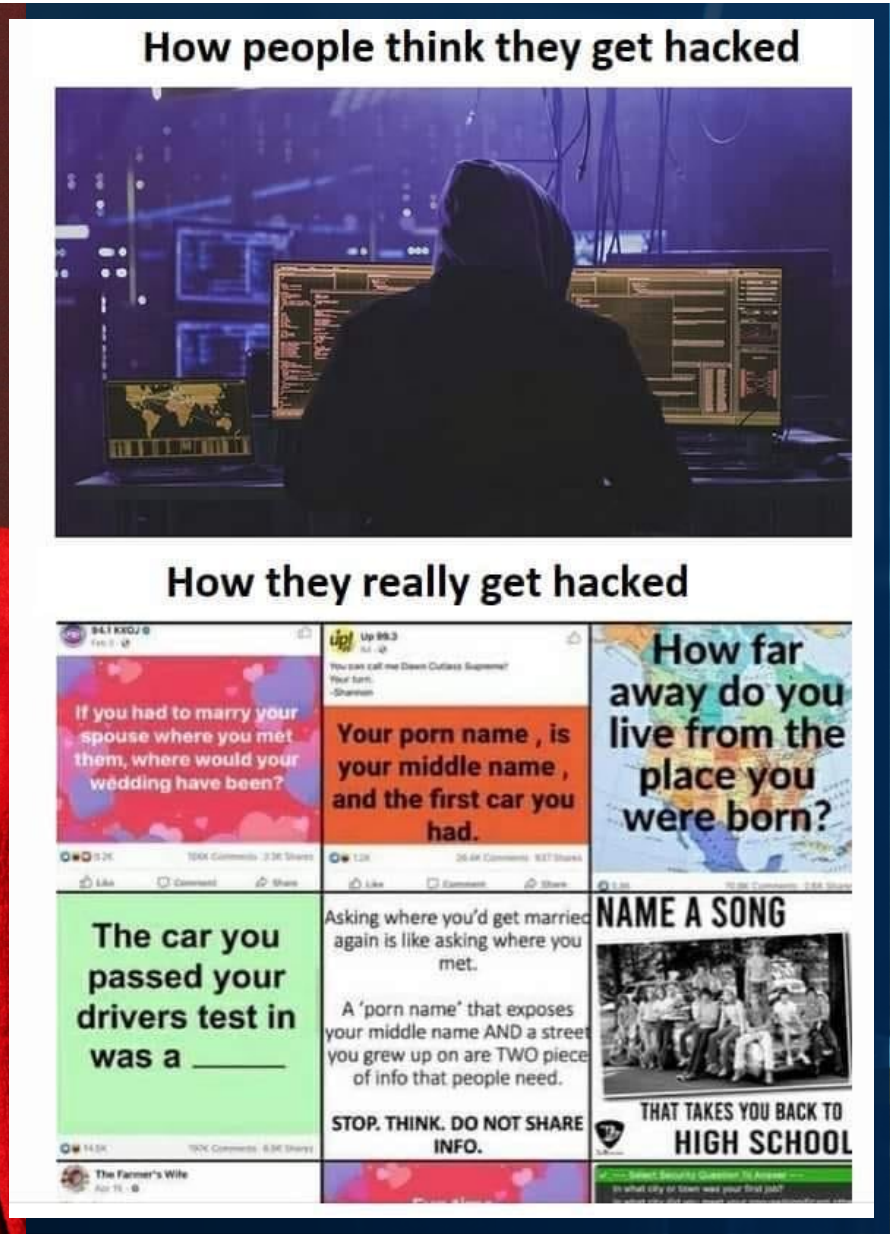
Examples of Security Tools
- Firewalls: Network security appliances.
- Intrusion Detection Systems (IDS): Detecting suspicious activity.
- Endpoint Protection: Securing devices.
- Encryption Tools: Protecting data in transit and at rest.

The Consequences for Developers
- Securing development environments and pipelines.
- Manage the Software Supply chain and automate the update vulnerable dependencies.
- Automatically securing your code and infrastructure as code using code scanning tools.
- Prevent leaked credentials and manage / rotate credentials in a secure way.

Attendees will gain a solid understanding of cybersecurity fundamentals, learn about zero trust principles, and discover practical tools to secure the entire development lifecycle.


Hackers in movies
vs
Image thanks to Gregor Biswanger @BFreakout
© 2021 Michael Kaufmann @mike_kaufmann




How people think they get hacked
How they really get hacked
If you had to marry your spouse where you met them, where would your wedding have been?
Your porn name , is your middle name , and the first car you had.
How far away do you live from the place you were born?
The car you passed your drivers test in was a _____
Asking where you'd get married again is like asking where you met.
A 'porn name' that exposes your middle name AND a street you grew up on are TWO pieces of info that people need.
STOP. THINK. DO NOT SHARE INFO.
NAME A SONG
THAT TAKES YOU BACK TO HIGH SCHOOL

Losses caused by cyber attacks reported to IC3

2021	2023
$ 6.900.000.000	$ 12.500.000.000

Loss caused by reported cyber crime (in billion USD)

Top 5 crime types:

- Phishing
- Non-Payment / Delivery
- Data Breach
- Identity Theft
- Extortion

Trends

- Confidence fraud / Romance scams
- Cryptocurrency
- Ransomware
- Tech support fraud

https://www.ic3.gov/Media/PDF/AnnualReport/2023_IC3Report.pdf

Michael Kaufmann

Founder & Managing Director, Xebia Germany

@mike_kaufmann

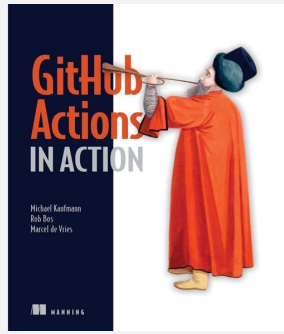
@wulfland

https://writeabout.net

Microsoft REGIONAL DIRECTOR

MVP Microsoft® Most Valuable Professional

- 25 years software developer
- 15 years ALM & DevOps
- 10 years Git and GitHub
- Microsoft RD & MVP



7

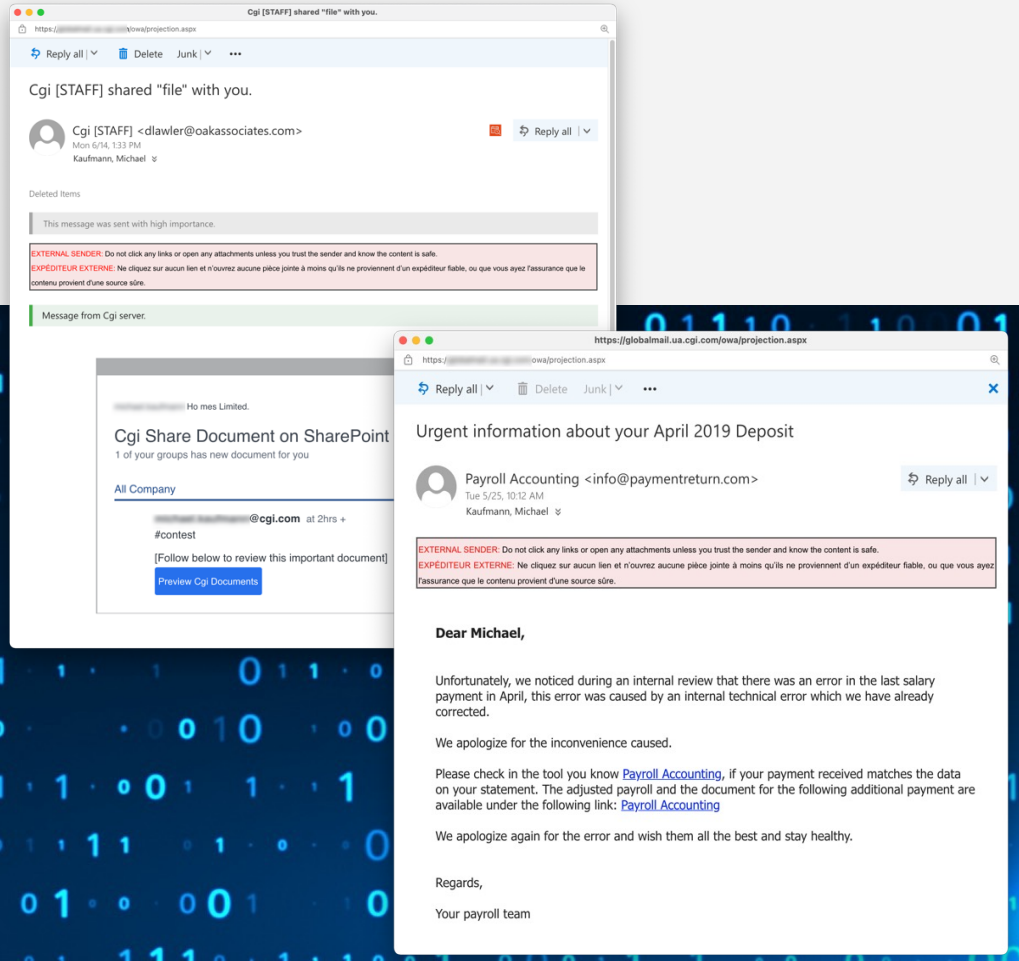
What is Cybersecurity?

Cybersecurity is the protection of computer software, systems, networks, devices, and data from threats that can lead to unauthorized information disclosure, theft or damage, as well as from the disruption or misdirection of the services they provide.

Most Common Types of Cynersecurity Threats

- Phishing
- Malware
- Denial of Service (DoS) and Distributed Denial of Service (DDoS)
- SQL Injection
- Cross-Site Scripting (XSS)
- Social Engineering
- Zero-Day Exploits
- Credential attacks

Phishing or Spear Phishing



Phishing involves sending deceptive emails or messages that appear to be from legitimate sources in order to trick recipients into revealing sensitive information, such as passwords, credit card numbers, or personal details.

Phishing can also lead victims to malicious websites or to download malware.

Malware

Ransomware: Encrypts files and demands a ransom for decryption.

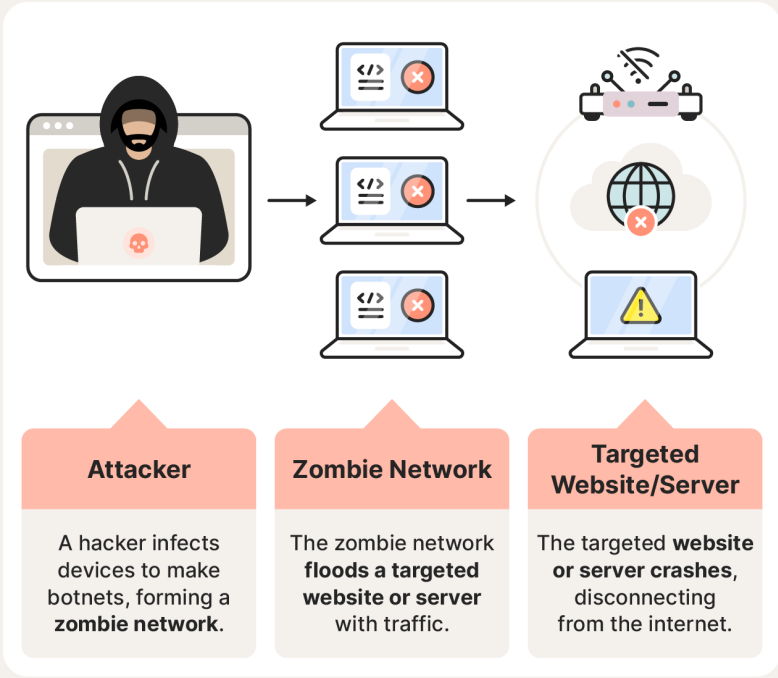
Trojans: Disguised as legitimate software, they give attackers unauthorized access.

Viruses: Self-replicating programs that attach to files and spread.

Worms: Self-replicating programs that spread through networks.

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

DoS and DDoS attacks overload a target system, rendering it unavailable to users. DDoS attacks involve using a network of compromised devices to flood a target with traffic, making it difficult for the system to function properly or they may stop the system working entirely.



SQL Injection

```
txtUserId = getRequestString("UserId");
txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;
```

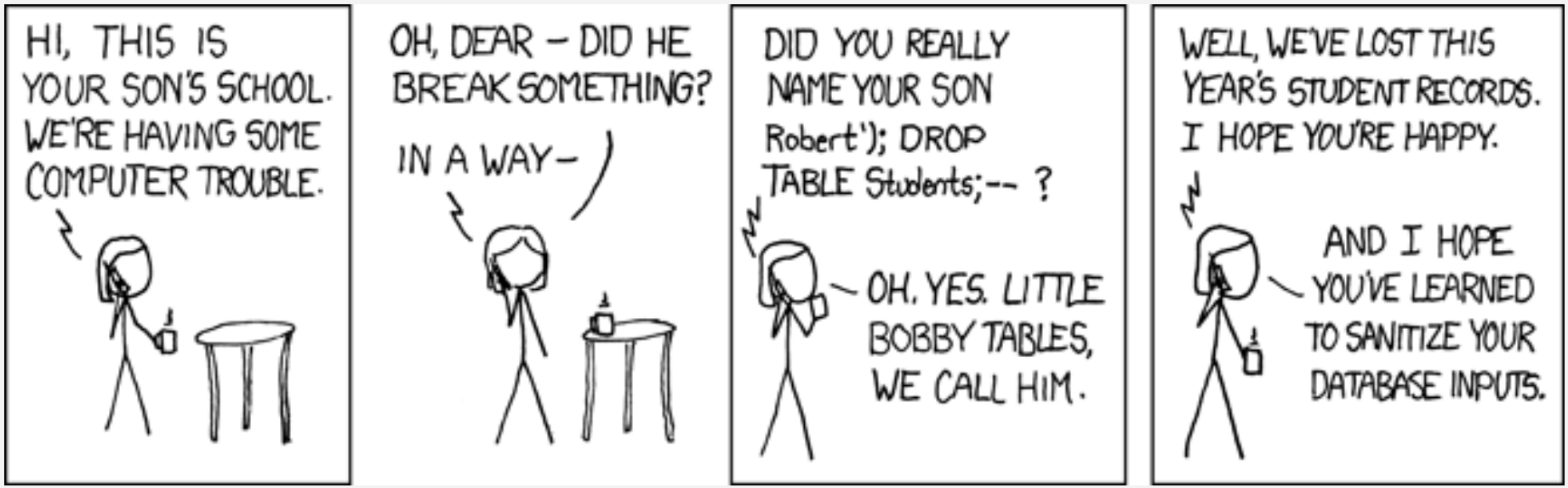
User ID: 105 OR 1=1

```
module.exports = function searchProducts () {
  return (req, res, next) => {
    let criteria = req.query.q === 'undefined' ? '' : req.query.q || ''
    criteria = (criteria.length <= 200) ? criteria : criteria.substring(0, 200)
    models.sequelize.query(`SELECT * FROM Products WHERE ((name LIKE '%${criteria}%' OR description LIKE '%${criteria}%') AND deletedAt IS NULL) ORDER BY name`)
      .then(([products]) => {
        const dataString = JSON.stringify(products)
        if (utils.notSolved(challenges.unionSqlInjectionChallenge)) { // vuln-code-snippet hide-start
          let solved = true
          models.User.findAll().then(data => {
            const users = utils.queryResultToJson(data)
            if (users.data?.length) {
              for (let i = 0; i < users.data.length; i++) {
                solved = solved && utils.containsOrEscaped(dataString, users.data[i].email) && utils.contains(dataString, users.data[i].password)
                if (!solved) {
                  break
                }
              }
              if (solved) {
                utils.solve(challenges.unionSqlInjectionChallenge)
              }
```

SQL Injection

```
txtUserId = getRequestString("UserId");
txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;
```

User ID: 105; DROP TABLE Suppliers



(Source: https://xkcd.com/327/)

Cross-Site Scripting (XSS)

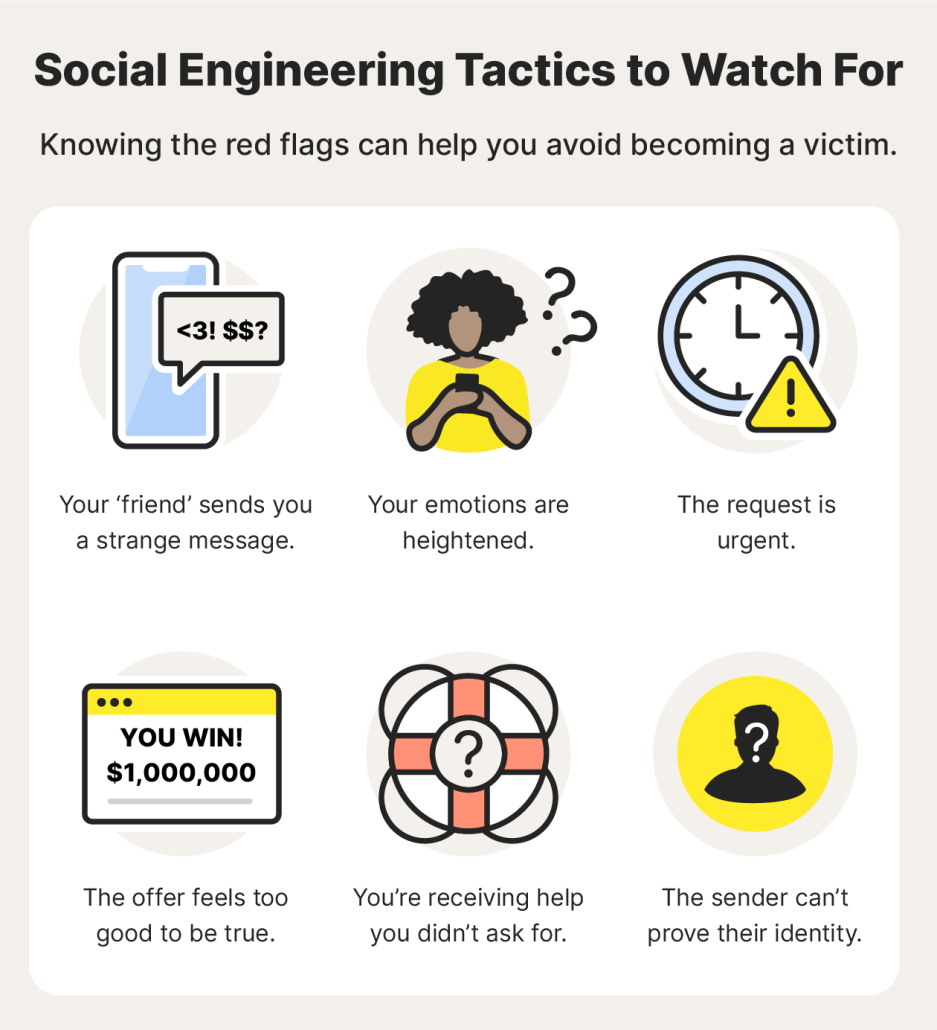
Attackers inject malicious scripts into web applications, which are then executed by unsuspecting users' browsers. This can lead to the theft of user data and/or the spreading of malware.

```
var Affix = function (element, options) {
  this.options = $.extend({}, Affix.DEFAULTS, options)

  this.$target = $(this.options.target)
    .on('scroll.bs.affix.data-api', $.proxy(this.checkPosition, this))
    .on('click.bs.affix.data-api',  $.proxy(this.checkPositionWithEventLoop, this))

  this.$element     = $(element)
  this.affixed      = null
  this.unpin        = null
  this.pinnedOffset = null

  this.checkPosition()
}
```

```
143   // vuln-code-snippet start localXssChallenge xssBonusChallenge
144   filterTable () {
145     let queryParam: string = this.route.snapshot.queryParams.q
146     if (queryParam) {
147       queryParam = queryParam.trim()
148       this.ngZone.runOutsideAngular(() => { // vuln-code-snippet hide-start
149         this.io.socket().emit('verifyLocalXssChallenge', queryParam)
150       }) // vuln-code-snippet hide-end
151       this.dataSource.filter = queryParam.toLowerCase()
152       this.searchValue = this.sanitizer.bypassSecurityTrustHtml(queryParam)
153       this.gridDataSource.subscribe((result: any) => {
154         if (result.length === 0) {
155           this.emptyState = true
156         } else {
157           this.emptyState = false
158         }
159       })
160     } else {
161       this.dataSource.filter = ''
162       this.searchValue = undefined
163       this.emptyState = false
164     }
165   }
```

Social Engineering

Social engineering exploits human psychology to manipulate individuals into divulging confidential information or performing actions that compromise security. (OSINT)



Zero-Day (Oday) Exploits

- These attacks target vulnerabilities in software or hardware that are not yet known to the vendor or public. Attackers take advantage of these vulnerabilities before patches are developed.

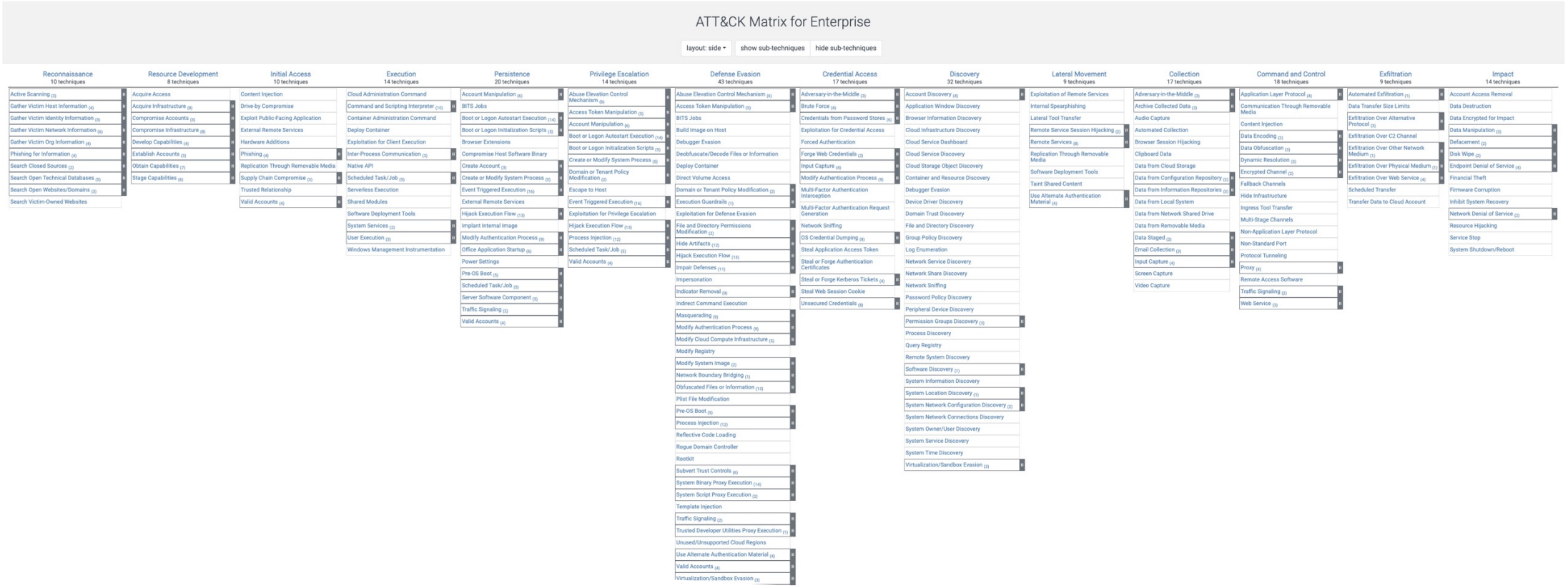
Credential Attacks

Brute force attacks: attackers repeatedly guess passwords

Credential stuffing attacks: stolen credentials used on other systems

Responder: https://github.com/lgandx/Responder

Pineapple: wireless auditing platform



The MITRE ATT&CK framework

ATT&CK: Adversarial Tactics, Techniques, and Common Knowledge: https://attack.mitre.org/

- Understand Adversarial Behavior
- Plan and Implement Defense Strategies
- Incident Response and Threat Hunting

Keep up to date:

Open Web Application Security Project (OWASP) top 10 vulnerabilities: https://owasp.org/Top10/

Common Vulnerabilities and Exposures (CVEs): https://cve.mitre.org/

Microsoft Security Response Center blogs: https://msrc.microsoft.com/blog/

National Institute of Standards and Technology (NIST): https://www.dhs.gov/topics/cybersecurity

Cybersecurity and Infrastructure Security Agency (CISA): https://www.cisa.gov/resources-tools/resources/free-cybersecurity-services-and-tools

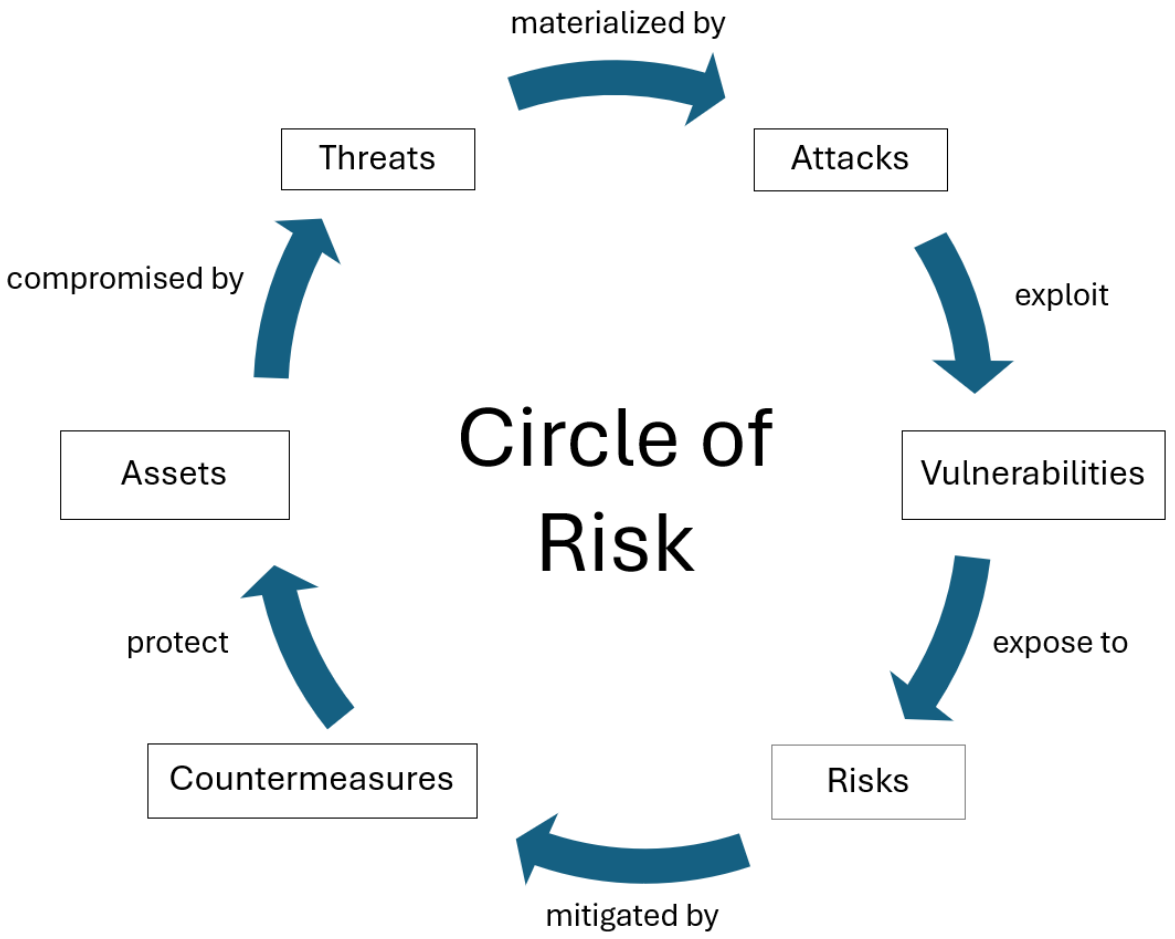
National Cybersecurity Center of Excellence (NCCoE): https://www.dhs.gov/topics/cybersecurity

United States Computer Emergency Readiness Team (US-CERT): https://www.cisa.gov/resources-tools/resources/free-cybersecurity-services-and-tools

Risk Management

Risk Assessment:

1. Identify Assets and Threats
2. Assess Vulnerabilities
3. Likelihood Assessment
4. Impact Assessment
5. Risk Calculation
6. Prioritization and Decision-Making
7. Risk Treatment
8. Continuous Monitoring and Review



Security Practices

Security Policy

- High-level document
- Outlines goals, principles, and guidelines

Security Standard

- More detailed
- Guidelines and requirements for implementing security controls and measures

Security Baseline

- Minimum security configuration
- Specific to a system, application, or environment

Security Guideline

- Recommendations and guidance, when standards do not apply
- Gray areas

Security Procedure

- Step-by-step guide
- Practical and actionable

Assume breach

„Fundamentally, if somebody wants to get in, they're getting in. Alright, good. Accept that.“

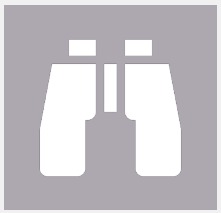
- Michael V. Hayden

former General of the US Air Force and former Director of NSA and CIA

From Prevent Breach to Assume Breach

Prevent Breach

- Security layers (Firewall, VPN, DMZ)
- Risk analysis
- Threat modeling
- Architecture reviews
- Code security reviews
- Security testing

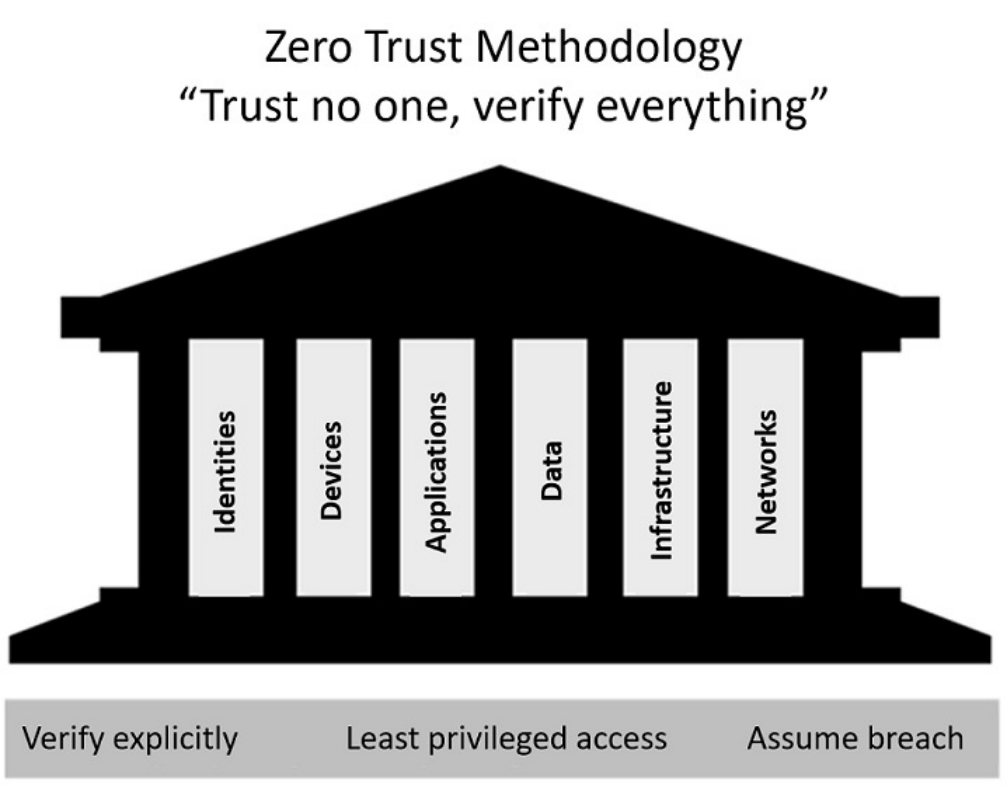


Assume Breach

- Red-Team | Blue-Team exercises (war games)
- Central security monitoring (SIEM)
- Penetration testing in production
- Detection of anomalies

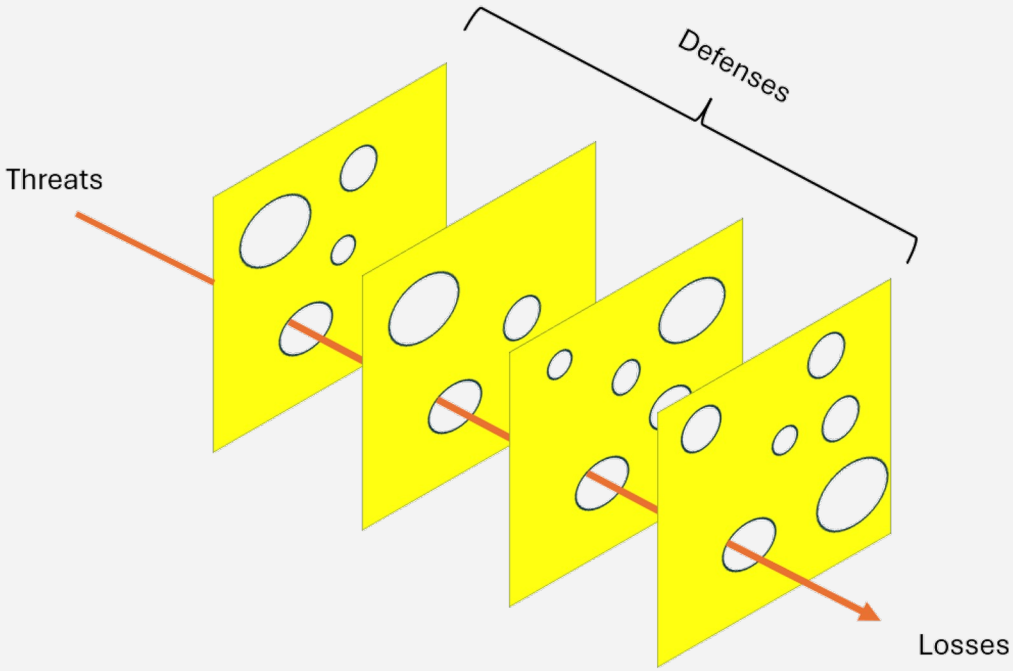
Zero Trust Principles

- **Verify Identity:**
  Authentication and authorization are rigorously applied to all users and devices, regardless of their location.
- **Least Privilege:**
  Users and devices are granted the minimum level of access necessary to perform their tasks.
- **Micro-Segmentation:**
  Network resources are divided into smaller segments to limit lateral movement within the network in case of a breach.
- **Continuous Monitoring:**
  Ongoing monitoring and analysis of user and device behavior. Detection of anomalies and potential threats.
- **Data Encryption:**
  Data is encrypted both in transit and at rest to prevent unauthorized access.
- **Strict Access Control:**
  Access controls are enforced based on context, such as user roles, device health, and network location.



Differences from Traditional Security Architectures

- **Perimeter vs. Identity-Centric:**
  Traditional models focus on securing the perimeter assuming that internal users and devices can be trusted. Zero Trust assumes that threats can originate from both inside and outside the network.
- **Implicit vs. Explicit Trust:**
  Traditional models implicitly trust devices and users within the network until proven otherwise. Zero Trust explicitly verifies identities and continuously monitors for anomalies.
- **Flat vs. Segmented Network:**
  Traditional architectures often involve flat networks where internal users have broad access. Zero Trust advocates segmenting the network into smaller, isolated zones to contain potential breaches.
- **Reactive vs. Proactive:**
  Traditional security often relies on reactive measures such as perimeter firewalls and intrusion detection. Zero Trust takes a proactive approach by assuming breaches are likely and minimizing their impact.



Identity and Access Management (IAM)

A set of processes, technologies, and policies that are implemented to ensure that the right individuals have the appropriate access to resources within an organization's digital environment.

Key concepts:

- **Least privilege**
- **Segregation of duties**

Identity and Access Management (IAM)

Authentication and Multi-Factor Authentication (MFA or 2FA):

- **Something you know**: This includes passwords, PINs, or another secret knowledge that only the authorized user should possess.
- **Something you have**: This involves physical tokens or devices like smart cards, security tokens, or mobile phones used to confirm the user's identity.
- **Something you are**: This refers to biometric factors like fingerprints, facial recognition, or retinal scans that are unique to an individual.

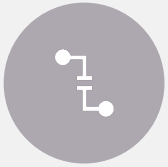
Identity and Access Management (IAM)

Authorization:

- Authorization is the process of granting or denying specific permissions and privileges to authenticated users or. It determines what actions or operations a user is allowed to perform within a system or on specific resources.
- Authorization can be thought of as answering the question, "What can an authenticated user do?" It involves defining and enforcing access control policies to protect sensitive data and resources from unauthorized access or modification.

Other IAM Capabilities

Single Sign-On (SSO): Accessing multiple applications and systems with a single set of credentials



Role-Based Access Control (RBAC): Assigning permissions based on predefined roles

Adaptive Authentication: Assessing risk factors such as location, device, time of access, and user behavior to dynamically adjust the level of authentication required

Biometric Authentication: Usinging unique biological characteristics such as fingerprints, facial features, voice patterns

Privileged Access Management (PAM): Securing access to critical systems and data by enforcing strict controls on privileged accounts (just-in-time access)

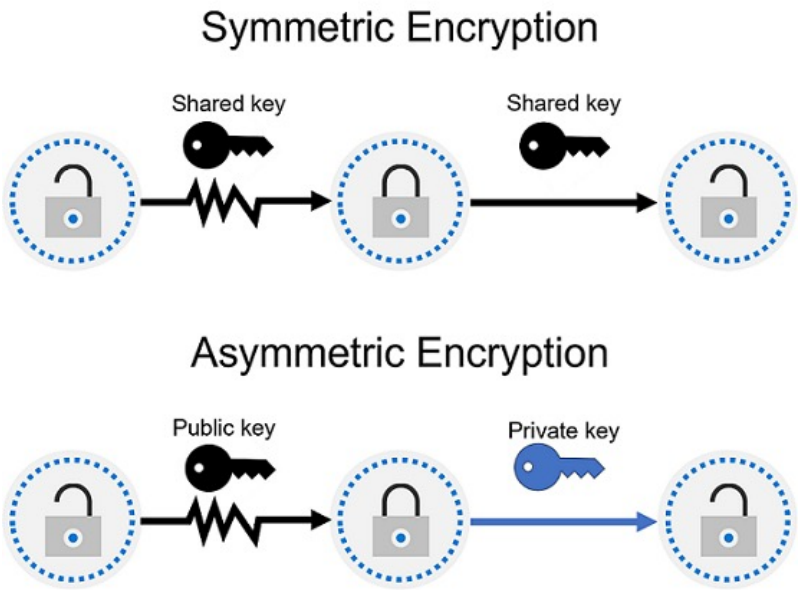
Behavioral Analytics: Monitoring user behavior and establishing baseline patterns. Deviations can trigger alerts.

Network Security

- Traditional firewalls
- Web application firewalls
- Cloud security groups
- CDN
- Load balancers
- Bastion hosts
- VPNs
- DDoS protection

Layer			Function
Host layers	7	Application	High-level protocols such as for resource sharing or remote file access, e.g. HTTP.
	6	Presentation	Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5	Session	Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4	Transport	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3	Network	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2	Data link	Transmission of data frames between two nodes connected by a physical layer
	1	Physical	Transmission and reception of raw bit streams over a physical medium

Encryption and hashing



Security Operations

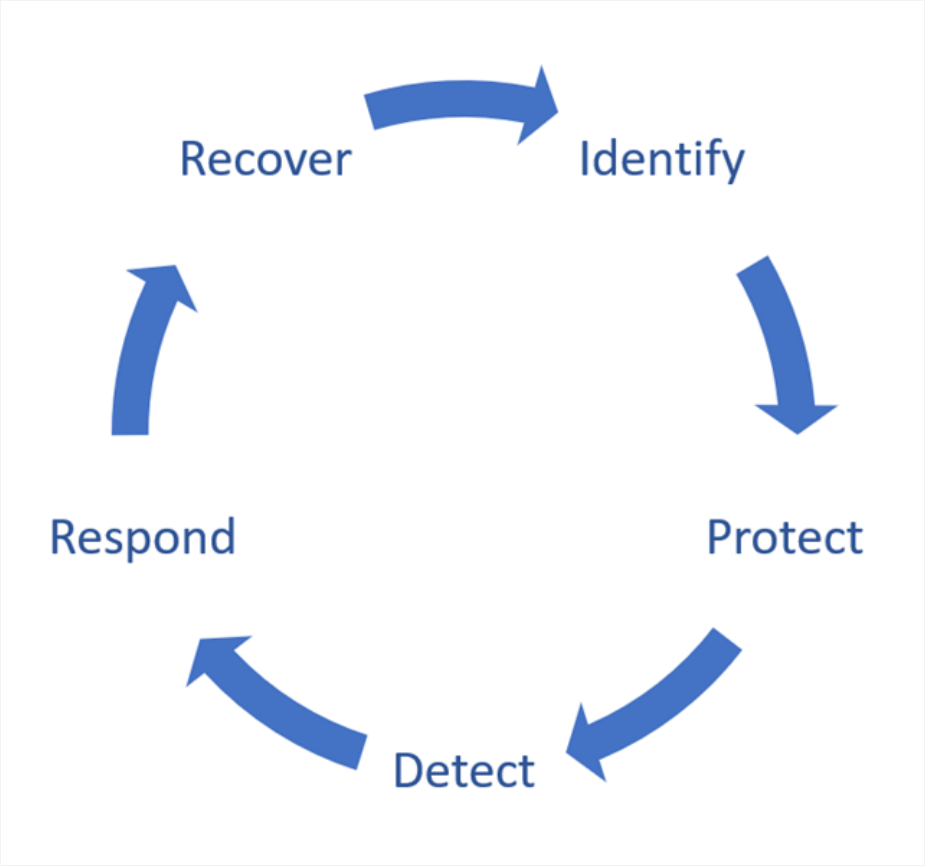
Security Operations Center (SOC):
A centralized team responsible for 24/7 monitoring, analysis, and response to security events.

Incident Response Team:
A specialized team focused on responding to security incidents and breaches. They conduct investigations, coordinate response efforts, and facilitate recovery.

Threat Hunting Team:
A team that proactively searches for signs of advanced threats and hidden vulnerabilities that may not be detected by traditional security tools.

Red Team/Blue Team:
The red team simulates attacks to identify vulnerabilities, while the blue team defends against those attacks. Both teams work together to improve security measures.

Managed Security Services Provider (MSSP):
Some organizations outsource their security operations to third-party providers specializing in security monitoring and incident response.



Security Information and Event Management (SIEM)

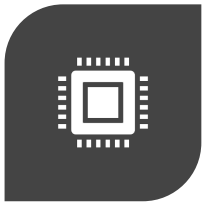
- **Log Collection**: Collect logs and security event data.
- **Data Normalization**: Normalize log data into a common format.
- **Event Correlation**: Correlate events to identify patterns and anomalies.
- **Alerting and Notification**: Generate alerts and notifications in real time.
- **Incident Detection**: Detection of security incidents, including unauthorized access, data breaches, malware infections, and insider threats.
- **User and Entity Behavior Analytics (UEBA)**: Identify abnormal user and entity behaviors that may indicate compromised accounts or insider threats.
- **Threat Intelligence Integration**: Integrate with threat intelligence feeds to enhance threat detection by comparing known indicators of compromise (IOCs) with network activity.
- **Automation and Orchestration**: Automate responses to common security incidents, reducing response times and manual effort.
- **Dashboard and Visualization**: Allow monitoring security data and creating custom reports.
- **Integration with Other Security Tools: I**ntegrate with other security tools and technologies, such as endpoint detection and response (EDR) solutions.

Application Security (AppSec)

Application security refers to the practice of protecting software applications from security threats, vulnerabilities, and attacks. It encompasses the processes, techniques, and tools used to identify, mitigate, and prevent security risks throughout the development, deployment, and maintenance lifecycle of an application.

SECURE BY DESIGN: SECURITY SHOULD BE INTEGRATED INTO THE APPLICATION'S DESIGN AND ARCHITECTURE.

INPUT VALIDATION: ALL USER INPUTS SHOULD BE VALIDATED.



OUTPUT ENCODING: DATA SENT TO THE CLIENT SHOULD BE PROPERLY ENCODED TO PREVENT VULNERABILITIES LIKE CROSS-SITE SCRIPTING (XSS).

AUTHENTICATION AND AUTHORIZATION: AUTHENTICATE USERS AND AUTHORIZE THEIR ACCESS TO RESOURCES BASED ON THEIR ROLES AND PERMISSIONS.

DATA PROTECTION: SENSITIVE DATA SHOULD BE ENCRYPTED WHEN STORED, TRANSMITTED, AND PROCESSED TO PREVENT UNAUTHORIZED ACCESS.

Application Security (AppSec)

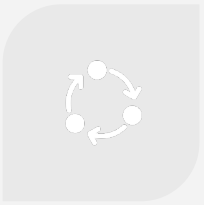
Application security refers to the practice of protecting software applications from security threats, vulnerabilities, and attacks. It encompasses the processes, techniques, and tools used to identify, mitigate, and prevent security risks throughout the development, deployment, and maintenance lifecycle of an application.

SESSION MANAGEMENT: SECURE SESSION MANAGEMENT ENSURES USER SESSIONS ARE PROTECTED FROM HIJACKING AND UNAUTHORIZED ACCESS.

SECURE DEPENDENCIES: KEEP ALL SOFTWARE DEPENDENCIES UP TO DATE WITH SECURITY PATCHES TO PREVENT VULNERABILITIES.

ERROR HANDLING AND LOGGING: IMPLEMENT SECURE ERROR HANDLING TO AVOID REVEALING SENSITIVE INFORMATION AND ENSURE SECURE LOGGING PRACTICES.

SECURITY TESTING: REGULARLY TEST APPLICATIONS FOR VULNERABILITIES USING METHODS LIKE PENETRATION TESTING, CODE REVIEWS, AND AUTOMATED SCANNING TOOLS.

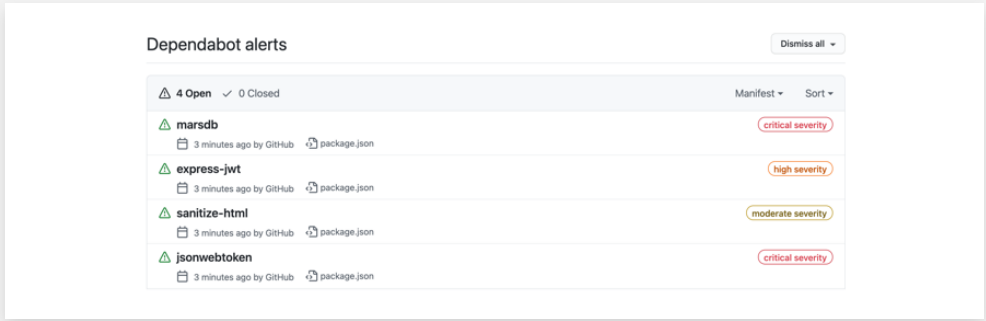


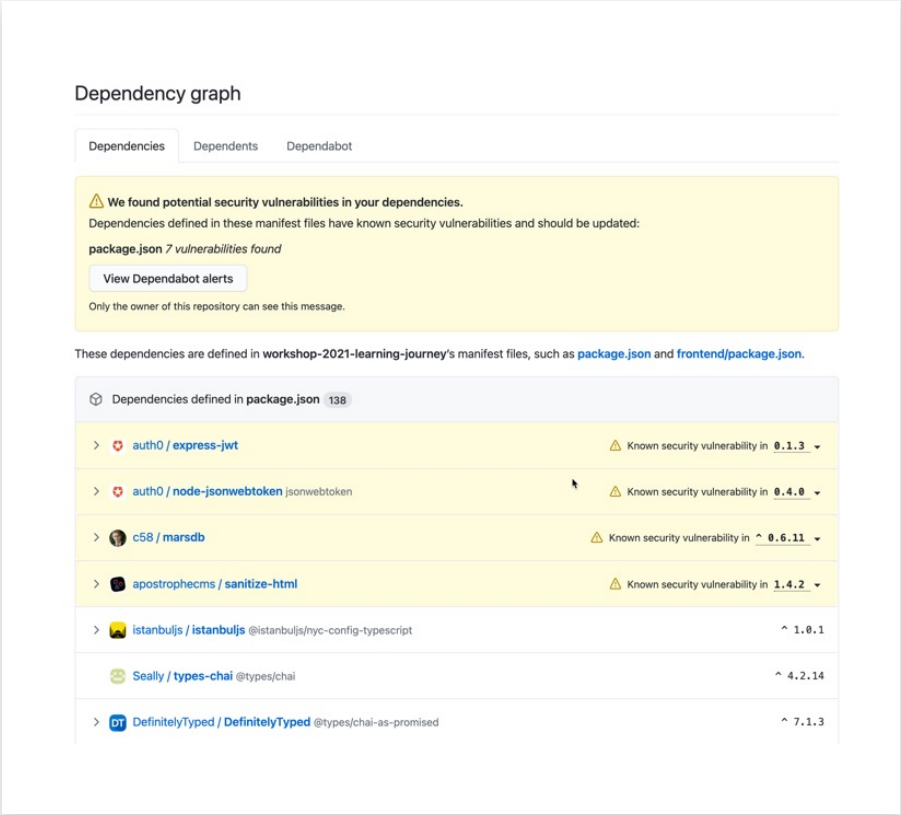
SECURE SOFTWARE DEVELOPMENT LIFECYCLE (SDLC): INTEGRATE SECURITY PRACTICES INTO EVERY PHASE OF THE SOFTWARE DEVELOPMENT LIFECYCLE.

AppSec Key Capabilities

Software Composition Analysis (SCA) a.k.a. Dependency Scanning

- GitHub (Dependency-Graph/Dependabot)
- anchore (https://anchore.com/)
- Dependency-Track (https://dependencytrack.org/)



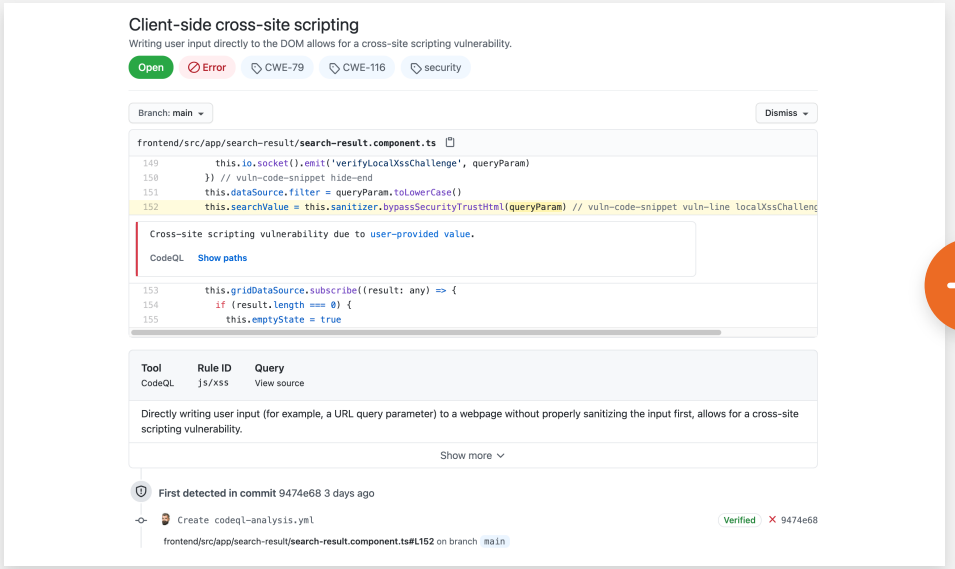


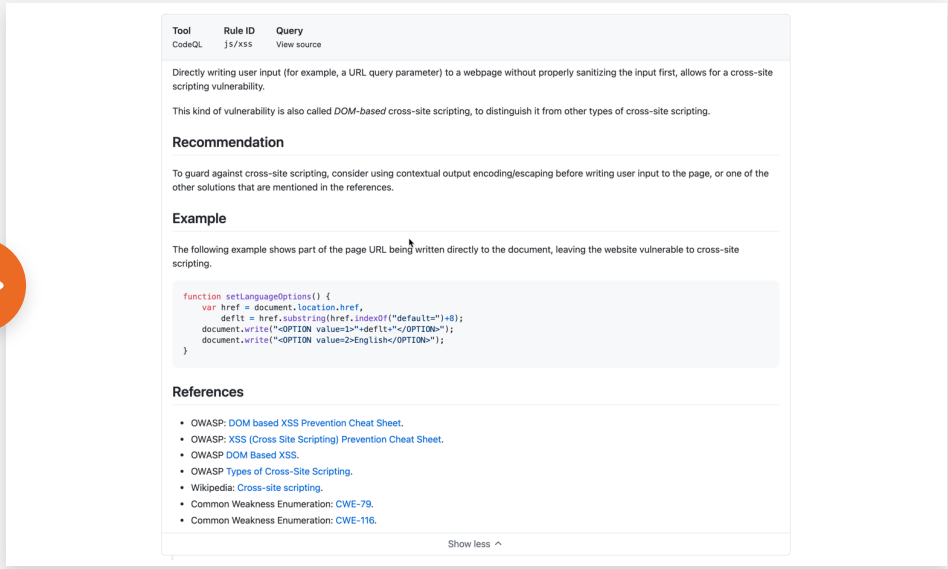
Static Application Security Testing (SAST)

Analyze source code, bytecode, or binary code to identify security vulnerabilities in the application's codebase.

Whitebox-Testing

- GitHub Code Analysis
- SonarQube, Fortify, Checkmarx, and Veracode
- Semgrep (https://semgrep.dev/)
- Mobile-Security-Framework (MobSF) (https://github.com/MobSF/Mobile-Security-Framework-MobSF)



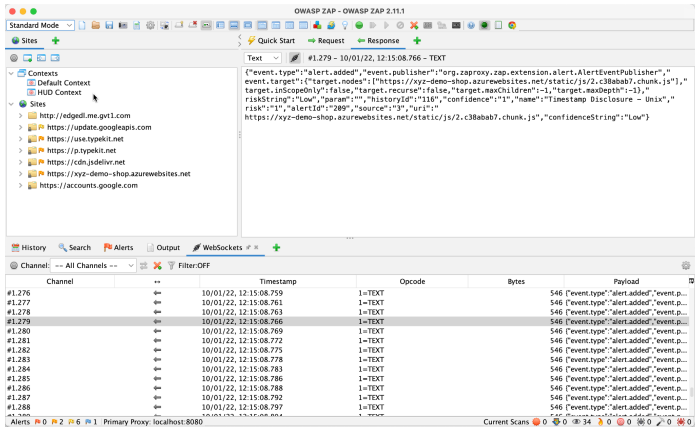


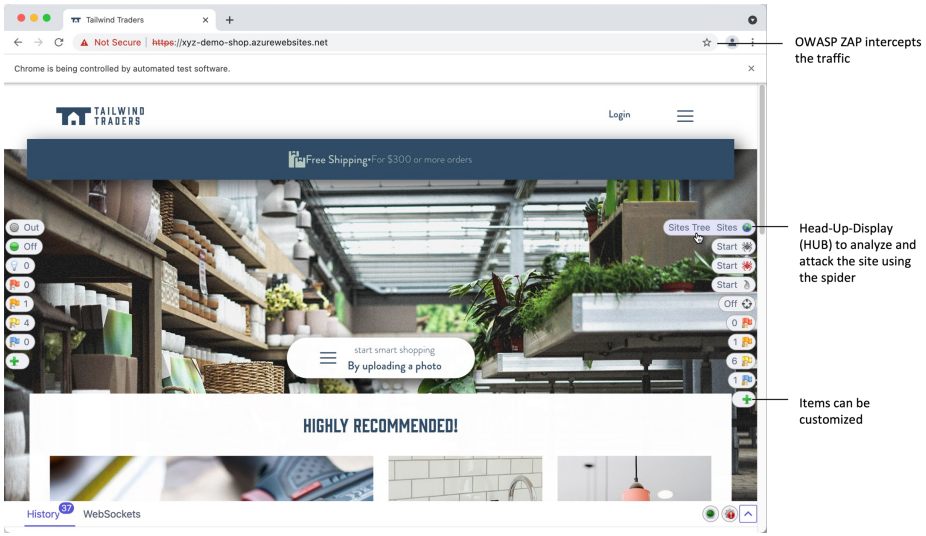
Dynamic Application Security Testing (DAST)

Scans a running application to identify vulnerabilities by sending input requests and analyzing responses.

Blackbox-Testing

- OWASP ZAP (Zed Attack Proxy, https://owasp.org/www-project-zap)
- Burp Suite von PortSwigger (https://portswigger.net/burp)



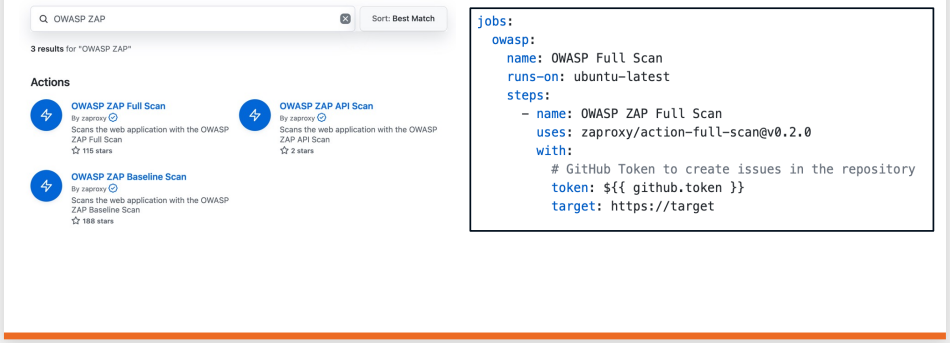


Dynamic Application Security Testing (DAST)

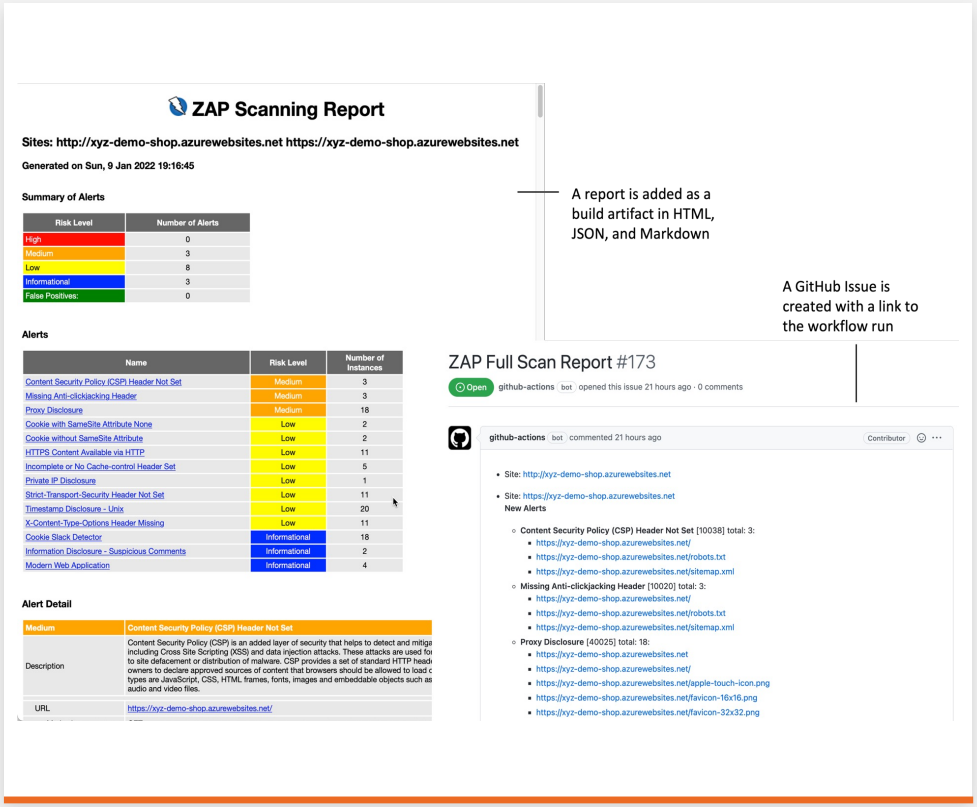
Blackbox-Testing

› OWASP ZAP (Zed Attack Proxy, https://owasp.org/www-project-zap)

› Burp Suite von PortSwigger (https://portswigger.net/burp)




```yaml
jobs:
  owasp:
    name: OWASP Full Scan
    runs-on: ubuntu-latest
    steps:
      - name: OWASP ZAP Full Scan
        uses: zaproxy/action-full-scan@v0.2.0
        with:
          # GitHub Token to create issues in the repository
          token: ${{ github.token }}
          target: https://target
```

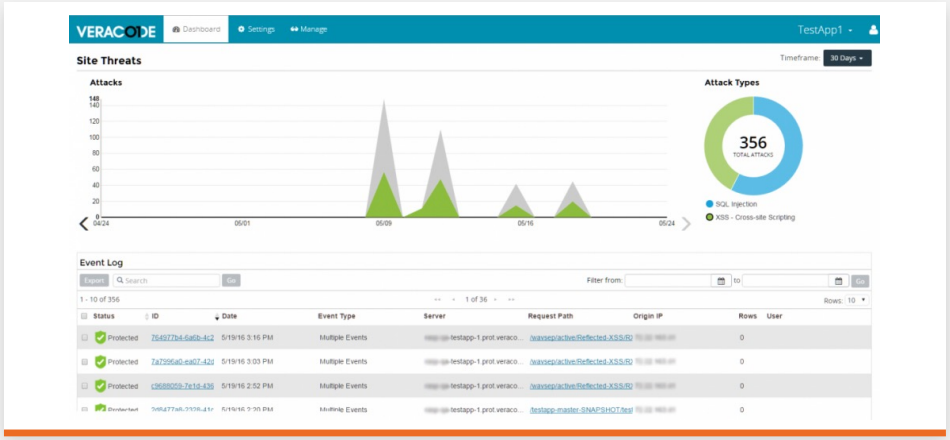


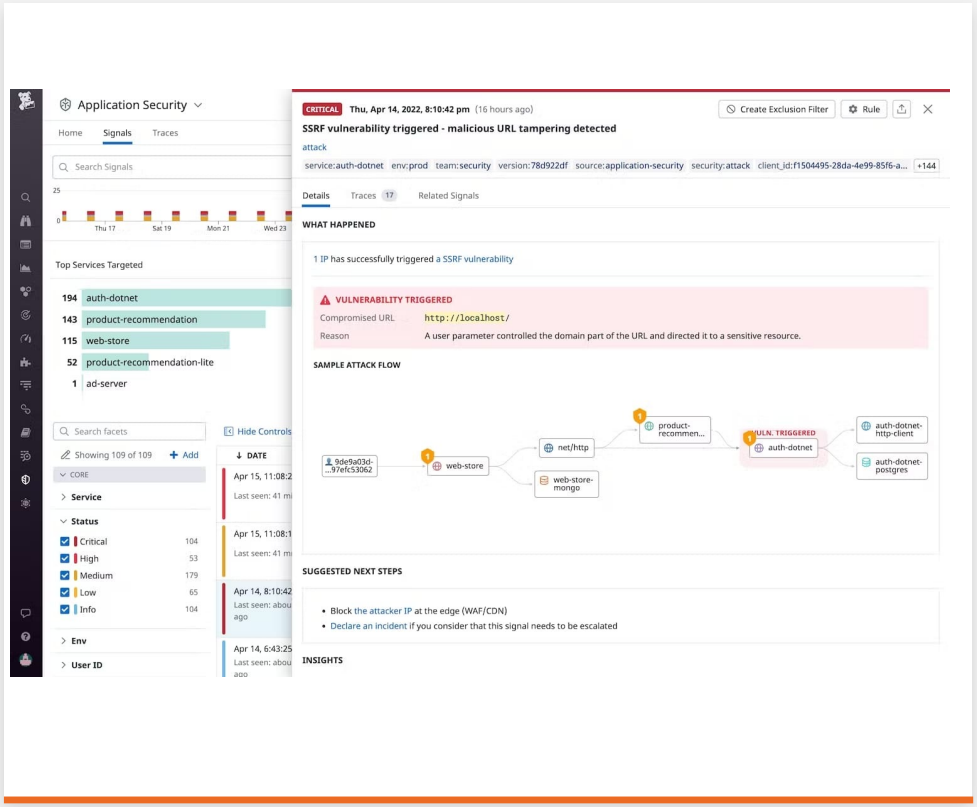
Runtime Application Self-Protection (RASP)

Monitors and protects applications in real-time, detecting and responding to security threats as they occur.

Blackbox-Testing

- Veracode Runtime Protection
- F5 Advanced WAF with RASP
- Datadog Application Security Management





Container Scanning

Container Vulnerability Analysis (CVA) /
Container Security Analysis (CSA)

Open source:

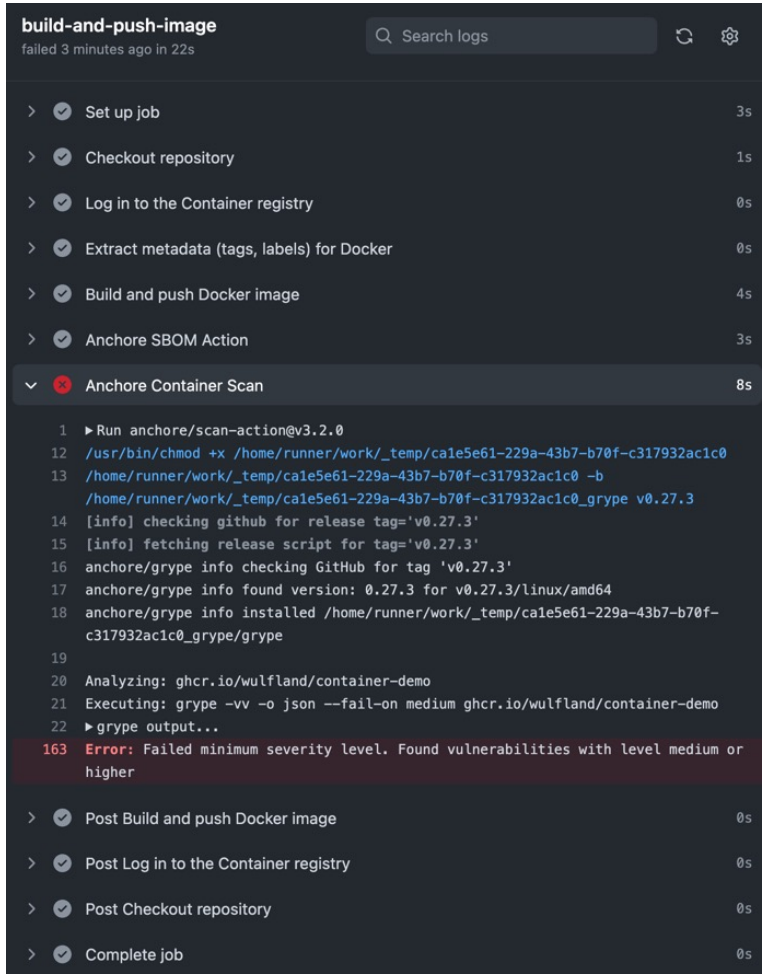
- Anchore gryp
  https://github.com/anchore/grype/
- Clair
  https://quay.github.io/clair/

Commercial:

- WhiteSource
  https://www.whitesourcesoftware.com/solution-for-containers/
- Aqua
  https://www.aquasec.com/products/container-security/

```yaml
- name: Anchore Container Scan
  uses: anchore/scan-action@v3.2.0
  with:
    image: ${{ env.REGISTRY }}/${{ env.IMAGE_NAME }}
    debug: true
```

https://github.com/wulfland/container-demo/actions/runs/2179243137



Infrastructure Scanning

Infrastructure policies

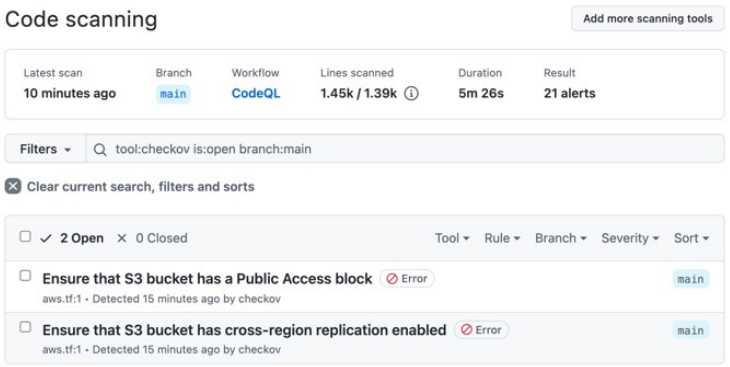
Open source:

- Checkov
  https://www.aquasec.com/products/container-security/
- OpenVAS

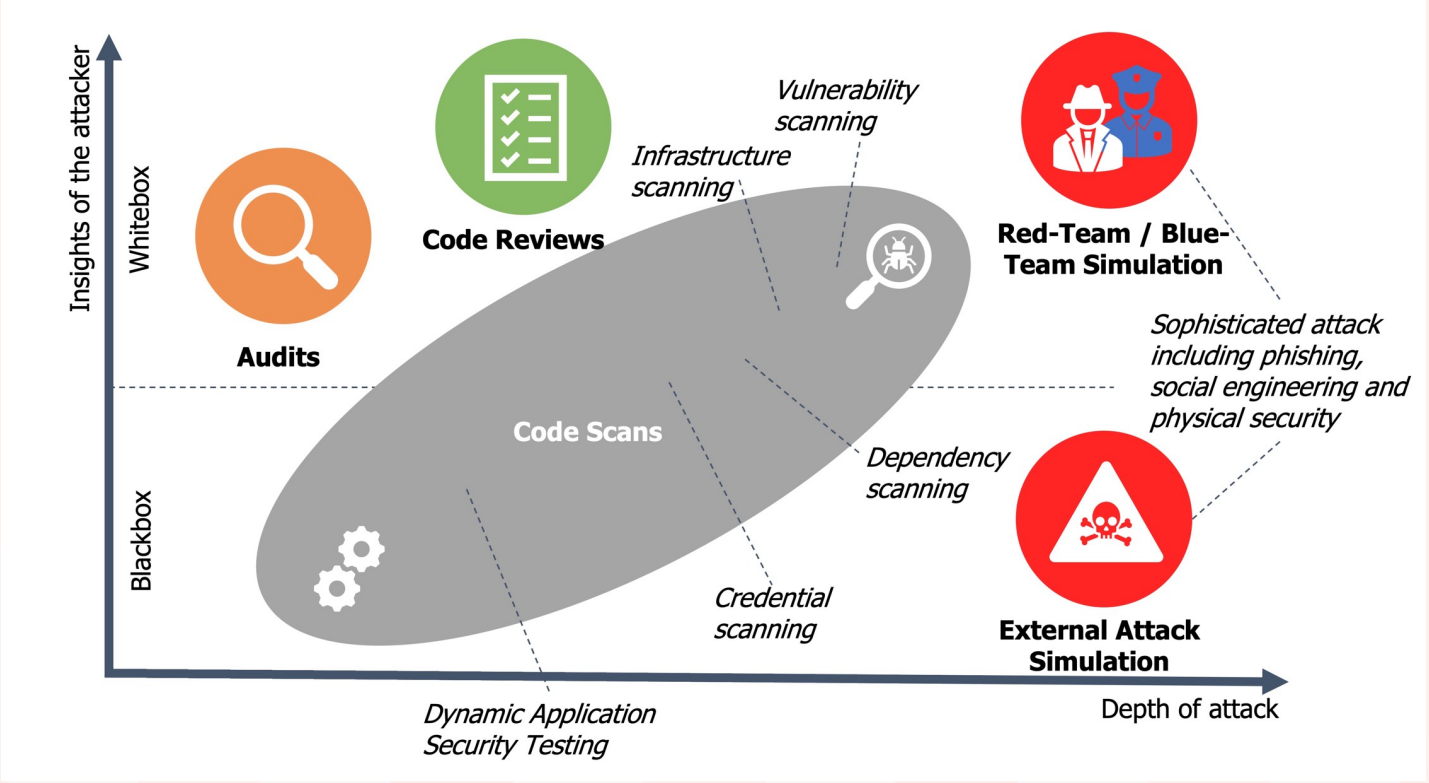
Commercial:

- Defender for Cloud
  https://azure.microsoft.com/en-us/services/defender-for-cloud
- Azure Policy
  https://docs.microsoft.com/de-de/azure/governance/policy/

```yaml
- name: Checkov GitHub Action
  uses: bridgecrewio/checkov-action@master
  with:
    directory: ch15_sec/
    output_format: sarif

- name: Upload SARIF file
  uses: github/codeql-action/upload-sarif@v1
  with:
    sarif_file: results.sarif
  if: always()
```



Security Testing



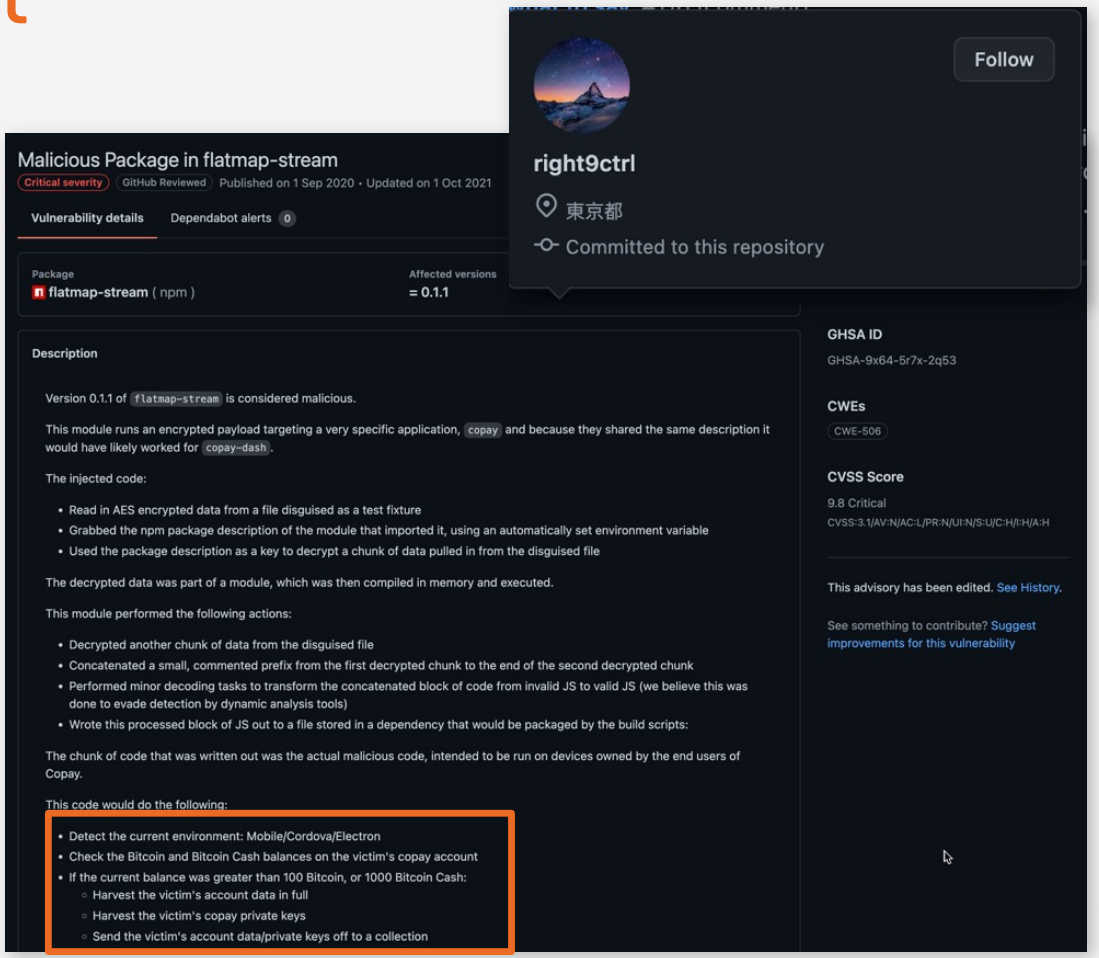
The event-stream incident

Social engineering attack

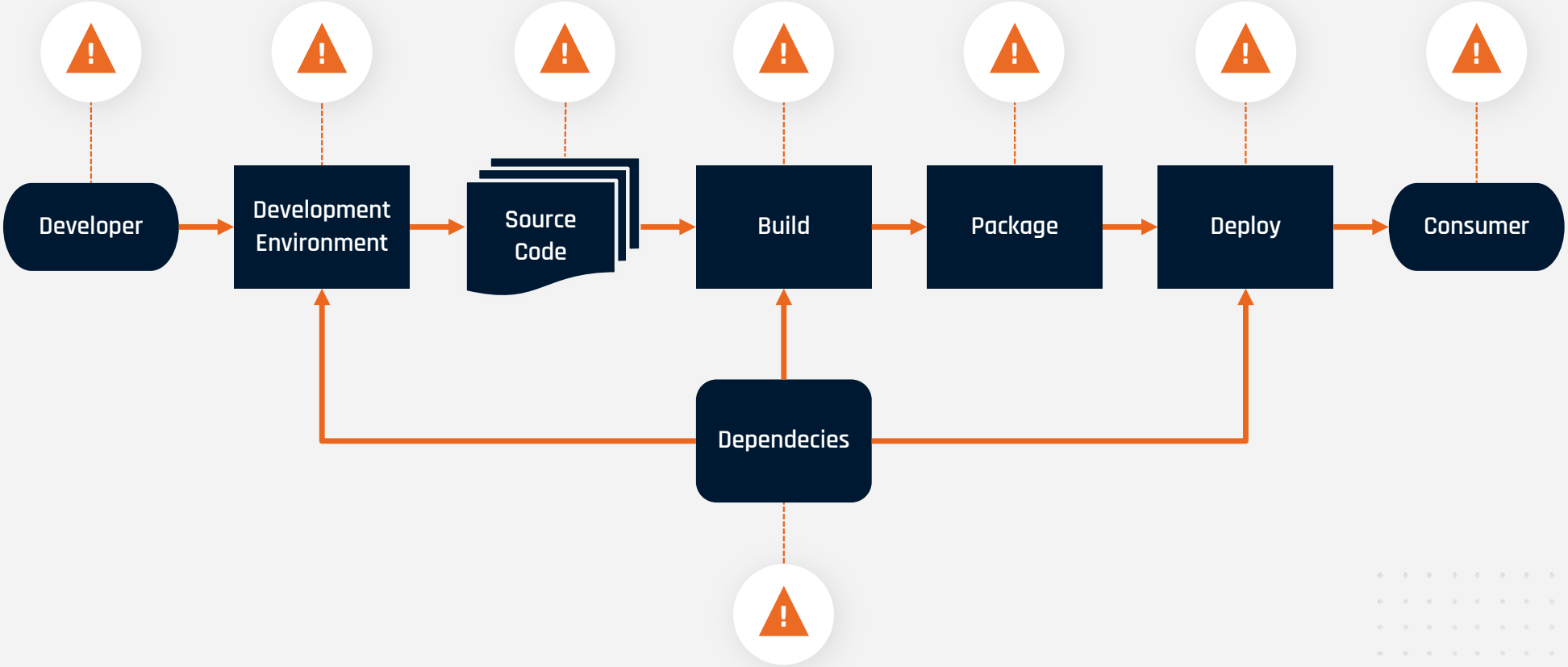
Supply chain attack:
event-stream@3.3.6 -> flatmap-stream@0.1.1

Code execution in build process targeting copay

Harvest the user's bitcoin and private keys

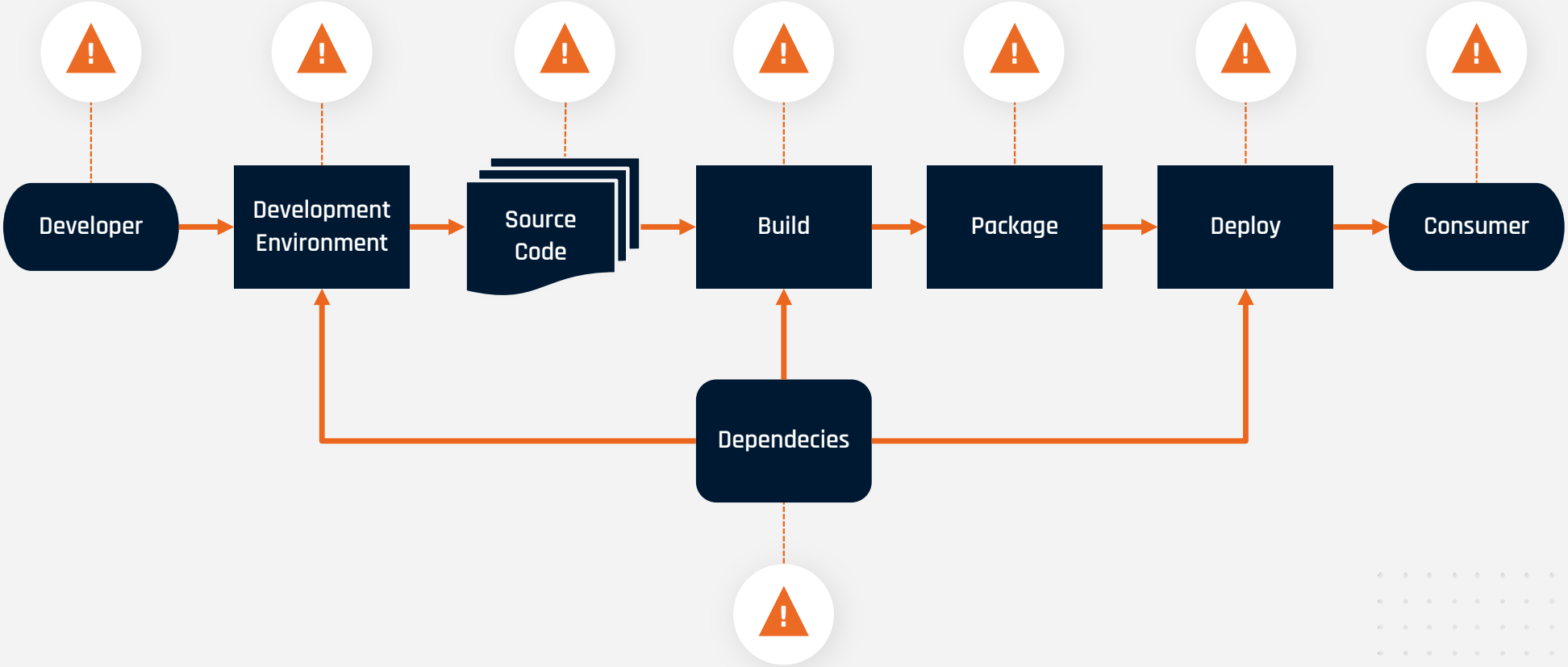


Attack vectors

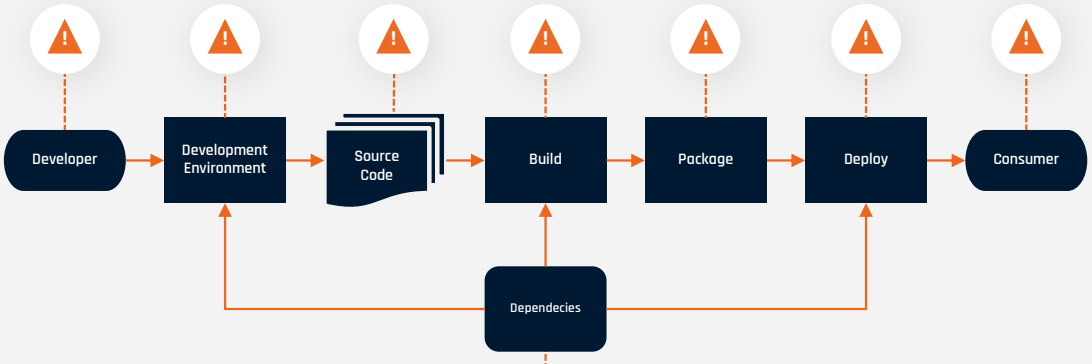



gatsby|
show
Remaining packages: 1
view source code
share to twitter
become a patron
NoteBook

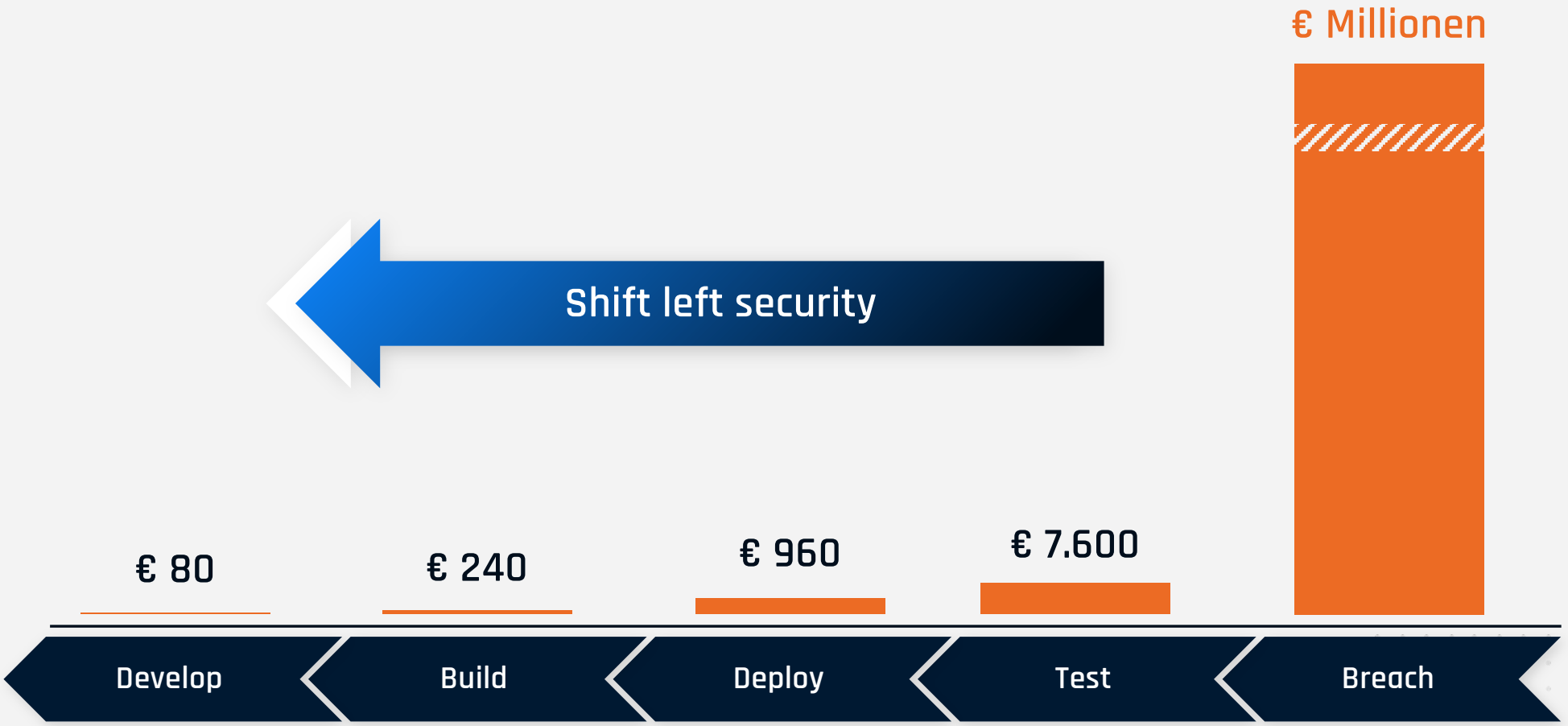
Attack vectors



Attack vectors



Costs for fixing a security vulnerability



Source: Ponemon Institute Cost of a Data Breach 2020

The Consequences for Developers

- Secure your development environments and pipelines!
- Manage your Software Supply chain and automate the update of (vulnerable) dependencies!
- Automatically securing your code and infrastructure as code using code scanning tools.
- Prevent leaked credentials and manage / rotate credentials in a secure way.
- Use best practices and common standards!
- Learn the basics – stay up to date with security!

Check out: https://github.com/microsoft/Security-101


Hackers in movies
vs
Image thanks to Gregor Biswanger @BFreakout
© 2021 Michael Kaufmann @mike_kaufmann

Thank you

Blog : https://writeabout.net

Twitter : @mike_kaufmann

GitHub : @wulfland

LinkedIn : https://www.linkedin.com/in/mikaufmann/