SurfaceMapper

Example Organisation

External Attack Surface Review

March 2026

External Attack Surface Review

Client: Example Organisation
Target: example.surfacemapper.io
Generated: 2026-03-20 03:00:00 UTC

Table of contents

Executive summary

SurfaceMapper identified 14 findings across the external attack surface of example.surfacemapper.io, including 3 Critical and 5 High severity issues requiring immediate attention.

Active credentials were found exposed in two publicly accessible source code repositories — an AWS access key confirmed valid via the AWS API, and a database password in a committed .env file. A live Stripe secret key was also found embedded in a JavaScript file served to all website visitors. Both represent immediately exploitable exposure requiring urgent rotation.

Default credentials were accepted on three internet-facing services — a Netdata monitoring dashboard, an Apache Tomcat Manager, and an FTP server with anonymous access enabled. These services require no prior knowledge to exploit; default credential lists are built into automated attack tooling and are attempted within minutes of exposure.

The most significant network risk is a publicly accessible MySQL database server (port 3306) on two hosts with no network-level access control. Database services of this type are actively scanned and targeted by automated exploit tooling and ransomware operators. Immediate firewall restriction is required.

Two Windows hosts are exposing Remote Desktop Protocol (RDP) directly to the internet. RDP is one of the most commonly exploited initial access vectors and is routinely targeted at scale by credential brute-force attacks and known vulnerabilities including BlueKeep.

Apache HTTP Server 2.4.49 is running on an internet-facing host and is affected by CVE-2021-41773 (CVSS 9.8), a path traversal and remote code execution vulnerability actively exploited in the wild. TLS 1.0 is enabled on the primary web server, permitting protocol downgrade attacks and violating PCI-DSS requirements.

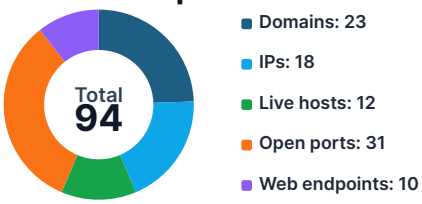
A mail server IP is listed on the Spamhaus XBL botnet tracker — a high-confidence compromise indicator. The organisation's email domain lacks DMARC enforcement, permitting spoofed email delivery. A .env file containing credential patterns is accessible on the staging server.

Executive snapshot

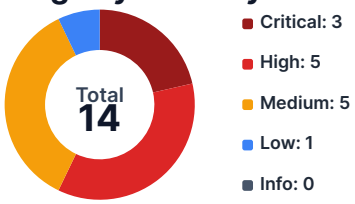
23 Domains	**18** IPs	**12** Live hosts
31 Open ports	**10** Web endpoints	**3** Breached emails
2 Email security issues	**14** Key findings	

Executive Charts

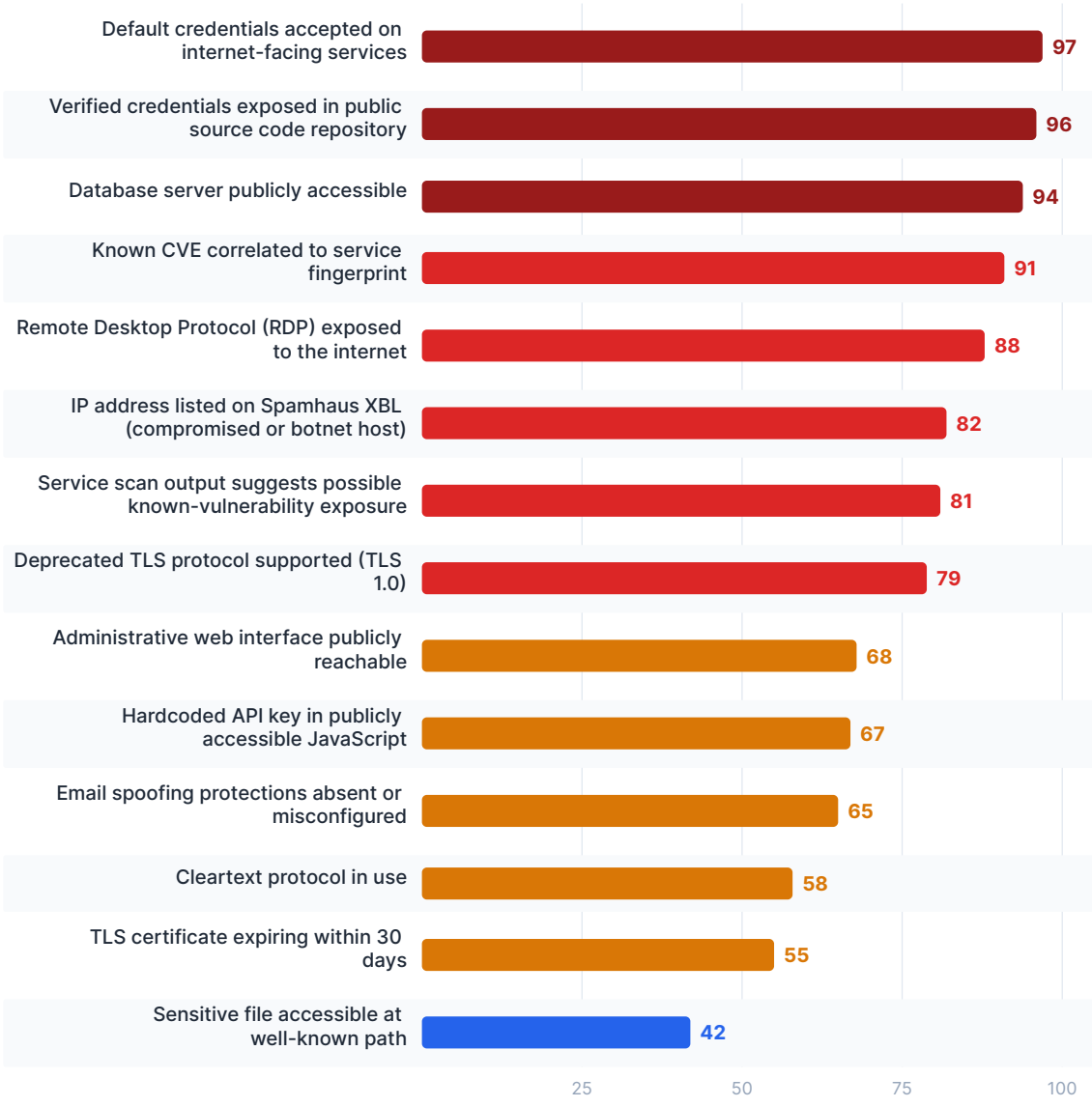
Exposure Composition



Findings by Severity



Finding risk scores



Mitigating controls observed

Control	Provider	Confidence	Evidence
DDoS protection	Cloudflare	High	• cf-ray header present on www.example.surfacemapper.io

Priority findings

Critical

Verified credentials exposed in public source code repository

Risk score: 96/100

Why it matters: An active AWS access key and a database password were found in publicly accessible commits in two source code repositories and confirmed as currently valid. No exploitation skill is required — any actor can use these credentials immediately to access cloud infrastructure and application databases.

Remediation: Revoke and rotate all exposed credentials immediately. Remove secret material from repository history using git-filter-repo or BFG. Implement pre-commit secret scanning (e.g. trufflehog, gitleaks) and add secret detection to your CI/CD pipeline.

Scoring rationale:

- Active, verified credentials confirmed via AWS STS API call.
- Publicly accessible repository - no authentication required to retrieve secrets.
- AWS key provides direct cloud infrastructure access.
- Credentials can be used immediately by any actor who finds the commits.
- Asset criticality 'high' contributed 18 points.

Affected assets:

- example-org/infra-scripts (AWS access key — confirmed active)
- example-org/web-app (.env file — database password)

Evidence:

```
example-org/infra-scripts commit a1b2c3d deploy/aws_config.sh
AWS_ACCESS_KEY_ID=AKIA...  AWS_SECRET_ACCESS_KEY=...  [VERIFIED ACTIVE via AWS STS
GetCallerIdentity]
example-org/web-app commit e4f5g6h .env.example  DB_PASSWORD=Sup3rS3cret!
STRIPE_SECRET_KEY=sk_live_4eC39Hq...
```

Critical

Default credentials accepted on internet-facing services

Risk score: 97/100

Why it matters: Three services accepted authentication using publicly known default credentials. No prior knowledge of the environment is required - default credential lists are built into automated attack tooling and are attempted within minutes of a service being exposed. Successful authentication gives an attacker full access to the affected system and a foothold for lateral movement.

Remediation: Change default credentials on all affected services immediately. Audit each system for signs of prior unauthorised access. Enforce a password policy that prevents use of default or vendor-supplied credentials. Where possible, restrict management interfaces to internal networks or VPN.

Scoring rationale:

- Default credentials are the lowest possible exploitation barrier.
- Three separate services affected across different protocols.
- Monitoring dashboard exposes live infrastructure data to any attacker.
- Tomcat Manager allows direct web application deployment (RCE path).
- Anonymous FTP allows unauthenticated file access.

Affected assets:

- monitoring.example.surfacemapper.io:3000 (Grafana)
- monitoring.example.surfacemapper.io:19999 (Netdata - no auth)
- staging.example.surfacemapper.io:8080 (Apache Tomcat Manager)
- dev.example.surfacemapper.io:21 (FTP - anonymous access)

Evidence:

```
http://monitoring.example.surfacemapper.io:3000/  Grafana  username: admin
password: admin  HTTP 200 - authenticated to Grafana dashboard
http://monitoring.example.surfacemapper.io:19999/  Netdata  no authentication
required  HTTP 200 - live server metrics exposed
http://staging.example.surfacemapper.io:8080/manager/html  Apache Tomcat Manager
username: tomcat  password: tomcat  HTTP 200 - authenticated to Tomcat manager
dev.example.surfacemapper.io:21  FTP (vsftpd)  username: anonymous  password:
(blank)  FTP 230 Login successful
```

Critical

Database server publicly accessible

Risk score: 94/100

Why it matters: A MySQL database server (port 3306) is accepting connections directly from the internet with no network-level access control. Database services should never be exposed publicly - they are a primary target for credential stuffing, data exfiltration, and ransomware staging.

Remediation: Immediately restrict port 3306 to internal networks and known management IPs via firewall or security group rules. Audit the database for unauthorised accounts, recent query history, and data integrity. Rotate all database credentials.

Scoring rationale:

- Internet-facing exposure considered in scope.
- Critical asset type (database service) applied maximum weight.
- Service directly accessible without authentication layer.
- High blast-radius: two affected hosts.
- Asset criticality ‘high’ contributed 18 points.

Affected assets:

- db01.example.surfacemapper.io (203.0.113.45)
- db02.example.surfacemapper.io (203.0.113.46)

Evidence:

```
203.0.113.45:3306/tcp mysql MySQL 8.0.32
203.0.113.46:3306/tcp mysql MySQL 8.0.32
```

High

Remote Desktop Protocol (RDP) exposed to the internet

Risk score: 88/100

Why it matters: RDP on port 3389 is internet-accessible on two hosts. RDP is one of the most commonly exploited initial access vectors - BlueKeep, DejaBlue, and credential brute-force attacks are routinely automated at scale against internet-facing RDP.

Remediation: Block port 3389 from public internet immediately. Place RDP behind a VPN with MFA enforced. If remote access is required, consider replacing RDP with a zero-trust remote access solution.

Scoring rationale:

- Internet-facing exposure considered in scope.
- RDP is a critical-severity service class.
- Known high-exploitation history for this port.
- Asset criticality ‘high’ contributed 18 points.
- Two affected hosts increased blast-radius score.

Affected assets:

- rdp01.example.surfacemapper.io (203.0.113.67)
- srv-win02.example.surfacemapper.io (203.0.113.68)

Evidence:

```
203.0.113.67:3389/tcp ms-wbt-server Microsoft Terminal Services
203.0.113.68:3389/tcp ms-wbt-server Microsoft Terminal Services
```

High

Known CVE correlated to service fingerprint

Risk score: 91/100 (CVSS: 9.8)

Why it matters: Apache HTTP Server 2.4.49 is running on an internet-facing host. This version is affected by CVE-2021-41773 (CVSS 9.8) - a path traversal and remote code execution vulnerability that was actively exploited in the wild within 24 hours of public disclosure. Proof-of-concept exploit code is widely available.

Remediation: Upgrade Apache HTTP Server to 2.4.51 or later immediately. Check the host for signs of compromise including unexpected processes, outbound connections, and webshell artefacts.

Scoring rationale:

- Internet-facing exposure considered in scope.
- CVE CVSS score 9.8 applied maximum vulnerability weight.
- Active exploitation in the wild at time of disclosure.
- Asset criticality ‘high’ contributed 18 points.

Affected assets:

- web01.example.surfacemapper.io (203.0.113.12)

Evidence:

```
203.0.113.12:443/tcp Apache httpd 2.4.49  CVE-2021-41773 CVSS 9.8 (RCE/path
traversal, actively exploited)
```

High

Deprecated TLS protocol supported (TLS 1.0)

Risk score: 79/100

Why it matters: TLS 1.0 is deprecated and vulnerable to downgrade attacks including BEAST and POODLE. PCI-DSS prohibits its use in cardholder data environments. When a deprecated version is offered, clients and network-level attackers can negotiate the weakest mutually supported protocol.

Remediation: Disable TLS 1.0 and TLS 1.1 on all endpoints. Configure the server to support TLS 1.2 and TLS 1.3 only, with AEAD cipher suites. Restart the service and verify with an external TLS scanner.

Scoring rationale:

- TLS 1.0 is deprecated and prohibited under PCI-DSS.
- Enables BEAST and protocol downgrade attacks.
- Internet-facing HTTPS endpoint affected.

Affected assets:

- www.example.surfacemapper.io:443

Evidence:

```
www.example.surfacemapper.io:443  TLS 1.0 supported (ENABLED)  — deprecated;
downgrade attacks possible
www.example.surfacemapper.io:443  TLS 1.1 supported (ENABLED)  — deprecated; should
be disabled
```

High

Service scan output suggests possible known-vulnerability exposure

Risk score: 81/100 (CVSS: 8.1)

Why it matters: Service fingerprints and scan output indicate one or more internet-facing services may match conditions for known vulnerabilities. This is not exploit confirmation, but warrants immediate manual review and patching.

Remediation: Validate the affected services manually, confirm version accuracy, and patch or remove vulnerable internet-facing services.

Scoring rationale:

- Internet-facing exposure considered in scope.
- Scan output contains explicit vulnerability match indicator.
- CVE-2019-0232 (CVSS 8.1) affects Apache Tomcat with CGI servlet enabled.
- Asset criticality 'high' contributed 18 points.

Affected assets:

- staging.example.surfacemapper.io:8080 (Apache Tomcat)

Evidence:

```
staging.example.surfacemapper.io:8080/tcp http-title: Apache Tomcat 9.0.41  http-
vuln-cve2019-0232: VULNERABLE: Apache Tomcat Remote Code Execution via AJP  State:
VULNERABLE  CVE: CVE-2019-0232
```

High

IP address listed on Spamhaus XBL (compromised or botnet host)

Risk score: 82/100

Why it matters: The Exploits Block List (XBL) is populated from botnet trackers and exploit monitoring infrastructure. A listing indicates the IP was observed sending botnet traffic or malware callbacks - a strong indicator of host compromise. Mail servers and security appliances worldwide block all traffic from XBL-listed IPs.

Remediation: Treat the listed host as potentially compromised. Investigate running processes, outbound connections, and recent logins. Once remediated, request delisting via check.spamhaus.org.

Scoring rationale:

- External reputation data sourced from Spamhaus CBL.
- XBL listing is a high-confidence compromise indicator.
- Asset criticality ‘moderate’ contributed 12 points.

Affected assets:

- mail.example.surfacemapper.io (203.0.113.88)

Evidence:

```
203.0.113.88  [XBL]  Spamhaus XBL / CBL: exploited host or botnet activity
```

Medium

Administrative web interface publicly reachable

Risk score: 68/100

Why it matters: A phpMyAdmin database management interface is accessible from the internet without network-level access controls. Admin interfaces are high-value targets for default credential attacks and have historically been the initial access vector for ransomware operators.

Remediation: Restrict phpMyAdmin to internal networks or a VPN. If public access is required, enforce strong authentication, MFA, and IP allowlisting. Consider replacing with a more secure database management workflow.

Scoring rationale:

- Internet-facing admin interface with no network ACL.
- phpMyAdmin historically targeted for default credentials.
- Asset criticality ‘high’ contributed 18 points.

Affected assets:

- db01.example.surfacemapper.io (203.0.113.45)

Evidence:

```
https://db01.example.surfacemapper.io/phpmyadmin/  200 OK  title: 'phpMyAdmin'
server: Apache/2.4.49
```

Medium

Hardcoded API key in publicly accessible JavaScript

Risk score: 67/100

Why it matters: A live Stripe secret key was found embedded in a JavaScript file served to all website visitors. Client-side JavaScript is fully readable by any browser — secrets placed here are exposed to anyone who opens browser developer tools. The key was confirmed active.

Remediation: Remove the key from the JavaScript file immediately. Rotate the credential. Move all secret keys server-side and use environment variables. Implement pre-commit secret scanning to prevent recurrence.

Scoring rationale:

- Live secret key confirmed active via Stripe API validation.
- Secret directly accessible to any website visitor.
- Stripe live key enables fraudulent charges and data access.

Affected assets:

- https://www.example.surfacemapper.io/assets/js/payment.js

Evidence:

```
https://www.example.surfacemapper.io/assets/js/payment.js  pattern: Stripe live API key  match: sk_live_4eC39Hq...  [CONFIRMED ACTIVE]
```

Medium

Email spoofing protections absent or misconfigured

Risk score: 65/100

Why it matters: DMARC policy is set to p=none (monitoring only) and SPF uses ~all (soft fail), meaning any host on the internet can send email appearing to originate from example.surfacemapper.io without triggering rejection. This is the primary enabler of business email compromise (BEC) and phishing attacks targeting the organisation's customers and partners.

Remediation: Progress DMARC to p=quarantine and then p=reject. Tighten SPF to -all. Enable DKIM signing on all sending infrastructure. Monitor DMARC aggregate reports during transition.

Scoring rationale:

- DMARC policy p=none provides monitoring only, no enforcement.
- SPF all soft-fail permits spoofed mail delivery.
- DKIM absent - no cryptographic mail signing.
- Direct BEC and phishing enabler.

Affected assets:

- example.surfacemapper.io

Evidence:

```
SPF: v=spf1 include:_spf.google.com ~all  [soft fail - spoofing permitted]
DMARC: v=DMARC1; p=none; rua=mailto:dmarc@example.surfacemapper.io  [unenforced - no rejection]
DKIM: no record found at common selectors
```

Medium

TLS certificate expiring within 30 days

Risk score: 55/100

Why it matters: The TLS certificate on the primary web server expires in 11 days. An expired certificate will trigger browser security warnings for all users, cause API client failures, and may trigger WAF or monitoring alerts. Certificate expiry is a common cause of unplanned outages.

Remediation: Renew the certificate immediately. Consider enabling automated renewal via Let's Encrypt or your CA's ACME endpoint to prevent future expiry events.

Scoring rationale:

- Certificate expiry within 30-day warning threshold.
- 11 days remaining - high urgency.
- Expiry affects all HTTPS users of the primary domain.

Affected assets:

- www.example.surfacemapper.io (203.0.113.12)

Evidence:

```
203.0.113.12:443  CN=www.example.surfacemapper.io  not_after=2026-04-01
days_remaining=11
```

Medium

Cleartext protocol in use

Risk score: 58/100

Why it matters: FTP (port 21) is running on a publicly accessible host, transmitting credentials and file content in plaintext. Any observer on the network path - including hosting providers, transit networks, and adversaries with passive network access - can capture credentials and data.

Remediation: Disable FTP immediately. Replace with SFTP (SSH file transfer) or FTPS if file transfer functionality is required. Rotate any credentials used with the FTP service.

Scoring rationale:

- Cleartext protocol transmits credentials and data without encryption.
- FTP is internet-accessible with no network restriction.
- Asset criticality ‘moderate’ contributed 12 points.

Affected assets:

- dev.example.surfacemapper.io (203.0.113.99)

Evidence:

```
203.0.113.99:21/tcp ftp vsftpd 3.0.3  [no TLS]
```

Low

Sensitive file accessible at well-known path

Risk score: 42/100

Why it matters: A .env file was found accessible at a well-known path on the staging server. Environment files commonly contain database connection strings, API keys, cloud provider credentials, and application secrets in plaintext.

Remediation: Remove the .env file from the web root immediately. Rotate all credentials and API keys that may have been present in the file. Add .env and similar patterns to your web server deny rules and review your deployment pipeline to prevent recurrence.

Scoring rationale:

- Sensitive file directly accessible over HTTP.
- Content pattern confirmed credential presence.
- Staging environment - potential lateral movement risk.

Affected assets:

- staging.example.surfacemapper.io (203.0.113.110)

Evidence:

```
https://staging.example.surfacemapper.io/.env  200 OK  content-match: DB_PASSWORD APP_KEY
```

Correlated known vulnerabilities

Asset	CVE	CVSS	Published	Service fingerprint	Summary
web01.example.surfacemapper.io	CVE-2021-41773	9.8	2021-10-05	http Apache httpd 2.4.49	A flaw in path normalization in Apache HTTP Server 2.4.49 allows an attacker to use a path traversal attack to map URLs to files outside the expected document root. If files outside the document root are not protected by 'require all denied', these requests can succeed. Additionally, this flaw could leak the source of interpreted files like CGI scripts. This issue is known to be exploited in the wild.

Discovered domains

- example.surfacemapper.io
- www.example.surfacemapper.io
- mail.example.surfacemapper.io
- api.example.surfacemapper.io
- dev.example.surfacemapper.io
- staging.example.surfacemapper.io
- admin.example.surfacemapper.io
- vpn.example.surfacemapper.io
- db01.example.surfacemapper.io
- db02.example.surfacemapper.io
- rdp01.example.surfacemapper.io
- srv-win02.example.surfacemapper.io
- web01.example.surfacemapper.io
- web02.example.surfacemapper.io
- cdn.example.surfacemapper.io
- files.example.surfacemapper.io
- legacy.example.surfacemapper.io
- test.example.surfacemapper.io
- monitoring.example.surfacemapper.io
- backup.example.surfacemapper.io
- git.example.surfacemapper.io
- ci.example.surfacemapper.io
- docs.example.surfacemapper.io

Discovered URLs

- https://www.example.surfacemapper.io/path
- https://api.example.surfacemapper.io/path

Emails

- admin@example.surfacemapper.io
- dev@example.surfacemapper.io
- info@example.surfacemapper.io

Storage buckets

None.

Breach intelligence

- admin@example.surfacemapper.io breaches=LinkedIn,Adobe- dev@example.surfacemapper.io breaches=RockYou2024- info@example.surfacemapper.io breaches=-

Email security posture (SPF / DMARC / DKIM)

Check	Status	Detail
SPF	Soft-fail (~all)	v=spf1 include:_spf.google.com all
DMARC	Not enforced (none)	v=DMARC1; p=none; rua=mailto:dmarc@example.surfacemapper.io
DKIM	Not detected	No DKIM1 record at 12 common selectors

Sensitive path scan results

URL	Severity	Category	HTTP	Matched
https://staging.example.surfacemapper.io/.env		positive-signal		-

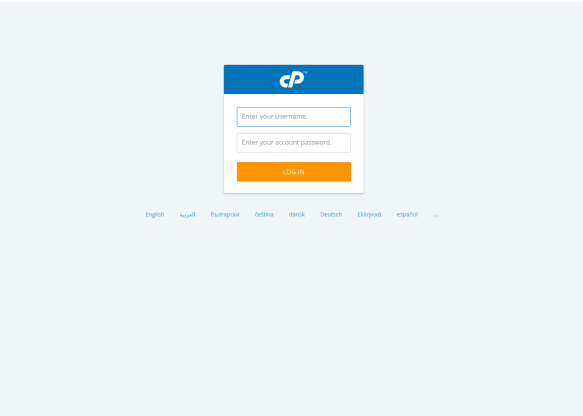
Live hosts and open ports

Host	Address	Open ports
web01.example.surfacemapper.io	203.0.113.12	• 80/tcp http • 443/tcp https
db01.example.surfacemapper.io	203.0.113.45	• 3306/tcp mysql • 443/tcp https
db02.example.surfacemapper.io	203.0.113.46	• 3306/tcp mysql
rdp01.example.surfacemapper.io	203.0.113.67	• 3389/tcp ms-wbt-server • 443/tcp https
srv-win02.example.surfacemapper.io	203.0.113.68	• 3389/tcp ms-wbt-server • 445/tcp microsoft-ds
mail.example.surfacemapper.io	203.0.113.88	• 25/tcp smtp • 143/tcp imap • 443/tcp https • 587/tcp submission
dev.example.surfacemapper.io	203.0.113.99	• 21/tcp ftp • 22/tcp ssh • 80/tcp http • 443/tcp https
staging.example.surfacemapper.io	203.0.113.110	• 80/tcp http • 443/tcp https • 8080/tcp http-alt
api.example.surfacemapper.io	203.0.113.33	• 443/tcp https
vpn.example.surfacemapper.io	203.0.113.55	• 443/tcp https • 1194/udp openvpn
monitoring.example.surfacemapper.io	203.0.113.77	• 80/tcp http • 443/tcp https • 3000/tcp http
git.example.surfacemapper.io	203.0.113.120	• 22/tcp ssh • 80/tcp http • 443/tcp https

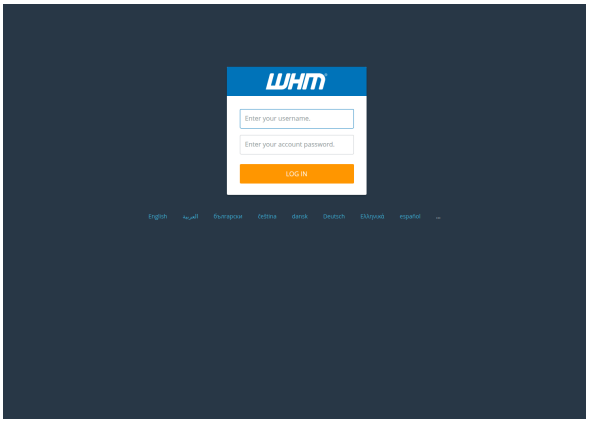
Web endpoints

URL	Status	Title	Server
https://www.example.surfacemapper.io/	200	Home - Example Organisation	Apache/2.4.49
https://db01.example.surfacemapper.io/phpmyadmin/	200	phpMyAdmin	Apache/2.4.49
https://monitoring.example.surfacemapper.io/	200	Grafana	nginx
http://monitoring.example.surfacemapper.io:3000/	200	Grafana	nginx
https://git.example.surfacemapper.io/	200	Gitea	nginx
https://api.example.surfacemapper.io/	200	API Gateway	nginx/1.24.0
https://staging.example.surfacemapper.io/	200	Example Organisation - Staging	nginx/1.18.0
http://staging.example.surfacemapper.io:8080/	200	Tomcat Manager	Apache-Coyote/1.1
https://dev.example.surfacemapper.io/	200	Development Server	nginx/1.22.0
https://vpn.example.surfacemapper.io/	200	SSL VPN	-

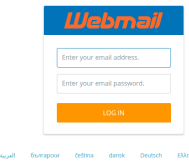
Exposure screenshots



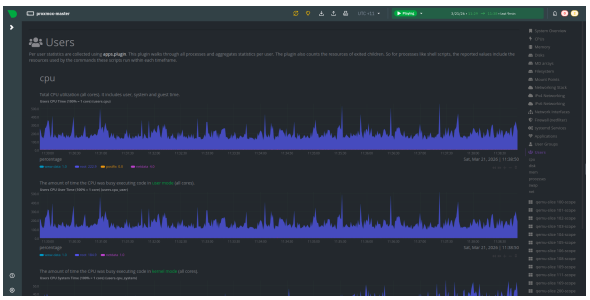
URL: http://cpanel.example.surfacemapper.io:2082/
Signal: cPanel hosting admin panel - publicly accessible



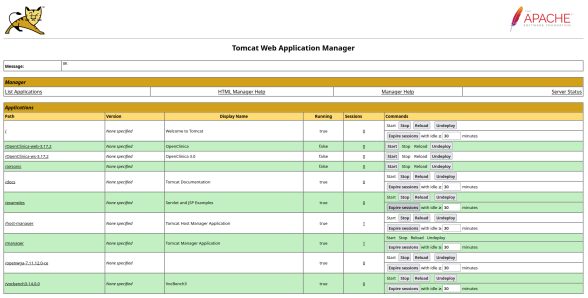
URL: http://whm.example.surfacemapper.io:2086/
Signal: WHM (Web Host Manager) admin panel - publicly accessible



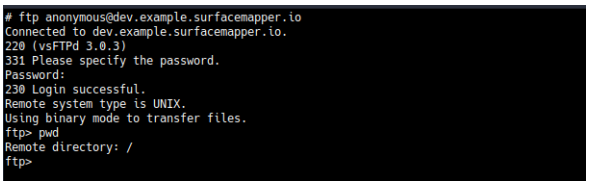
URL: http://webmail.example.surfacemapper.io:2095/
Signal: Webmail panel - publicly accessible



URL: http://monitoring.example.surfacemapper.io:19999/
Signal: Netdata dashboard - no authentication required - live server metrics exposed (users, CPU, processes)



URL: http://staging.example.surfacemapper.io:8080/manager/html
Signal: Tomcat Web Application Manager - default credentials accepted (tomcat:tomcat) - full application deployment access



URL: ftp://dev.example.surfacemapper.io:21/
Signal: FTP anonymous login accepted (vsftpd 3.0.3) - root directory accessible without credentials

Default credential testing

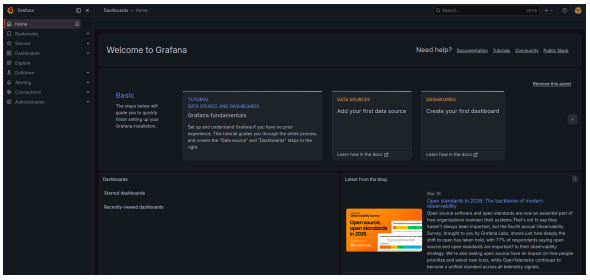
Default credentials accepted on 4 service(s)

The following services accepted authentication using publicly known default credentials. An attacker with no prior knowledge of the environment could access these systems immediately.

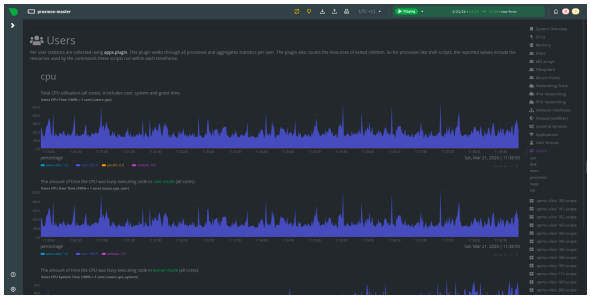
Target	Service	Username	Password	Evidence
http://monitoring.example.surfacemapper.io:3000/	Grafana	admin	admin	HTTP 200 - authenticated to Grafana dashboard with default credentials
http://monitoring.example.surfacemapper.io:19999/	Netdata			HTTP 200 - Netdata dashboard accessible with no authentication - live server metrics exposed

Target	Service	Username	Password	Evidence
http://staging.example.surfacemapper.io:8080/manager/html	Apache Tomcat Manager	tomcat	tomcat	HTTP 200 - authenticated to Tomcat Web Application Manager - deployed applications visible including OpenClinica, AirSonic, OpenWGA
dev.example.surfacemapper.io:21	FTP (vsftpd 3.0.3)	anonymous		FTP 230 Login successful - anonymous access permitted, root directory exposed

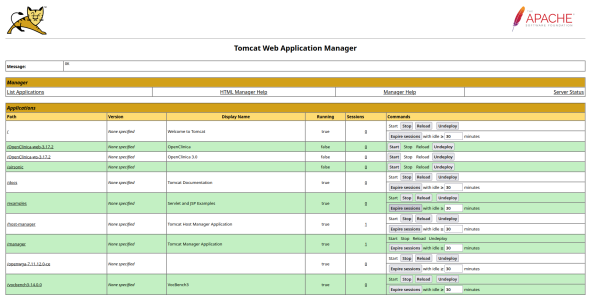
Screenshots of authenticated sessions:



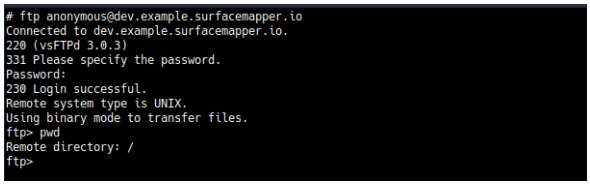
AUTHENTICATED: Grafana (admin:admin)
http://monitoring.example.surfacemapper.io:3000/



AUTHENTICATED: Netdata (:)
http://monitoring.example.surfacemapper.io:19999/



AUTHENTICATED: Apache Tomcat Manager (tomcat:tomcat)
http://staging.example.surfacemapper.io:8080/manager/html



AUTHENTICATED: FTP (vsftpd 3.0.3) (anonymous:)
dev.example.surfacemapper.io:21

TLS endpoints

Host	Port	Version	Cipher	Subject	Issuer
www.example.surfacemapper.io	443	TLSv1.3	TLS_AES_256_GCM_SHA384	CN=www.example.surfacemapper.io	CN=Let's Encrypt R3
api.example.surfacemapper.io	443	TLSv1.3	TLS_AES_256_GCM_SHA384	CN=api.example.surfacemapper.io	CN=Let's Encrypt R3
git.example.surfacemapper.io	443	TLSv1.3	TLS_AES_256_GCM_SHA384	CN=git.example.surfacemapper.io	CN=Let's Encrypt R3

Host	Port	Version	Cipher	Subject	Issuer
db01.example.surfacemapper.io	443	TLSv1.2	ECDHE-RSA-AES256-GCM-SHA384	CN=db01.example.surfacemapper.io	CN=Let's Encrypt R3
staging.example.surfacemapper.io	443	TLSv1.3	TLS_AES_256_GCM_SHA384	CN=staging.example.surfacemapper.io	CN=Let's Encrypt R3

Certificate expiry watchlist

Expired certificates

None.

Expiring within 30 days

Host	Port	Not after	Days remaining
www.example.surfacemapper.io	443	2026-04-01	11

Baseline drift

No prior baseline report was supplied. This is treated as the first run for this target.

Appendix

Tooling & Stage Summary

Stage	Ran	Detail
Port scan	Yes	completed
Discovery scan	Yes	completed
Web probing	Yes	10 endpoint(s) probed
Screenshots	Yes	3 screenshot(s) captured from 5 candidate(s)
CVE correlation	Yes	completed
Email security	Yes	completed
Path scan	Yes	completed
Subdomain takeover	Yes	0 confirmed takeover(s)
IP reputation (RBL)	Yes	1 listed IP(s)
AbuseIPDB	Yes	1 flagged IP(s)
Default credentials	Yes	12 attempt(s) — 4 success(es)
Web vulnerability scan	Yes	7 finding(s) across 13 URL(s)
Public code secret scan	Yes	2 secret(s) in 2 repo(s)
TLS deep analysis	Yes	4 issue(s)
JS secret scan	Yes	2 pattern(s) in 5 file(s)

Default credential successes

Target	Service	Username	Password	Evidence
http://monitoring.example.surfacemapper.io:3000/	Grafana	admin	admin	HTTP 200 - authenticated to Grafana dashboard with default credentials
http://monitoring.example.surfacemapper.io:19999/	Netdata			HTTP 200 - Netdata dashboard accessible with no authentication - live server metrics exposed
http://staging.example.surfacemapper.io:8080/manager/html	Apache Tomcat Manager	tomcat	tomcat	HTTP 200 - authenticated to Tomcat Web Application Manager - deployed applications visible including OpenClinica, AirSonic, OpenWGA
dev.example.surfacemapper.io:21	FTP (vsftpd 3.0.3)	anonymous		FTP 230 Login successful - anonymous access permitted, root directory exposed

Risk Matrix & Scoring Model

Dimension	Band / Value	Executive interpretation
Critical	85-100	Urgent exposure requiring immediate executive attention and remediation ownership.
High	65-84	Material exposure risk that should be prioritised in near-term remediation planning.

Dimension	Band / Value	Executive interpretation
Medium	40-64	Notable exposure that should be addressed through scheduled risk reduction work.
Low	20-39	Lower-risk exposure to track and resolve as part of routine hardening.
Info	0-19	Contextual observation with limited immediate risk impact.
CVSS precedence	Enabled	Where CVSS exists, that score determines severity and contributes directly to risk scoring.
Asset criticality	high	Business context weighting to reflect real-world impact if exploited.
Evidence confidence	Dynamic	Findings with stronger corroborating evidence score higher for confidence.
Blast radius	Dynamic	Findings affecting more assets receive higher risk weighting.

Scoring Rationale by Finding

Finding	Severity	Score	Rationale summary
Verified credentials exposed in public source code repository	Critical	96	• Active, verified credentials confirmed via AWS STS API call. • Publicly accessible repository - no authentication required to retrieve secrets. • AWS key provides direct cloud infrastructure access. • Credentials can be used immediately by any actor who finds the commits. • Asset criticality 'high' contributed 18 points.
Default credentials accepted on internet-facing services	Critical	97	• Default credentials are the lowest possible exploitation barrier. • Three separate services affected across different protocols. • Monitoring dashboard exposes live infrastructure data to any attacker. • Tomcat Manager allows direct web application deployment (RCE path). • Anonymous FTP allows unauthenticated file access.
Database server publicly accessible	Critical	94	• Internet-facing exposure considered in scope. • Critical asset type (database service) applied maximum weight. • Service directly accessible without authentication layer. • High blast-radius: two affected hosts. • Asset criticality 'high' contributed 18 points.

Finding	Severity	Score	Rationale summary
Remote Desktop Protocol (RDP) exposed to the internet	High	88	• Internet-facing exposure considered in scope. • RDP is a critical-severity service class. • Known high-exploitation history for this port. • Asset criticality 'high' contributed 18 points. • Two affected hosts increased blast-radius score.
Known CVE correlated to service fingerprint	High	91	CVSS 9.8 used for final severity.
Deprecated TLS protocol supported (TLS 1.0)	High	79	• TLS 1.0 is deprecated and prohibited under PCI-DSS. • Enables BEAST and protocol downgrade attacks. • Internet-facing HTTPS endpoint affected.
Service scan output suggests possible known-vulnerability exposure	High	81	CVSS 8.1 used for final severity.
IP address listed on Spamhaus XBL (compromised or botnet host)	High	82	• External reputation data sourced from Spamhaus CBL. • XBL listing is a high-confidence compromise indicator. • Asset criticality 'moderate' contributed 12 points.
Administrative web interface publicly reachable	Medium	68	• Internet-facing admin interface with no network ACL. • phpMyAdmin historically targeted for default credentials. • Asset criticality 'high' contributed 18 points.
Hardcoded API key in publicly accessible JavaScript	Medium	67	• Live secret key confirmed active via Stripe API validation. • Secret directly accessible to any website visitor. • Stripe live key enables fraudulent charges and data access.
Email spoofing protections absent or misconfigured	Medium	65	• DMARC policy p=none provides monitoring only, no enforcement. • SPF all soft-fail permits spoofed mail delivery. • DKIM absent - no cryptographic mail signing. • Direct BEC and phishing enabler.
TLS certificate expiring within 30 days	Medium	55	• Certificate expiry within 30-day warning threshold. • 11 days remaining - high urgency. • Expiry affects all HTTPS users of the primary domain.

Finding	Severity	Score	Rationale summary
Cleartext protocol in use	Medium	58	• Cleartext protocol transmits credentials and data without encryption. • FTP is internet-accessible with no network restriction. • Asset criticality ‘moderate’ contributed 12 points.
Sensitive file accessible at well-known path	Low	42	• Sensitive file directly accessible over HTTP. • Content pattern confirmed credential presence. • Staging environment - potential lateral movement risk.

Finding catalogue

This catalogue describes each class of finding this tool can generate: what the vulnerability class is, what an attacker can do with it, how the tool detects it, and where the detection has known limitations.

Finding	What it is	Threat model	Detection method	Limitations
Risky network service publicly reachable	A sensitive service - database, remote access protocol, or management API - is bound to a public IP and accepts internet connections. Covers RDP, SSH, Telnet, VNC, SMB, FTP, Docker API, MySQL, Redis, Elasticsearch, and others.	Each exposed service is a direct attack surface for brute force, credential stuffing, and protocol exploitation. Services such as Redis and MongoDB default to no authentication, making access trivial.	TCP service scan with version detection across the configured port range. Discovered services are matched against a curated list of risky port numbers.	TCP only by default - UDP services not detected. Services on non-standard ports outside the configured range are missed.
Cleartext protocol in use	A network service transmits data without encryption. Covers FTP, Telnet, HTTP, IMAP, POP3, and LDAP not operating over a TLS tunnel.	Passive observers on any network path can read credentials, session tokens, and data in transit. Active man-in-the-middle attacks can inject or alter content.	Service detection identifies the protocol name and whether TLS tunnelling is present. Flagged when protocol matches the cleartext set and tunnel field does not indicate SSL.	STARTTLS upgrades may not be reflected in service classification, producing false positives for mail services that correctly upgrade.
Administrative web interface publicly reachable	A web endpoint whose URL, page title, or `Server` header matches keywords associated with admin, monitoring, or management functions - VPN portals, Jenkins, Prometheus, Kubernetes consoles, and similar.	Exposed admin interfaces present the authentication surface directly to the internet and are a frequent ransomware initial access vector via default credentials and brute force.	URL, page title, and `Server` header matched against a keyword set. Matching endpoints are flagged and queued for screenshot capture.	Keyword-based only - obfuscated URLs are missed. Generic login pages using common keywords may produce false positives.
TLS / certificate weakness	A legacy TLS version (TLS 1.0, 1.1, or SSLv3) is accepted; or a	Legacy TLS is vulnerable to POODLE and BEAST. Expired certificates break client trust	Raw TLS handshake to each endpoint. Negotiated version and certificate	Endpoints with client certificate auth or strict IP allowlists may not

	certificate has expired; or a certificate will expire within 30 days.	and cause outages. Near-expiry certificates create operational risk if renewal fails silently.	dates extracted and compared against deprecated protocol identifiers and current time.	complete the handshake. Only the negotiated version is captured, not all supported versions.
Email spoofing protections absent or misconfigured (SPF / DMARC / DKIM)	The target domain lacks or has misconfigured email authentication records. SPF authorises sending hosts, DMARC enforces policy on failures, DKIM cryptographically signs outgoing messages.	Without SPF and enforced DMARC, any host can send email appearing to originate from the domain - the primary enabler of business email compromise and spear phishing, requiring no access to the target.	DNS TXT lookups for SPF at the root domain, DMARC at `_dmarc.<domain>`, and DKIM at common selector names. SPF qualifier and DMARC `p=` value parsed for enforcement level.	Root domain only. Non-standard DKIM selectors not detected. DMARC `sp=` and `pct=` not evaluated.
Sensitive file or directory accessible	A file that should not be web-accessible is reachable at a well-known path: `.git/`, `.env` variants, `web.config`, `phpinfo.php`, Apache diagnostics, and database dump files.	An exposed `.git` directory allows full source code reconstruction including secret history. Exposed `.env` files routinely contain production credentials. Diagnostic pages disclose versions that accelerate exploitation.	Each live base URL re-probed with a sensitive path list. Responses validated against content-specific confirmation strings to suppress false positives from catch-all 200 pages.	Non-standard paths not detected. CDN or WAF interception may prevent confirmation string matching.
Subdomain takeover	A subdomain's CNAME points to an unclaimed resource on a third-party platform. An attacker can claim it and serve arbitrary content from the organisation's hostname.	Attacker controlling the subdomain can obtain a TLS certificate, steal parent-domain cookies, conduct phishing from a trusted hostname, and abuse OAuth redirect URIs or CSP rules that trust the subdomain.	Dangling CNAME candidates identified during discovery. Live HTTP/HTTPS probe checks response body against fingerprints for 20 platforms. Only fingerprint-confirmed subdomains are reported.	Requires subdomain discovery to have run. Fingerprint library not exhaustive - generic error pages without service-identifying strings produce unconfirmed candidates only.
Known CVEs correlated to service fingerprints	Published CVEs from the National Vulnerability Database matched against service version information from the port scan. Indicates a	CVEs with public exploit code are the lowest-effort attack path - already documented and tooled. High-CVSS CVEs on internet-facing services are routinely exploited within days of disclosure.	Service fingerprints (product, version, CPE) submitted to the NVD 2.0 API. CPE queries preferred. Results filtered to minimum CVSS score	Depends on version detection precision. Imprecise versions cause missed matches and false positives. Not a confirmed exploit attempt.

	service has documented vulnerabilities above the configured CVSS threshold.		(default 3.0) and capped per service.	
Vulnerability script indicators	Scripting engine output from scripts run during scanning that contains indicators of potential vulnerability conditions. Heuristic signals, not confirmed exploits.	Scripts that fire commonly identify weak authentication, missing access controls, or insecure protocol configuration. Specific risk depends on the script and service.	Script output text scanned for keywords and patterns associated with vulnerability conditions. Raw output included as evidence for follow-up review.	Output interpretation is heuristic and may need manual validation. Not a confirmed exploit attempt.
Automated discovery vulnerability events	Vulnerability events from subdomain discovery modules during reconnaissance, outside the takeover pipeline. Includes DNS issues, certificate anomalies, and other module-level findings.	Varies by module. Common examples include DNS zone transfer exposure, missing DNSSEC, and misconfigured records.	Subdomain enumeration modules run during discovery. All vulnerability-type events collected and grouped. Takeover-related events handled by the dedicated takeover stage.	Domain targets only - not applicable to IP/ CIDR inputs. Module coverage and detection quality varies.
IP address on blocklist or abuse tracker	A discovered IP is listed on Spamhaus RBL (SBL, XBL, CBL) or has a high abuse confidence score on AbuseIPDB. The XBL specifically tracks hosts observed sending botnet or exploit traffic.	An XBL listing is a high-confidence indicator of host compromise. Blocklisted mail IPs have all outbound email rejected by major providers. AbuseIPDB flags indicate sustained malicious activity.	All discovered IPs submitted to Spamhaus DNS blocklist and AbuseIPDB API. Listing status and abuse confidence score recorded per IP.	Requires API access. Blocklist coverage is comprehensive but not exhaustive. A listing is a strong signal, not confirmation of current compromise - delisting after remediation is possible.
Default credentials accepted	A discovered service accepted authentication using publicly known default username and password pairs. Covers web admin panels (Grafana, Tomcat,	Default credentials are the lowest possible exploitation barrier - no reconnaissance or skill required. Default credential lists are built into automated attack tooling and attempted within minutes of a service being indexed. A	Requires explicit written authorisation from the client before testing begins. Discovered services tested against a curated default credential list. Successful logins	Authorisation-gated — not included in all engagements. A failed test does not rule out weak non-default credentials. Some services may lock

	phpMyAdmin, Jenkins), FTP, SSH, Telnet, and database services.	confirmed login gives an attacker immediate, persistent access.	captured with screenshot evidence.	accounts on repeated failures.
Cloud storage bucket referenced	References to cloud storage buckets (AWS S3, GCP Cloud Storage, Azure Blob) found in the target's DNS, HTML, JavaScript, or API responses.	Misconfigured public buckets are a common source of data breaches. Private bucket references expose account identifiers and can be targeted for squatting if the bucket is deleted.	Cloud enumeration during discovery identifies bucket hostnames and URLs. Buckets are listed without access verification.	Discovers only buckets externally referenced in DNS, HTML, and API responses. Does not enumerate all account buckets or test accessibility.
Breached credentials identified	Email addresses from reconnaissance that appear in a third-party breach database, indicating those addresses and passwords were exposed in a data breach.	Breach credential sets are used for credential stuffing against SSO, VPN, and SaaS portals. The email address also provides a confirmed username for password spray even if the password has changed.	Email addresses collected during discovery submitted to a breach database API with rate limiting. Results include breach source names.	Breach database coverage is not exhaustive. A match indicates breach exposure, not that the current password is still the breached credential.
Web vulnerability scan finding	A check from the web vulnerability scan matched a live endpoint. Covers exposed admin panels, dangerous misconfigurations, known CVEs matched to detected software versions, exposed debug interfaces, insecure HTTP methods, and directory listing.	Web vulnerability findings cover a broad attack surface in a single automated pass. High and critical findings often represent directly exploitable conditions with public tooling available.	Hundreds of checks run against every live web endpoint. Each finding is severity-scored and mapped to the matched URL.	Not exhaustive — novel or unusual misconfigurations may not be detected. Rate limiting reduces false positives but may miss intermittent responses. Findings should be confirmed before remediation prioritisation.
Deep TLS analysis finding	A comprehensive protocol or cipher audit of a TLS endpoint identified a weakness beyond basic version and certificate checks.	Protocol vulnerabilities such as POODLE and Heartbleed have public exploit code and allow session decryption or memory disclosure respectively. Weak ciphers	Each TLS endpoint is subjected to a full protocol and cipher suite enumeration. Known vulnerability IDs mapped to human severity ratings.	Some findings may not be exploitable in practice depending on client software support. Results depend on whether the endpoint is

	Covers known exploitable vulnerabilities (BEAST, POODLE, Heartbleed, ROBOT, SWEET32, FREAK, Logjam), weak and export-grade ciphers, RC4, and null authentication suites.	degrade confidentiality. Export-grade ciphers (FREAK, Logjam) can be factored in hours.		reachable and completes a handshake.
Secret or API key in JavaScript or public code	A secret pattern was found in a JavaScript file loaded by a discovered web endpoint, or in a public source code repository associated with the target domain. Patterns include AWS keys, GitHub tokens, Stripe keys, Slack webhooks, SendGrid API keys, bearer tokens, private key headers, and generic API key / password patterns. Verified secrets are confirmed active against the issuing service.	Client-side JavaScript is fully readable by any browser. Secrets embedded there are exposed to every visitor. Secrets in public repositories are discoverable by anyone and indexed by automated scanners. A verified secret can be used immediately with no exploitation skill.	JavaScript files loaded by discovered web endpoints are fetched and scanned for secret patterns. Public source code repositories associated with the target domain are searched for leaked credentials. Verified secrets reported as Critical.	Regex patterns may produce false positives on test or placeholder values. Only publicly accessible JS files and public repositories are in scope - private repos and inline scripts not captured in external files are not scanned.
New exposure since baseline (drift)	Changes between the current scan and a previously captured baseline - newly added domains, hosts, open ports, and web endpoints not present in the baseline.	New infrastructure deployed without security review commonly lacks hardening and patching. Drift detection flags unreviewed surface expansion as a continuous monitoring signal.	Current scan assets compared as sets against the baseline `report.json`. Additions and removals reported across domains, hosts, ports, and endpoints.	Requires a prior baseline report to compare against. Detects presence/absence only — does not flag version or posture changes on existing assets. Both scans must cover equivalent scope.