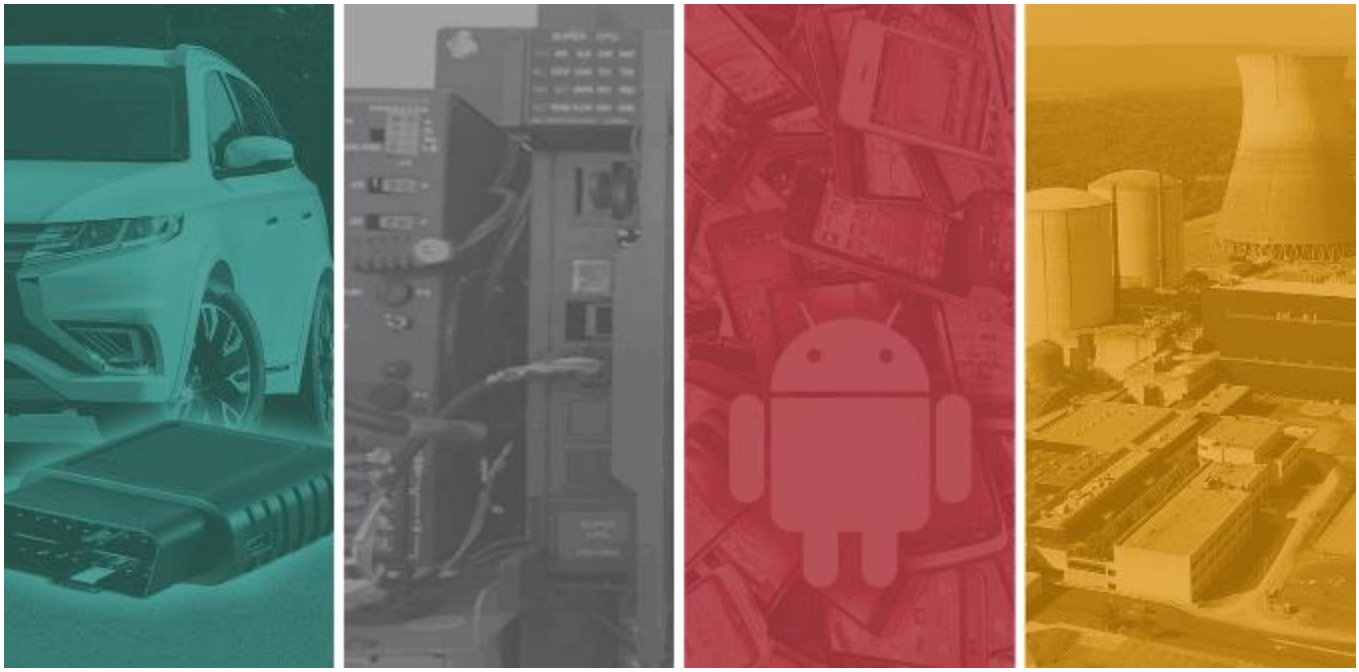
PEN TEST PARTNERS

G-Cloud 14 (RM1557.14)

Lot 3 Cloud Support – Security Services Definition Document – Penetration Testing



Version 1.0
05 April 2024

Contact: bidteam@pentestpartners.com

Carbon Neutral Organisation

CSS CREST

Why Choose Pen Test Partners?

What we provide

Responsiveness
We are recognised as being extremely responsive.
We can begin an engagement with as little as 24 hours notice.

Follow-up
One of the reasons we have such loyal clients is the availability our consultants have for follow-up work.

Ongoing guidance
Once an engagement is over...
It's never truly over, there will always be questions and queries and advice needed, so we make a point of always being available post-test.

Table of Contents

1. Service Definition Overview

This document set out the range of Pen Testing services offered by the PTP Pen Test team. Services can be purchased individually or grouped based on our clients' specific needs. These services assess the security posture of a variety of assets (Infrastructure, Web & API etc), the result of which is documented in a report that details the risk presented by the asset and potential business impact should the asset be exploited, a list of all vulnerabilities found during the assessment, the corresponding risk score of each vulnerability, and remediation recommendations to address each vulnerability and harden the asset to improve its security posture.

1.1. PTP Pen Testing Services included in this document:

API Testing	Internal Infrastructure
Build Review	IT Health Check (Regulated and Nonregulated)
Cloud Breach Assessment	Kubernetes
Cloud Config Review and Testing	Mobile Application Security Testing
Code Review	Mobile Device Testing
Compiled Application Review	VPN Assessment
External Infrastructure	Web Application Assessment
Firewall & Network Device review	Wireless Testing

1.2. PTPs Approach to Working Together and Pricing

Every project we deliver is custom managed. From initial scoping through to debrief we will ensure the right approach and people are being utilised. PTP's implementation plan typically follows an established workflow based on actions from PTP and our clients, although we can of course work around a client if they have a preferential implementation model.

PTP typical working model:

- **Dedicated Account Manager:**
  - PTP will allocate a dedicated account manager as a central point of contact for the duration of the relationship.
- **Initial Consultation:**
  - For each new engagement, PTP will initiate a conversation via email and/or call to understand your service requirements. This serves as the introduction call.
- **Scope of Works (SOW) Creation:**
  - A PTP technical consultant will review all relevant information and create a detailed Scope of Works (SOW). This document will include any necessary prerequisites.
- **Proposal and SOW Delivery:**
  - PTP will send a comprehensive Proposal/SOW, specifying:
    - Duration
    - Cost
    - Grade of consultant required
    - Proposed dates (if already discussed)
- **Pricing Calculation:**
  - Pricing will be determined based on the number of days required to deliver the services using the service day rates.
- **Acceptance of Proposal / SOW**
  - Upon acceptance of the SOW, delivery dates are agreed and scheduled.
- **Authorisation and pre-requisite information**
  - PTP will send a Authorisation form for signature and return.
- **Pre engagement Kick off Call**
  - PTP will host a pre-engagement call to discuss the engagement and introduce the team.
- **PTP undertakes the required services.**

Planning and Preparation:

- Objectives – what are the customer's goals? What is at risk?
- Scope – what is going to be covered by the test?
- Timing – when is the test happening, and how long will it go on for?
- Secrecy – is the test being carried out with wider knowledge, or in secret?
- Risks – what are the risks of testing? Have steps been taken to mitigate these?
- Authorisation – have all parties authorised testing?
- Emergencies – what should be done in an emergency?

Testing Conducted:

- Information gathering and analysis. This involves gathering as much information as possible about the target systems using a multitude of tools largely dependent on the tester's references. Opensource intelligence and reconnaissance using the Internet will be involved, even for internal penetration tests.
- Research of any frameworks or technologies in use will be done by the tester.

- Vulnerability detection
    - The tester will use a combination of tools and experience to find vulnerabilities on the systems.
- Penetration
    - The tester will leverage any vulnerabilities found to penetrate the network. Often at this stage,
      further information gathering, and vulnerability detection will happen as part of an iterative process as the tester gains deeper access.
- Reporting and analysis
    - Detailed reporting is written and provided, along with remediation steps. The tester is available to debrief and discuss findings and assist with fixing them.
- The tools used by a tester are varied and largely depend on personal preference. All testers will be using Kali Linux, nmap, Nessus, Burp Suite, and Metasploit.

- Report delivery (within 5 working days of completion) and optional wash-up call.

Not all requirements need all these stages, simple small engagements for example, may be quickly understood and proposed in a short period of time as the relationship develops between PTP and the Client.

2. Pen Testing Services Summary

API Testing

Web APIs are core to the operation of web applications, mobile applications, and interactions between other systems. Weaknesses in these APIs can lead to loss of sensitive information, damage to the brand's image, denial-of-service, and loss of revenue. It is recommended that all web applications are tested at least once a year and after major code changes.

The primary source of the Pen Test Partners (PTP) web application assessment methodology is the OWASP Web Security Testing Guide (WSTG) and Mobile Application Security Testing Guide (MASTG). However, reliance on a static resource would result in vulnerabilities being missed. As a result, we use a combination of our own experience and techniques. We stay up to date with cutting-edge research, resulting in a hybrid testing methodology.

To fully test modern web services and APIs, it is essential that highly skilled and experienced security consultants are deployed. Purely automated scanning can rarely be used due to the nature of web services and, when it can be used, it cannot provide good coverage of complex APIs and is prone to showing many false positive and negative findings. Skilled manual testing ensures excellent coverage, drawing on the consultant's experience to uncover even deeply hidden issues.
Full Service Methodology detailed in Section 5.

Build Review

Defending against modern threats requires a layered defence which includes hardening of devices that an attacker may attempt to breach. PTP recommend that build reviews are conducted to evaluate the security of end devices and servers within a network. This is particularly important for large deployments, such as laptops used by a mobile workforce. Ensuring that a base build is well-secured can avoid costly remediation after systems are deployed.

Areas of Testing

Each system in scope will be examined in depth, and may include:

- Operating system patch levels
- Installed software and services
- Hardening and operating system configuration
- Stolen device protection
- Endpoint protection efficacy
- Traffic analysis for mobile devices to ensure adequate protection

Common vulnerabilities can include:

- Out-of-date software
- Misconfigured services
- Unknown or excessive services running
- Weak or default credentials

- Unused or vulnerable vendor installed “bloatware”
- Missing operating system patches

Full Service Methodology detailed in Section 5.

Cloud Breach Assessment

The PTP cloud breach assessment is a real-world engagement, to ascertain the impact a compromised user account or resource, would have on a company. The interaction is scenario-based and concentrates on role-based access controls, deployed cloud resources, and supporting infrastructure. For each scenario, the PTP consultant will act as a genuine threat actor to find potential attack paths and develop recommendations, to lessen the impact of any compromise. This builds upon your traditional testing and benchmarking engagements.

Full Service Methodology detailed in Section 5.

Cloud Config Review and Testing

Cloud computing is the on-demand delivery of computer power, applications, databases, storage, and other IT resources, with a pay-as-you-go pricing model. Cloud computing such as Microsoft Azure, Amazon Web Services, and Google Cloud Platform is becoming a common solution for many enterprises looking to move away from physical servers in data centres. This reduces costs and increases survivability in the event of a catastrophic event. However, with new technologies comes new security requirements, along with familiar security principles. Reviewing both and providing usable, secure recommendations ensures data in the cloud is kept secure.

The Key Premises of Cloud Computing:

- Decentralised
- Rapid provisioning
- Remote access
- Minimum hardware management
- Reduced IT hardware upfront costs
- Flexible and scalable
- Shared pool of configurable system resources

The Types of Cloud Systems:

- **Public:** Publicly accessible over the Internet
- **Private:** Accessible to only a specific set of people or organizations
- **Community:** Accessible to groups of organizations and individuals with similar interest
- **Hybrid:** Combination of the above models

Full Service Methodology detailed in Section 5.

Code Review

This class of testing is aimed at reviewing an application's source code to ensure that it considers security in its critical paths. This includes several steps such as tracing the path of user-controlled input through the application and verifying it is correctly handled, checking that any encryption is performed appropriately, ensuring authorisation is effective and looking for memory manipulation (buffer under- and overflows, heap overflows, integer overflows, format string manipulation).

This is important particularly where such information is used in privileged operations, for example, whether OS command injection or buffer overflows are possible to trigger from a client perspective.
Full Service Methodology detailed in Section 5.

Compiled Application Review

What should be included in Compiled Application Test?

The functionality of traditional compiled applications, including their local functions, dependencies, and interactions with other systems. With applications that make use of multiple processes, the inter-process communication methods should be analysed for opportunities to manipulate behaviour.

Areas of Testing

- Access Controls
- Cryptographic Failures and Data exposure
- Design Flaws in application functionality
- Software and Data Integrity (e.g third-party dependencies and software supply chain)

Common Vulnerabilities

- DLL Hijacking
- Privilege Escalation
- Hardcoded encryption material
- Insecure Inter-Process Communication (IPC)
- Credentials stored in configuration files

Full Service Methodology detailed in Section 5.

External Infrastructure

External or public-facing infrastructure is defined as all the servers and services that are reachable from the Internet. Network infrastructure covers the services offered at an operating system level but would not, for example, include web applications.

The internet-facing infrastructure includes:

- One or more firewalls that provide protection against Internet-borne threats and are used to restrict access
- Servers that provide various services, such as web servers, email servers, and so forth

These are generally considered to be the most 'at risk' from an attacker or malware, as it is near impossible to restrict access to the hacker while granting access to the genuine prospective client requiring your services.

As a rule, the more functionality a server or network delivers, the more likely it is to be attacked. As functionality increases, the opportunity for misconfiguration and vulnerability increases. Hence a web site running a complex transactional web application is far more likely to be vulnerable to security flaws than your upstream router. The core of any penetration test should include your public infrastructure, but do not forget that there are other routes into your network.

Full Service Methodology detailed in Section 5.

Firewall & Network Device review

A firewall has a very specific role on the network. It is responsible for controlling the flow of information between devices, and more specifically preventing devices from communicating with other devices or services, unless specifically required. This is essential for Internet-facing devices, however, is also important internally.

A poorly defined ruleset may allow excessive access to devices or services which may increase the risk of compromise. As a rule, a minimum number of hosts and services should be allowed to communicate with each other to reduce the risk of attack.

If the configuration of the firewall itself is incorrect, then a hacker may be able to exploit the firewall from a network perspective. This could be used to launch a denial-of-service attack against the network or even alter the rules to allow access to otherwise restricted networks.

To identify any risks posed by firewalls, PTP can carry out several tests against the firewall. A firewall ruleset review is always recommended. In this case, a copy of the current ruleset would have to be made available. Ideally this is exported from the device and a copy is given to the consultant, however, in restrictive environments this can be carried out from the firewall's administrative interface if necessary.

Full Service Methodology detailed in Section 5.

Internal Infrastructure

The internal network of a business is often key to their operations, containing critical systems, information and functionality. PTP therefore recommended that internal networks are subject to regular security testing to ensure all data and systems are adequately protected against intrusion by malicious attackers.

Areas of Testing

All systems within the specific network ranges are tested from an unauthenticated and, if requested, authenticated, perspective. The following aspects will be examined:

- Complete inventory of networks, with a focus on identifying all live hosts
- Enumeration of services that are hosted on servers
- Discovery of vulnerable or out of date software
- Network segregation

Common vulnerabilities can include:

- Outdated operating systems
- Default or weak credentials
- Vulnerable software due to being out of date or poorly configured
- Active Directory misconfigurations
- Weak or no encryption found on services
- Legacy or previously unknown hosts, protocols, or services presenting unnecessary risk

Full Service Methodology detailed in Section 5.

IT Health Check (Regulated and Non regulated)

PTP has one of the largest established NCSC CHECK teams in the UK and has been performing regulated and non-regulated ITHC's for over 13 years. Our check team of leaders and members are certified in both application and infrastructure testing projects.

Kubernetes testing

Kubernetes, a widely adopted container orchestration system, is integral to managing containerised applications in various environments. Vulnerabilities in Kubernetes clusters can lead to severe security breaches, including unauthorised access, data theft, and service disruption. Regular testing of Kubernetes configurations and deployments is essential to ensure the security and resilience of these systems. It is recommended to perform these tests at least annually and following significant updates or changes to the infrastructure.

The PTP Kubernetes security assessment methodology draws inspiration from established frameworks like the CIS Kubernetes Benchmark and the NSA-CISA Kubernetes Hardening Guidance. However, we understand that relying solely on these resources is insufficient. Our approach blends these standards

with our expertise and awareness of the latest security research, forming a dynamic and comprehensive testing methodology.

For thorough testing of Kubernetes clusters, it is crucial to engage with experienced security professionals. Automated tools, while helpful, often fall short in thoroughly evaluating complex Kubernetes environments. They might miss intricate vulnerabilities or generate false positives and negatives. PTPs manual testing, leveraging the insights and experience of our consultants, ensures comprehensive coverage and the discovery of subtle, yet critical security issues.
Full Service Methodology detailed in Section 5.

Mobile Application Security Testing

Our mobile application testing methodology is in line with the OWASP Mobile Application Security project which provides guidelines for mobile application assessments via the OWASP Mobile Application Security Verification Standard (MASVS) and OWASP Mobile Application Security Testing Guide (MASTG).

As mobile applications continue to gain relevance, it is increasingly important to security test them to identify and address security vulnerabilities and weaknesses before they can be exploited by attackers, potentially leading to data breaches, loss of sensitive information, financial loss, and reputational damage. A mobile penetration test can help to uncover vulnerabilities such as insecure data storage, weak authentication and authorization mechanisms, insecure communications, and other security issues that could be exploited by attackers. By identifying and fixing these vulnerabilities, mobile applications can be made more secure and better able to protect users' sensitive information.
Full Service Methodology detailed in Section 5.

Mobile Device Testing

Mobile phones and tablets are increasingly issued by businesses to their staff to improve productivity and mobile working whilst still ensuring the security of data. These devices, however, can provide a gateway for an attacker to access internal corporate resources.

A penetration test of your mobile devices can help identify any misconfigurations or other vulnerabilities, giving you assurance that they won't be used to compromise the confidentiality or integrity of your company.

Testing Overview

There are three main activities:

- Device Inspection: This is a check of the mobile device itself to ensure it is up to date and securely configured.
- Connectivity Inspection: The device will be checked to see if it connects to rouge wireless and mobile networks, if a private SIM is in use.
- MDM Assessment: When applicable, any MDM solution installed on the device will also be assessed, both the MDM configuration and the MDM application itself.

Vulnerability and extended manual testing will explicitly identify where security holes lie and remove false positives. When applicable, other methods will be utilised.
Full Service Methodology detailed in Section 5.

VPN Assessment

Remote access solutions or VPNs are commonly used to provide access to remote working staff. To review these, PTP work alongside our other methodologies such as our external infrastructure and build review methodologies when reviewing VPN endpoints from an external perspective or reviewing VPN software from an on-host build review perspective.
Full Service Methodology detailed in Section 5.

Web Application Assessment

Web applications are often critical to business and, if not properly secured, can lead to loss of sensitive information, damage to the brand's image, denial-of-service, and loss of revenue. It is recommended that all web applications are tested at least once a year and after major code changes.

The primary source of the Pen Test Partners web application assessment methodology is the OWASP Web Security Testing Guide (WSTG). However, reliance on a single static resource would result in vulnerabilities being missed. As a result, we use a combination of our own experience and techniques while keeping up to date with cutting-edge research, resulting in a hybrid testing methodology.

The testing process is driven by the application and how it functions. With many sites having complex flows and multiple user roles, it is essential that both automated and manual testing are used to discover deeply hidden issues. We search for weaknesses within the application, looking to chain vulnerabilities together to maximise their impact. Due to the rich and varied nature of web applications, we take an approach that combines frameworks such as OWASP and combine it with years of web application testing experience.

When a high level of assurance is required, source code can be used to augment web application testing. With the ability to directly observe the inner workings of a web application, it is possible to uncover weaknesses that would remain hidden as part of a normal web application test.
Full Service Methodology detailed in Section 5.

Wireless Testing

Wireless networks are commonplace in both home and corporate environments. They are more flexible and can be configured to a more secure standard than Ethernet-based local area networks. However, wireless signals cannot easily be constrained within the perimeter of a building or campus in the way that Ethernet can.

Wireless networking is nonetheless susceptible to attack due to configuration weaknesses or known or unknown security flaws in the software that implements the solution, including controllers, base stations, or access points and client stations.

Normally in a corporate environment there would be several wireless networks with different purposes, which would require adequate settings depending on the risk profile that they represent to the business. For example, a guest network that provides no connectivity to the corporate LAN, and a wireless network that provides access to corporate services would have very different risk profiles and would have different security requirements. Unlike Ethernet-based LANs, wireless networks are often accessible from locations outside office spaces, which increases their physical attack surface.

Several approaches could be taken to test wireless networks, depending on the risk profile of the network and the perceived threats to the solution. Perceived threats could be, for example, attacks from a public space with a high gain antenna or gaining access to a network by stealing and breaking into a mobile device to extract authentication information.

Full Service Methodology detailed in Section 5.

3. Reporting

Upon completion of the engagement, PTP will provide a comprehensive report that follows the format illustrated in the screenshots below.

The report provides an in-depth technical detail of any vulnerabilities that have been identified within the target environment(s).

The report begins with a Management Summary which breaks down a technical overview into layman's terms, allowing the risks uncovered during the assessment to be distributed to a wider audience, who may have differing levels of technical competence.

The summary is complimented by a more in-depth Technical report, which will provide details on how to implement all recommended fixes. Any critical risk issues uncovered during the assessment, will be raised on the day of test with the nominated third-party contact; allowing urgent remedial action to be initiated and subsequently recorded in the final report.

As you would expect from a market leader, PTP imposes a strict two-step Quality Assurance process, where both grammar and technical content are thoroughly checked on each report. For a typical test, we aim to deliver your report within 5 working days of test completion, however this may be increased to 10 working days for more complex tests such as red team or hardware related engagements.

4. Introduction to PTP

Pen Test Partners LLP (PTP) is focussed on delivering innovative and meaningful penetration testing. It's a simple mandate, and one that we have built our business and reputation with.

Why choose Pen Test Partners?

Established in June 2010, we've got consultants with a vast range of skills and experience, some with extremely niche skills.

As an entire company we know our stuff and we'll work at your pace.

We research, test, and assure a lot of interesting and complex things!

We've provided testing and assurance for all sorts of things; ships at sea, international finance infrastructure, mobile apps for smart toys, airplane systems and avionics, power stations and critical national infrastructure, automotive and telematics, mobile banking apps, physical security, cloud services to rail infrastructure.

Our Credentials

PTP are a CREST, CBEST, ASSURE, STAR, CSIR, CHECK, PCI QSA, ISO27001, Cyber Essentials/+, NCSC CIR L2 accredited ; this ensures the highest quality of testing.

All PTP staff are vetted prior to commencement of employment in line with British Standard 7858 Clearance Service and through the Disclosure and Barring Service (DBS). National Vetting Solutions. All our security consultants are a minimum of SC cleared.

What we provide

Responsiveness

We are recognised as being extremely responsive. We can begin an engagement with as little as 24 hours' notice.

Follow-up

One of the reasons we have such loyal clients is the availability our consultants have for follow-up work.

Ongoing guidance

Once an engagement is over… It's never truly over, there will always be questions, queries, and advice needed, so we make a point of always being available post engagement.

Our Reputation

The BBC and other news agencies often contact us to comment on the latest cyber news.

We are frequently called upon to provide keynotes at events such as BSides, TED talks, InfoSecurity Europe and the US Chamber of commerce.

We can't name names, but our clients come from a wide range of verticals and sizes including: Automotive, Banking, Education, Engineering, Energy, Oil & Gas, FinTech, Government, Healthcare, Manufacturing, Retail, Telco's, Insurance, Legal, Transport and Finance.

5. Full Service Methodologies

5.1. API Testing Methodology

Web APIs are core to the operation of web applications, mobile applications, and interactions between other systems. Weaknesses in these APIs can lead to loss of sensitive information, damage to the brand's image, denial-of-service, and loss of revenue. It is recommended that all web applications are tested at least once a year and after major code changes.

The primary source of the Pen Test Partners (PTP) web application assessment methodology is the OWASP Web Security Testing Guide (WSTG) and Mobile Application Security Testing Guide (MASTG). However, reliance on a static resource would result in vulnerabilities being missed. As a result, we use a combination of our own experience and techniques. We stay up to date with cutting-edge research, resulting in a hybrid testing methodology.

To fully test modern web services and APIs, it is essential that highly skilled and experienced security consultants are deployed. Purely automated scanning can rarely be used due to the nature of web services and, when it can be used, it cannot provide good coverage of complex APIs and is prone to showing many false positive and negative findings. Skilled manual testing ensures excellent coverage, drawing on the consultant's experience to uncover even deeply hidden issues.

What should be covered as part of an API Security Assessment?

A web API penetration test includes an assessment of the security of data transmission, proper implementation of authentication and authorization controls, verification of data input, handling, and output, and thorough examination of HTTP methods and status codes. APIs should be tested for vulnerabilities such as injection flaws, cross-site scripting, XML external entity attacks, and insecure deserialization, among others. Additionally, a thorough penetration test should inspect the usage of third-party components and libraries to ensure they do not have known vulnerabilities. Lastly, the penetration test should also review logging, monitoring, and alerting practices to verify the system's ability to detect and respond to security incidents promptly.

For systems that require a high level of security assurance, a code review can be carried out alongside testing of the API.

Areas of testing should include:

- Broken Object-Level Authorization
- Excessive data exposure
- Broken authentication
- Lack of rate limiting and resources
- Injection vulnerabilities
- Security misconfiguration

- Server-side request forgery
- Insecure business logic

Common vulnerabilities can include:

- Insecure Direct Object reference vulnerabilities, where object-level authorization has not been implemented correctly
- Lack of rate limiting leading to brute force or availability issues
- Injection vulnerabilities such as, but not limited to, NoSQL, SQL injection, and XXE
- Broken authentication, such as flaws in user authentication or session management
- Undocumented and unknown endpoints that can expose insecure methods to attackers
- Insecure deserialization where an attacker can execute code by passing crafted data in serialised objects
- Weak encryption configuration leading to unnecessary risk that data is intercepted or tampered with in transit
- Use of out-of-date or known vulnerable components that could be exploited to gain access to data or the underlying system

High and Critical Risk Vulnerabilities

At PTP, it is our policy to directly contact the designated client contact upon the discovery of a high or critical risk vulnerability, or one which poses an immediate threat to the environment or its users. Direct contact is made through email or phone with a detailed description of what the vulnerability is, how it would be exploited by an attacker, and how to remediate it.

Example Attacks from Previous Engagements

To illustrate the type of real-world attack, these are some possible attacks based on previous tests carried out across a range of API security assessments:

An endpoint used as the primary authorization route of an API was found to be protected by a robust account lockout policy. During testing, the consultant identified that an alternate endpoint, which was used for a “Current Password” check on the password reset mechanism, was not protected by the same account lockout policy. Due to a combined lack of rate limiting and account-wide lockout policy, the consultant was able to brute force several developer test accounts during the engagement which were found to have weak passwords set.

A common API framework was found to be in use. Using a list of known files from the Open-Source project, a brute-force search of the API was carried out. This uncovered several diagnostic pages exposing environment variables for the project, including credentials for the internal APIs that were consumed by the one being tested. Network scanning indicated that these were not exposed to the Internet. Further testing of the API found that it was possible to carry out server-side request forgery, allowing the credentials to be used against the internal APIs. This resulted in complete access to the underlying dataset that the API was intended to protect.

Testing Process

The testing is expected to follow the following phased approach:

Functionality Review and Analysis: The consultant will examine the APIs as described in the designated scope and familiarise themselves with the application and environment. Documentation, Postman/Swagger files, associated web applications, and code will be used to understand the API in depth. Where the framework, language, or

technologies in use are less commonplace, documentation and previous security research will be checked to understand how they are used, what common pitfalls there are, and if any known vulnerabilities exist.

Reconnaissance: The consultant will perform a reconnaissance phase against the API and its environment. This phase is used to identify the server software, CMS and/or the framework in use, and any undocumented or unknown end points. If the API is hosted in a production environment that has, for example, been cached by search engines, publicly available information may also be collected by the consultant during this phase of testing. This phase may also include identifying what is visible prior to authenticating in an attempt to see what an attacker from an unauthenticated perspective can identify.

Enumeration and Testing: Each API endpoint will be examined, carrying out the expected actions and also modifying the payloads, headers, and other attacker-controlled inputs. Where there are multi-stage processes (such as user registration, placing an order etc.), the documented flow will be followed as well as attempting to alter the flow to perform malicious actions. Multiple accounts will be used to ensure that authentication and authorisation are correctly carried out.

Automation: Based on the previous stages and any vulnerabilities found, it is common to develop automation to check all endpoints for specific issues or to accelerate testing of lengthy multi-stage processes.

Vulnerability Assessment: Based on previous stages, parts of the API will undergo light vulnerability scanning. This identifies any low-hanging fruit vulnerabilities that may be present. This follows a similar pattern to most vulnerability assessments, with asset discovery, application mapping, software and library identification, and manual testing.

Attack Scenario and Demonstration: With knowledge of vulnerabilities and the design of the API, viable attacks scenarios will be planned against the web application or its users. It may be possible at this stage to build an example realistic attack scenario that an attacker may formulate. Sometimes, one or more vulnerabilities identified during the engagement may be chained together to create an advanced proof of concept attack. Ultimately, the goal is to promote defence-in-depth – security controls should be present on many aspects of the API and underlying infrastructure.

Iteration: At this stage, it may be necessary to perform further vulnerability assessment. This could involve more in-depth testing of individual components or vulnerabilities and chaining of attack vectors together to create an example of a realistic real-world attack scenario. This phase is also used to ensure no false positive findings are present and to take necessary evidence used for the reporting phase.

Reporting

A report is written detailing the processes carried out and the issues found. Generally, this will contain a prioritised list of vulnerabilities discovered grouped by functional area. Remediation advice will be provided, both in terms of an immediate fix and any defence-in-depth measures that could be taken to mitigate risk. Attack chains, alongside their impact, will be documented. Any higher-level findings that can be abstracted from the testing will be provided. Any architectural or design weaknesses will be highlighted so that these can be avoided in the future. Finally, an executive summary is produced to allow the most severe issues to be communicated quickly to stakeholders.

Requirements and Prerequisites

The following is required from the client:

Accounts: This is used to permit authenticated testing. The more information that can be provided regarding each user role or privilege, the better. Information relating to the authentication process (e.g. OAuth, OpenID Connect, and SAML) and use of Multi-Factor Authentication (MFA) setup is also required where applicable.

Should the API be internally accessible only, a VPN solution or jump-box with the relevant software needed for testing should be set up (if possible) ahead of testing for the consultant to access the web application. Please note that the implementation time of such solutions can be lengthy, so it is advised that sufficient time and resources be allocated in order to ensure the proper functioning of the test machine before the go-live date.

Pre-access checks will be conducted ahead of testing by PTP and confirmed by the consultant to ensure testing runs smoothly on the commencement date.

Environment and Setup Details

- A summary of what the API purpose is and brief description of functionality
- The API stack details, examples of which may include:
    - Server software
    - Framework type
    - CMS type
    - In-house developed
- Whether the API is hosted in a production or testing environment
- Whether the API shares any backend services with neighbouring services (e.g. whether the test environment's backend DBMS is shared with production)
- Whether there is sensitive functionality that should not be tested using automated tools (e.g. system deletion or calls to third parties that attract cost)
- Where any third-party Web Application Firewalls (WAFs) are in place such as Akami, Cloudflare, or F5 BIG-IP
    - PTP testing IP addresses should be added to an allow-list prior to testing to ensure testing of the actual application takes place.
- API documentation for web services that may be used by the web application as part of its overall functionality:
    - Swagger documentation, Postman collections
    - Test harnesses and documentation

5.2. Build Review Methodology

As a CREST accredited provider, Pen Test Partners (PTP) can deliver testing to the highest standards. In addition to CREST, we are certified by other bodies to deliver high quality cyber security services. These include CHECK and Cyber Essentials Plus.

Defending against modern threats requires a layered defence which includes hardening of devices that an attacker may attempt to breach. PTP recommend that build reviews are conducted to evaluate the security of end devices and servers within a network. This is particularly important for large deployments, such as laptops used by a mobile workforce. Ensuring that a base build is well-secured can avoid costly remediation after systems are deployed.

Areas of Testing

Each system in scope will be examined in depth, and may include:

- Operating system patch levels
- Installed software and services
- Hardening and operating system configuration
- Stolen device protection
- Endpoint protection efficacy
- Traffic analysis for mobile devices to ensure adequate protection

Common vulnerabilities can include:

- Out-of-date software
- Misconfigured services
- Unknown or excessive services running
- Weak or default credentials
- Unused or vulnerable vendor installed “bloatware”
- Missing operating system patches

Example Attacks from Previous Engagements

The following are issues that have been found during build reviews:

- The PTP consultant identified a service installed on the system that ran as NT AUTHORITY\SYSTEM. This service was out of date and was vulnerable to a local privilege escalation vulnerability. This would allow a normal user account to completely compromise the system. The device was protected by an Endpoint Detection and Response (EDR) system that made exploitation difficult. The consultant leveraged knowledge gained during previous engagements to write a custom loader, allowing them to bypass the EDR and completely compromise the system. The remediation advice included removing the service and altering the configuration of the EDR to render the attack ineffective.

- A client had developed a specifically hardened laptop build for staff travelling to high-risk regions of the world. This was designed to have limited access to corporate resources and minimise the amount of data stored at-rest on the machine. To ensure that all traffic over the network was properly secured, interception and tampering was attempted. An application designed to view sensitive documents normally

used TLS encrypted traffic, but by tampering with the traffic, this could be forced to fall back to plaintext communication, leaking the contents of the documents. The software vendor was informed, and an update resolved the issue.

Testing Types

Different types of systems can be tested. The threat model under which they operate, and the level of assurance required, can vary depending on their use case.

Server Testing: For example, web servers, email servers, and database servers. All provide critical business functionality. Often, sensitive data is stored on servers that an attacker is trying to access.

Laptop Testing: Laptops are the most common devices used day-to-day by staff. Due to the portable nature of laptops, they are at high risk of physical attacks, such as theft. Therefore, these devices should be hardened to a higher standard to minimise the risk in the event of a compromised physical device.

Desktop Testing: Very similar to laptop testing, although the risk of device theft is generally reduced.

Kiosk Testing: Some machines are design to be left unlocked in physically exposed locations. With these, it is crucial that breakout to the underlying operating system is prevented and, if breakout is achieved, the impact is limited.

Testing Process

The following steps are taken during the testing process:

Setup: The consultant will access the device to be reviewed. They may use remote access, for example, via RDP or SSH. Alternatively, they may gain physical access, for example, by attending the client site or taking delivery of the device.

Access will be required as both a standard user and administrator level user.

Automated Scans: The first phase of testing consists of automated scans being conducted using tools such as Nessus. This allows the consultant to enumerate the patch level, services install, and configuration of the device. Additionally, this allows testing of the device against CIS benchmarks. Custom PTP PowerShell scripts will also be run, which interrogates the system for misconfigurations which could aid an attacker in achieving privilege escalation.

Manual Testing: Upon completion of the automated scans, the consultant will verify the results using manual techniques, removing any false positives identified.

Additionally, further testing will be conducted, such as attempting to exploit vulnerable software, brute force valid credentials, or review software not scanned by automated tools. At this point, the consultant may also test controls that protect against a stolen device, such as effective full disk encryption.

During this phase of testing, a specific focus will be placed on identifying issues that may allow remote compromise of the device, or privilege escalation from a low-privileged account to administrative access.

Reporting

A report is written detailing the processes carried out and the issues found. Generally, this will contain a prioritised list of vulnerabilities discovered grouped by functional area. Remediation advice will be provided, both in terms of an immediate fix and any defence-in-depth measures that could be taken to mitigate risk. Attack-chains, alongside their impact, will be documented. Any higher-level findings that can be abstracted from the testing will be provided. Any architectural or design weaknesses will be highlighted so that these can be avoided in the future. Finally, an executive summary is produced to allow the most severe issues to be communicated quickly to stakeholders.

CHECK Testing

In some cases, testing may need to be conducted under the NCSC CHECK scheme. When this occurs, certain requirements must be met:

- Testing led by a CHECK Team Leader – Infrastructure (CTL INF)
- On large or complicated tests, the CTL may be supported by one or more CHECK Team Members (CTM)

It is important to consider whether testing will need to be conducted under the CHECK scheme at the scoping phase.

Considerations for SECRET Networks

Where testing will be conducted on a network rated for SECRET information, additional requirements must be met:

- Two CTL INFs are required at all times.
- No storage devices may leave the facility upon testing being completed, requiring the use of new drives.
- All testing and reporting must be conducted on site.

Requirements

For all build review tests, the following prerequisites are required:

- Standard user accounts for all devices in scope
- Administrator level accounts for all devices in scope
- The ability to run custom PowerShell scripts and executables on the devices. This might also necessitate changes being made to the EDR solution.

Access

The consultants will require access to the devices in scope for testing. This can be facilitated in a number of ways. For example:

- Remote: The consultant is given remote access to the system, such as via RDP on a VPN.
- Remote Physical Device: The device to be tested is shipped to the consultant to conduct testing.
- Onsite: The consultant attends the client site and connects to the machine via the local network or by physically interacting with the machine.

5.3. Cloud Breach Assessment Methodology

The PTP cloud breach assessment is a real-world engagement, to ascertain the impact a compromised user account or resource, would have on a company. The interaction is scenario-based and concentrates on role-based access controls, deployed cloud resources, and supporting infrastructure. For each scenario, the PTP consultant will act as a genuine threat actor to find potential attack paths and develop recommendations, to lessen the impact of any compromise. This builds upon your traditional testing and benchmarking engagements.

Authentication

Authentication is mandatory for this service, as user or service accounts are required to fulfil the agreed scenarios, and to identify targetable resources. The consultant will attempt to bypass applicable conditional access policies when encountered by modifying their browser's user-agent or attempting to spoof measured metrics.

When testing from the perspective of a compromised service such as a virtual machine or container, access to the resource should be given in a manner that allows interaction with the operating system (such as SSH or RDP). This allows the consultant to emulate a breach in the environment.

When testing code or image repositories, personal access tokens may be a more appropriate method of authentication as this type of credential is more likely to be leaked or stolen due to their simplicity.

Reconnaissance and Enumeration

Once access is achieved, the consultant will begin the reconnaissance and enumeration stage where they will identity (but not limited to) the following:

- Accessible cloud and code resources.
- Accessible automated processes and servers
- Utilised services and licenses.
- Trawl files using cloud resources such as SharePoint, Delve and any other file sharing SaaS applications.
- Third-party app integrations.
- Infrastructure hierarchy and hybrid connectivity.

This is achieved by using a mixture of bespoke and open-source security capabilities, manual investigations using various cloud portal web applications and services, and by leveraging SDKs and APIs of the cloud and repository providers.

Exploitation

The consultant will utilise accessible cloud and repository resources and services to achieve the following:

- Unintentional and/or unauthorised access to data and services.
- Unauthorised data modification.
- Data modification to gain a significant foothold and/or to escalate their privileges.
- Gain elevated access via RBAC abuse or by compromising additional user or service accounts.

Limitations

Unfortunately, this methodology cannot be exactly and accurately documented due to the complex and ever-changing nature of cloud environments and services. With that said, all exploitation will be against the client's resources and not backbone cloud infrastructure. Additionally, exploitation is usually a form of service abuse rather than a traditional exploit.

Due to these limitations, an engagement will involve defining specific scenarios with the client that accurately reflect the risks to their own environment and therefore is highly specific to each engagement.

5.4. Cloud Configuration Review and Testing Methodology

Cloud Systems

Cloud computing is the on-demand delivery of computer power, applications, databases, storage, and other IT resources, with a pay-as-you-go pricing model. Cloud computing such as Microsoft Azure, Amazon Web Services, and Google Cloud Platform is becoming a common solution for many enterprises looking to move away from physical servers in data centres. This reduces costs and increases survivability in the event of a catastrophic event. However, with new technologies comes new security requirements, along with familiar security principles. Reviewing both and providing usable, secure recommendations ensures data in the cloud is kept secure.

The Key Premises of Cloud Computing:

- Decentralised
- Rapid provisioning
- Remote access
- Minimum hardware management
- Reduced IT hardware upfront costs
- Flexible and scalable
- Shared pool of configurable system resources

The Types of Cloud Systems:

- **Public:** Publicly accessible over the Internet
- **Private:** Accessible to only a specific set of people or organizations
- **Community:** Accessible to groups of organizations and individuals with similar interest
- **Hybrid:** Combination of the above models

The Types of Cloud Service Models:

- **Infrastructure as a Service (IaaS)** is a form of cloud computing that provides virtualized computing resources over the Internet.
- **Platform as a Service (PaaS)** is a cloud computing model in which a third-party provider delivers hardware and software tools, usually those needed for application development. A PaaS provider also typically hosts the hardware and software on their own infrastructure.
- **Function as a Service (FaaS)** is a category of cloud computing services that provide a platform allowing clients to, develop, run, and manage application functionalities, without the complexity of building and maintaining the infrastructure.
- **Software as a Service (SaaS)** is a method of software delivery and licensing whereby software is accessed online via a subscription, rather than bought and installed on individual computers.

Cloud Shared Responsibility Model

Cloud security is a shared responsibility model. While cloud providers offer a secure foundation, users must take active steps to secure their data, applications, and configurations within the cloud environment. Collaboration between the cloud provider and the user is essential for creating a comprehensive and effective security strategy. Our goal is to assist you in strengthening the “client” security responsibilities from your client side.

Cloud Provider Responsibilities:

Physical Infrastructure Security: Cloud providers are responsible for securing the physical infrastructure, including data centers, servers, networking hardware, and other hardware components. This involves measures like access controls, surveillance, and environmental protection.

Hypervisor Security: The hypervisor, which manages the virtual machines and their resources, is the responsibility of the cloud provider. It's important to ensure that virtual machines are isolated from each other and that the hypervisor itself is secure.

Network Security: Cloud providers manage the underlying network infrastructure, including firewalls, load balancers, and routing. They are responsible for protecting the network against external threats and providing network isolation between different clients.

Data Center Security: Physical security measures such as access controls, surveillance, and disaster recovery plans are the responsibility of the cloud provider. This ensures the availability and integrity of the data centers.

Compliance and Certifications: Cloud providers often obtain various compliance certifications to demonstrate their adherence to industry-specific security standards. They ensure that the underlying infrastructure complies with these standards.

User Responsibilities:

Data Security: Users are responsible for securing the data they store in the cloud. This includes encrypting sensitive data, managing access controls, and applying proper data classification.

Identity and Access Management (IAM): Users are responsible for managing user access to their cloud resources. This involves setting up strong authentication mechanisms, defining access policies, and managing roles and permissions.

Application Security: Securing applications and workloads running in the cloud is the responsibility of the user. This includes applying security patches, configuring firewalls, and implementing application-level security controls.

Configuration Management: Users are responsible for properly configuring their cloud resources to align with security best practices. Misconfigured resources are a common cause of security breaches.

Encryption: While cloud providers often offer encryption services, users are responsible for encrypting data at rest and in transit. This ensures that even if a breach occurs, the data remains protected.

Monitoring and Logging: Users are responsible for setting up monitoring and logging for their cloud resources. This helps detect and respond to security incidents in a timely manner.

Security Compliance: Users need to ensure that their usage of cloud services aligns with regulatory requirements and industry standards. Cloud providers may offer compliant infrastructure, but users need to configure their resources accordingly.

Cloud Service Responsibility Matrix

Layer	IaaS	PaaS	SaaS
Identity and Access	Customer	Customer	Customer
Application	Customer	Customer	CSP
Platform	Customer	CSP	CSP
Operating System	Customer	CSP	CSP
Network Configuration	Customer	CSP	CSP
Compute & Storage	CSP	CSP	CSP
Network Services	CSP	CSP	CSP
Data Centres	CSP	CSP	CSP

Figure 1: Cloud Service Responsibility Matrix (CSP = Cloud Service Provider)

The Common Cloud Security Standards:

- ISO/IEC 27017:2015
- MTCS SS 584
- CCM
- NIST 800-53
- Centre for Internet Security Benchmarks (CIS)

The General Security Risks of Cloud Systems and How They Are Assessed:

- Auditing to ensure effective governance, risk and compliance processes exist
- Auditing operational and business processes
- Auditing the management of people, roles and identities
- Functional testing to ensure proper protection of data and information
- Functional testing to validate the enforcement of privacy policies
- Functional testing to assess the security provisions for cloud applications
- Functional testing to ensure cloud networks, data and connections are secure

Testing Cloud Systems

When penetration testing cloud environments, there are some different perspectives the assessment could consider. Some are very similar to external infrastructure and web application assessments; however, they can also be vastly different.

- **Testing outside the Cloud:** Traditional testing of systems which are simply hosted within a cloud environment. For example, this can be virtualised systems that have been moved from on-premises to the cloud, requiring only an external infrastructure penetration test. This could also be a web application which is hosted on the Cloud; however, the infrastructure is out of scope, resulting in a requirement for standard web application testing only.
- **Testing inside the Cloud:** Testing systems within the Cloud could include an authenticated build review of a webserver, requiring authenticated access into the cloud. This could typically be a review of systems which are hosted on the cloud but have firewall rules preventing direct access (only accessible through a bastion host or a private VPC etc.).
- **Testing the Cloud Console:** Testing the configuration of the Cloud console (sometimes referred to as the Portal) would generally be a full Cloud security assessment and could involve reviewing user accounts which have

been set up, their permissions, the access-control lists which have been configured, alongside the cloud providers platform services.

Cloud Testing Limitations

Amazon AWS clients are welcome to carry out security assessments or penetration tests against AWS infrastructure they provision, without prior approval, for the following example services:

- Amazon EC2 instances, NAT Gateways, and Elastic Load Balancers
- Amazon RDS
- Amazon CloudFront
- Amazon Aurora
- Amazon API Gateways
- AWS Lambda and Lambda Edge functions
- Amazon Lightsail resources
- Amazon Elastic Beanstalk environments

However, there are currently several prohibited activities which are not permitted to be performed in an AWS environment. These generally refer back to the "cloud service responsibility matrix". Examples of these include:

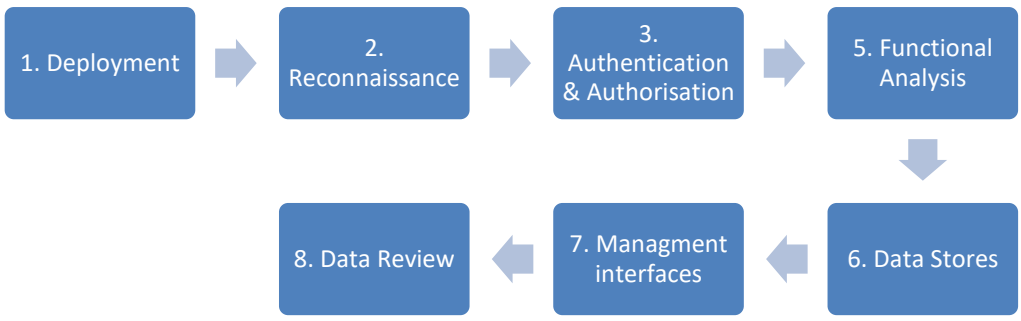
- DNS zone walking via Amazon Route 53 Hosted Zones
- Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), Simulated DoS, Simulated DDoS
- Port flooding, Protocol flooding, Request flooding (login request flooding, API request flooding)

More information can be found here: https://aws.amazon.com/security/penetration-testing/

Microsoft Azure has a similar level of restrictions, and more information can be found here: https://www.microsoft.com/en-us/msrc/pentest-rules-of-engagement?rtc=1

Stages of Cloud Testing

Testing a cloud implementation can generally involve the following stages:



1. **Cloud Deployment:** This phase generally involves carrying out a platform infrastructure test against the underlying cloud platform. This ensures a level of assurance can be gained that the services, applications, and virtual computers are hosted within a secure Cloud network infrastructure.

   Testing will be carried out to ensure that there are minimal ports accessible, and services and content are securely configured. Identified services will be tested for known vulnerabilities and web services will be tested for typical web application security flaws.

2. **Cloud Platform Reconnaissance:** The aim of the reconnaissance phase is to build a complete picture of the technical components and services provisioned by the client that make up the Cloud platform environment.

   In addition to reviewing typical services presented by the client's Cloud environment, reconnaissance and review of external code repositories can be conducted (GitHub etc.) which has been seen in the past to contain access keys and passwords etc., within unauthenticated publicly accessible repositories.

3. **Authentication and Authorisation:** The authentication and authorisation models for the Cloud environments will be reviewed, identifying that users and groups are fit for purpose without security weaknesses that could be utilised to escalate privileges or gain access to unauthorised features. This will include validating key rotation and management, password policy management and reviews of defined groups and users.

4. **Functional Analysis:** Each individual functional component of the cloud platform is examined during this phase of testing. These components could include multiple solutions such as Lambda functions, load balancers, account management, and storage solutions. The platform will be tested to ensure that authorised actions can only be carried out in the predefined order and within the restrictions imposed by the access model defined in the previous section.

5. **Data Stores:** Data stores will be reviewed to ensure that data is only accessible by the defined users and groups. Testing will also be performed to ensure that storage data is not publicly accessible and encrypted where required.

6. **Management Interfaces:** Remote management is often a requirement to enable the on-going support and maintenance of the cloud environment. Unauthorised access to such an interface could lead to the compromise of the entire environment.

7. **Data review:** A review will be performed to validate that data is not publicly accessible or exposed within the cloud environment. This typically includes a review of the following:
   - Exposure of clear text credentials
   - Exposure of access keys
   - Exposure of access tokens

General Access Requirements

Console Access and Configuration Reviews

For Amazon AWS testing, the following is required:

- API and console access
- Secret Keys and passwords for the account
- Ideally, an administrative account would be provided to ensure full coverages but, at a minimum, the following AWS Managed Policies could be attached to the principal in order to grant the necessary permissions:
  - ReadOnlyAccess
  - SecurityAudit

For Azure testing:

- Usernames, passwords, and MFA details. Often this takes the form of an Azure Active Directory account.
- Preferably the user is an administrator in the cloud estate but, as a minimum, the Reader and Security Reader roles are assigned to the user in all subscriptions in scope.

No changes to the cloud configuration would be made during the test; however, administrative access would allow the consultant to review all areas of the configuration.

Infrastructure Penetration Test and Build Reviews
For infrastructure penetration tests and general server build reviews, ideally an instance of the Kali Linux image and a Windows image is created within the test environment, with network access to the VPCs and Instances being assessed.

Inbound SSH to the Kali image and Remote Desktop Service to the Windows image is required from PTP's IP address range. These images will be utilised as a testing platform for PTP.

Outbound Internet access is also required for installation and licensing of tools from both the Kali and Windows images.

Presentation of Services

There are, in general, two types of use for cloud-based computing:

- Internet accessible. For example, where an ecommerce web application is served from the cloud to the Internet.
- Private access. For example, an internal server farm could be located in the cloud that can only be accessed via a Virtual Private Network (VPN).

Ensuring the configuration is correct is critical in preventing unauthorised access to data or services.

Cloud Management Interface
Cloud server instances are provisioned and managed through the management interface. It is critical that this interface is secure, thus preventing an attacker from accessing the interface and being able to take control of the environment. The following factors are reviewed:

- User segregation and principle of least privilege. For example, if a user only needs to monitor the environment, then they should only be given the minimum amount of privilege in order to do so.
- Use of Two-Factor Authentication (2FA) to ensure access to the management interface is secure and strictly controlled.

Cloud Virtual Network
The Cloud Virtual Network configuration allows multiple cloud server instances to be interconnected. For example, a web application environment could have frontend webservers, middleware servers, and backend databases. All of these could be interconnected to provide an application stack that gives a seamless experience to the user and to the application's developers.

A review of the network configuration, including such details as on-host firewalls, segregation testing, and routing is carried out to ensure all data is secure and cannot be tampered with by an attacker.
Network access control is implemented using Network Security Groups (NSG). A thorough review of the NSG is carried out to ensure it is configured.

The following best practices are reviewed:

- **New Rules:** All new rules should have minimal access rights, source/destination, and service.
- **Compliance:** All new firewall rules should adhere to all compliance policies enforced by the company.
- **Change control:** Ensure that a process is defined for changing network access control rules. This should ensure rule approval is required, request reasons, documentation and back out procedures defined.
- Rules should be removed when services decommissioned to ensure connectivity to unintended systems are not allowed.

Cloud Virtual Machines

The cloud is designed to present a seamless view of a server within the cloud environment. As with any other server, the operating system should be reviewed to ensure that it is secure and fit for purpose. The following provides a high-level overview of the areas investigated during a configuration review of the operating system:

- Currently installed patches and patch management policy. This is for the operating system and any installed third-party software.
- A review of the operating system configuration. An in-depth look at how the operating system is configured and recommendations on how to improve the security.

Configuration reviews are carried out using the latest applicable vendor guides and years of experience reviewing and pen testing multiple operating systems.

Cloud Identity Management

The ability to control who has access ensures that data stays secure. We assess the identity management configuration to provide assurance that the configuration of the users is as expected.

The following is a brief list of the factors we cover:

- Centralize identity management
- Enable Single Sign-On (SSO)
- Deploy password management
- Enforce Multi-Factor Authentication (MFA) for users
- Use Role-Based Access Control (RBAC)
- Control locations where resources are created using resource manager
- Guide developers to leverage identity capabilities for SaaS apps
- Actively monitor for suspicious activities

5.5. Code Review Methodology

Source Code Review

This class of testing is aimed at reviewing an application's source code to ensure that it considers security in its critical paths. This includes several steps such as tracing the path of user-controlled input through the application

and verifying it is correctly handled, checking that any encryption is performed appropriately, ensuring authorisation is effective and looking for memory manipulation (buffer under- and overflows, heap overflows, integer overflows, format string manipulation).

This is important particularly where such information is used in privileged operations, for example, whether OS command injection or buffer overflows are possible to trigger from a client perspective.

Common vulnerabilities can include:

Input Validation:
The correct use of input validation is particularly important in applications written with older frameworks, such as C, C++, VB and PHP; as these tend to have fewer built-in safeguards against such problems as SQL injection or OS command injection. Java and .NET based applications tend to exhibit fewer of these sorts of issues, partly because of superior library support for safe ways of carrying out such operations. However, robust input validation is required in any language.

Timing Attacks and Buffer Overflows
Information disclosure issues will also be checked for, to ensure that no data is accidentally leaked via timing attacks; such as where a password is rejected in different amounts of time, depending on how many initial characters are correct. Checking for leaked data in the slack space of packets or unallocated data will also be checked for.

Race Conditions: Where applications are multi-threaded, it is important to consider race conditions, where two threads may be contending for the same resource – if this is not handled appropriately, security vulnerabilities can result. An example might be where an application checks a file's permissions and then writes to the file. If an attacker can swap the file for another one in between the check and the operation, they may be able to cause unintended side effects.

Logic Errors: Another class of errors is semantic, or application logic errors, for example, where a low memory condition might cause certain checks to abort and leave the application in an inconsistent state. A further example would be failing to check for negative numbers properly when performing a financial transaction, both in a classic sense (-1) and also for integer overflow/underflow conditions; as the maximum representable integer, plus one, can often end up as being a negative number. Some special cases can also cause a buffer to be freed twice if the particular logic path has not been examined fully – this can lead to arbitrary code execution in certain circumstances.

Cryptography: The use of cryptography will also be examined to check for errors, such as padding oracles where too much information is leaked back to an attacker revealing whether a particular packet is correctly or incorrectly constructed. All encryption should make use of an HMAC in addition to the encryption itself to prevent chosen ciphertext attacks.

The correct use of Pseudo Random Number Generators (PRNGs) is also extremely important to make sure they are properly seeded and of a sufficient quality for cryptographic applications. Where Initialisation Vectors are used in CBC ciphers, these must also be treated carefully so they are not repeated. Electronic Code Book mode ciphers are generally suboptimal, unless being used to encrypt data with extremely high entropy such as other encryption keys; otherwise, their use may reveal structure in messages by XORing two distinct messages. Any application code

updates must always have their signatures checked to make sure an attacker is not trying to introduce their own code into the application.

Front End Presentation: Web applications will need to be checked for all these issues, plus some web-specific problems such as Cross-Site Request Forgery which arises when a web application cannot easily tell whether a request is client-initiated or has been spoofed by JavaScript. The safest way to reliably prevent this is to include an unpredictable token on each page which is then checked upon submission – which in theory means the attacker cannot spoof such requests.

Approach / Tools

Depending on the language and platform in use, various static code analysis tools may be used to pick up the more obvious problems. These are usually customised towards the language, but include generic SAST tools, such as sonarqube. These sorts of tools can be incorporated into the development lifecycle to cut down on the number of security-related issues. DAST checking will depend on the framework and language in use and will be performed during the assessment.

Manual assessment of the code will also be performed, this will be mainly on the critical paths of the code and concentrate on security critical areas of the application. All PTP code reviewers have worked in development and have experience with both programming in general and specialisations in specific languages.

The reverse engineering of existing libraries through the use of decompilers and disassemblers may also be performed, using tools such as Ghidra, IDA, jadx and dnspy. This is generally suboptimal for assessment and source code is more efficient.

Device Independent

Previous Code review's carried out have included embedded C/C++ devices, iPhone and Android applications and web applications written in a variety of languages, including PHP, JavaScript, Java and C#.

Native application reviews have been carried out for applications written in languages from R to Delphi to Python to Scala.

Code Assisted Testing

An effective alternative to a pure code review is to assess and application with access to the code at the same time. This allows for a concentration of effort for both the application and code review. This process with normally include running SAST on the code and manual investigation of hot points within the code whilst using the application to understand how this could be impacted.

This can reduce the number of theoretical vulnerabilities in both the application and code review tests.

5.6. Compiled Applications Methodology

What should be included in Compiled Application Test?

The functionality of traditional compiled applications, including their local functions, dependencies, and interactions with other systems. With applications that make use of multiple processes, the inter-process communication methods should be analysed for opportunities to manipulate behaviour.

Areas of Testing

- Access Controls
- Cryptographic Failures and Data exposure
- Design Flaws in application functionality
- Software and Data Integrity (e.g third-party dependencies and software supply chain)

Common Vulnerabilities

- DLL Hijacking
- Privilege Escalation
- Hardcoded encryption material
- Insecure Inter-Process Communication (IPC)
- Credentials stored in configuration files

Testing Approach

Testing Scenarios:

Blackbox Testing: The consultant will be given access to the application in the same form as a legitimate user; this may include user manuals, training material, and other documentation, however, it will not include the application source code. The consultant will then attempt to identify and exploit any security weaknesses in the application via reverse engineering techniques.

Whitebox Testing: The consultant will be given access to the application, alongside the applications source code. The consultant will then conduct testing using the source code as a reference to allow faster identification of issues and identification of more complex security issues.

Setup: The consultants will be provided with access to the application, either by being provided with a valid installer, or access to a device with the application installed. Where the consultant is provided with access to a device, they should be granted local administrator privileges to allow installation of reverse engineering tools.

Reverse Engineering: The consultant will begin to reverse engineer the application in order to understand how it functions and where weaknesses may lie. This process includes identification of communication protocols used, such as Inter Process Communication, or network-based protocols. This process consists of both static and dynamic analysis techniques that will allow the consultant to identify security issues with the application.

Unexpected Use Cases: In addition, there is an expectation from developers that users will use the application in the designed way, which cannot be guaranteed and is certainly not the case for anyone attempting to attack the application or the service it provides. In some cases, access controls could be applied on the frontend but not on the server side, which means that certain tasks could be issued bypassing the access control layer if the task itself was not executed within the application's frontend as expected.

Exploitation: Upon identification of potential security issues, the consultant will attempt to develop proof-of-concept exploit for the application. These will allow the consultant to prove the existence of the vulnerability and demonstrate the impact it may have.

Reporting

A report is written detailing the processes carried out and the issues found. Generally, this will contain a prioritised list of vulnerabilities discovered grouped by functional area. Remediation advice will be provided, both in terms of an immediate fix and any defence-in-depth measures that could be taken to mitigate risk. Attack-chains, alongside their impact, will be documented. Any higher-level findings that can be abstracted from the testing will be provided. Any architectural or design weaknesses will be highlighted so that these can be avoided in the future. Finally, an executive summary is produced to allow the most severe issues to be communicated quickly to stakeholders.

CHECK Testing

In some cases testing may need to be conducted under the NCSC CHECK scheme. When this occurs, certain requirements must be met.

- Testing led by a CHECK Team Leader - Applications (CTL APP)
- On large or complicated tests, the CTL may be supported by one or more CHECK Team Members

It is important to consider whether testing will need to be conducted under the CHECK scheme at the scoping phase.

Considerations for SECRET Networks

Where testing will be conducted on a network rated for SECRET information, additional requirements must be met:

- Two CTL INFs are needed at all times.
- Additional expenses must be accounted for, as no storage material may leave the facility upon testing being completed.
- All testing and reporting must be conducted on-site.

Example Attacks from Previous Engagements

To illustrate the type of real-world attack, these are some possible attacks based on previous tests carried out across a range of networks.

The consultant identified that the application would attempt to load multiple DLLs that did not exist. The process was executed under the context of a privileged user, with the search order for these DLLs including a folder controlled by the user. Therefore, by creating a malicious version of a missing DLL and placing it within the user-controlled folder, the consultant was able to gain code execution as the privileged user, allowing local privilege escalation on the system.

Within the same application, it was found there were hard-coded encryption keys used to decode settings from a configuration file. By using this key, the consultant was able to decrypt the configuration file and identified administrator level credentials for remote systems stored within them. This would have allowed the consultant to gain access to remote systems via a user-level terminal.

Finally, the application made use of multiple processes linked via IPC. Weaknesses were identified in this IPC process that allowed the consultant to inject commands, bypassing an authentication requirement that would normally be required by the application. This allowed the consultant to conduct privileged operations via the application without knowledge of the pre-requisite password.

Pre-requisites and Requirements

General

- Any documentation available for the application
- Application source code in the case of whitebox testing

Access

The consultant will require access to the application to be tested via one of the following methods:

- Copies of the installation files to setup a local installation for testing
- Access to a system with the application installed
    - Alongside Administrator rights to the system to allow installation of tools

Inductions and Training

In some situations, additional inductions or training may be required before testing can occur; this is most common with onsite work. Where this occurs, additional requirements exist:

- Any inductions or training required to go onsite will need to be provided and accounted for in the testing time.
- Any PPE required should be stated.

5.7. External Infrastructure Methodology

External or public-facing infrastructure is defined as all the servers and services that are reachable from the Internet. Network infrastructure covers the services offered at an operating system level but would not, for example, include web applications.

The internet-facing infrastructure includes:

- One or more firewalls that provide protection against Internet-borne threats and are used to restrict access
- Servers that provide various services, such as web servers, email servers, and so forth

These are generally considered to be the most 'at risk' from an attacker or malware, as it is near impossible to restrict access to the hacker while granting access to the genuine prospective client requiring your services.

As a rule, the more functionality a server or network delivers, the more likely it is to be attacked. As functionality increases, the opportunity for misconfiguration and vulnerability increases. Hence a web site running a complex transactional web application is far more likely to be vulnerable to security flaws than your upstream router. The core of any penetration test should include your public infrastructure, but do not forget that there are other routes into your network.

Areas of Testing

- **Information Gathering:** This is a passive activity that seeks to discover information about the target.
- **Host Discovery and Identification:** Determining what hosts are visible and what operating systems they are running. An attacker would do this in order to better target an attack.
- **Port and Service Scanning:** This involves actively testing the hosts to determine what ports are open and what services they are offering. For example, a web server would likely present 80/TCP and 443/TCP upon which a web application is running.

- **Vulnerability Analysis:** This phase of testing determines what, if any, vulnerabilities are present on the targets. Using automated scanners to assess the targets quickly and accurately, a picture is built up of the target's security posture.
- **Extended Manual Testing:** Once automated scans have been completed, each discovered issue is manually tested to verify whether or not it exists. This phase may, if appropriate, also include exploiting any issues to see how far an attacker would be able to get.

Vulnerability and extended manual testing will explicitly identify where security holes lay and remove false positives. When applicable, other methods will be utilised.

Testing

PTP will assess and thoroughly test all internet-facing servers within the scope. Any security issues identified will be highlighted, and appropriate recommendations will be made to ensure all risks are appropriately minimised. Any high-risk issues, or ones that could be exploited easily, will be reported immediately as they could pose an imminent threat were they to be discovered by an attacker.

Testing is an iterative process. Each step feeds information into the next to identify and exploit targets. External infrastructure testing will usually use (but not restricted) the following testing procedures.
Information Gathering: WHOIS databases, e.g. Ripe or InterNIC, are used to confirm the IP addresses belong to or are likely to belong to the client; we assess the information contained therein to determine if it could be used in a social engineering attack.

A selection of other open-source resources such as Google, Google Groups, and Netcraft will be used to acquire additional information relating to the client. Any information about the client that could be useful to an attacker will be analysed and reported.

Services such as DNS on the in-scope IP range will be analysed to determine if they reveal further information regarding the external infrastructure. For example, if a zone transfer is possible, it could reveal significant information about internal and external targets. This is useful in being able to map out the external infrastructure.

Port and Service Scan: A full port scan of all 65535 TCP ports of each host in scope will be performed in order to determine what services are being presented to the Internet. For UDP ports, a common (top 1000) port scan will be performed initially, with further investigation as time allows. This also helps to determine whether or not any external firewalls are correctly configured. Numerous times we have discovered that services are being offered that the client was unaware of. This could lead to an attacker being able to exploit the services.

Host Discovery and Identification: Using a combination of port scan results, and other protocol and packet fingerprinting techniques, additional hosts can be identified, and further information learnt such as the operating system and platform utilised.

Vulnerability Analysis: We use a combination of commercial and open-source vulnerability scanners. These are chosen for their accuracy and repeatability. Such tools have been honed over years of use by the security community so that they can accurately assess vulnerabilities.

Unless specifically requested, we do not carry out Denial-of-Service (DoS) or distributed Denial-of-Service (DDoS) attacks.

Extended Manual Tests: All results found by toolset use, whether false positive or not, will be manually explored to ensure that they are either present or are a false positive from the vulnerability scanner.

Any services that require authentication are checked manually for default usernames and passwords combinations. Brute force attempts are only attempted if account lockouts will not occur.

Connections will be made to all open TCP services to check how these services react and for banner information such as host types and version numbers.

Version numbers of targets services are checked against recent exploits that are unlikely to be addressed in vulnerability scanners.

Exploits are downloaded, compiled, and executed where appropriate (after receiving the client's permission) to confirm the level of risk. Every effort will be made by us to ensure the client is provided with a full representation of their externally facing network, and its associated risks. Any issues identified will have the appropriate recommended actions to be taken.

Reporting

A report is written detailing the processes carried out, and the issues found. Generally, this will contain a prioritised list of vulnerabilities discovered grouped by functional area. Remediation advice will be provided, both in terms of an immediate fix and any defence-in-depth measures that could be taken to mitigate risk. Attack-chains, alongside their impact, will be documented. Any higher-level findings that can be abstracted from the testing will be provided. Any architectural or design weaknesses will be highlighted so that these can be avoided in the future. Finally, an executive summary is produced to allow the most severe issues to be communicated quickly to stakeholders.

CHECK Testing

In some cases, testing may need to be conducted under the NCSC CHECK scheme. When this occurs, certain requirements must be met:

- Testing led by a CHECK Team Leader – Infrastructure (CTL INF)
- On large or complicated tests, the CTL may be supported by one or more CHECK Team Members (CTM).

It is important to consider whether testing will need to be conducted under the CHECK scheme at the scoping phase.

Requirements and Pre-Requisites

The following is required for external infrastructure tests:

- A list of IP address (ranges) in scope
- Additional authorisation if the infrastructure is hosted by a third party

5.8. Firewall/Network Device Review Methodology

Introduction

A firewall has a very specific role on the network. It is responsible for controlling the flow of information between devices, and more specifically preventing devices from communicating with other devices or services, unless specifically required. This is essential for Internet-facing devices, however, is also important internally.

A poorly defined ruleset may allow excessive access to devices or services which may increase the risk of compromise. As a rule, a minimum number of hosts and services should be allowed to communicate with each other to reduce the risk of attack.

If the configuration of the firewall itself is incorrect, then a hacker may be able to exploit the firewall from a network perspective. This could be used to launch a denial-of-service attack against the network or even alter the rules to allow access to otherwise restricted networks.

Testing Overview

To identify any risks posed by firewalls, PTP can carry out several tests against the firewall. A firewall ruleset review is always recommended. In this case, a copy of the current ruleset would have to be made available. Ideally this is exported from the device and a copy is given to the consultant, however, in restrictive environments this can be carried out from the firewall's administrative interface if necessary.

A network diagram is also desirable so that the consultant can understand the intent of the firewall rules. The ruleset is then manually assessed to identify any overly permissive, suspicious, or unexplainable rules.

A network-based assessment of the device itself can be carried out as per a standard infrastructure test. This would involve the consultant scanning the device from an adjacent network to see if any insecure services are running, if any default credentials are in use, or any other network-based issues are present that could allow an attacker to compromise the device itself.

Testing Breakdown

The configuration of all devices is also subjected to review against industry best practices, such as CIS, NSA and PTP custom security checklists. Firewall reviews give a more in-depth view of device security and can identify several underlying issues including deficiencies in the following:

Patch management weaknesses

An attacker could gain administrative access or cause a denial-of-service attack depending on the functionality available and exploits available in the wild on unpatched devices.

ACL implementation weaknesses

Firewall rules are made up of several component parameters, primarily source address, destination address, and service. Each of these parameters can be single entities or groups of hosts or services. The more restrictive the parameters are, the better the security profile of the protected assets. Conversely, if these parameters are less restrictive, the security afforded to networks being protected falls accordingly. A parameter setting of ANY effectively removes all restrictions against that parameter.

Use of unencrypted management services,

Similarly, to the ACL implementation, checks will be conducted to ensure that no unencrypted services are permitted on the network, such as FTP, Telnet etc. These unencrypted services put transported data at risk on the network.

Device administration configuration weaknesses

The device account and password policy will be reviewed to ensure that unauthorised access cannot be granted to attackers.

Auditing and accounting configuration weaknesses,

Logging provides a mechanism for auditing actions on network devices. At a basic level, it will detail users and timestamps for changes to the configuration of hosts or devices.

All testing will comprise of a variety of manual and automated tests using open source and commercial tools, along with bespoke tools and custom scripts. This low-level approach to testing will be backed up with the use of automated tools such as Nipper.

As the above states, all tests will be manual and automated to a certain level. The length of engagement will determine which aspect the consultant will focus testing on.

Manual approach

This provides the most coverage, however, requires more time to manually liaise with firewall administrators. This in turn provides the consultant with context surrounding the firewall rules required for business purposes. Without the context provided by the firewall administrator, consultants are forced to list all rules that appear to be potentially suspicious or those of which do not follow best security practices.

Automated approach

This approach provides a large coverage of firewall rules and settings as generically insecure or overly permissive rules that do not conform to best security practices are provided within the report. These rules may be functionally required by the business; therefore, firewall administrators will generally review the provided findings at a later date to determine whether they pose a risk to the network.

5.9. Internal Infrastructure Methodology

As a CREST-accredited provider, Pen Test Partners (PTP) can deliver internal infrastructure tests to the highest standards. In addition to CREST we are certified by other bodies to deliver best-in-class cyber security services. These include CHECK and Cyber Essentials Plus.

To keep abreast of new knowledge and techniques, PTP's team of security consultants propagate their skills across the wider team.

The internal network of a business is often key to their operations, containing critical systems, information and functionality. PTP therefore recommended that internal networks are subject to regular security testing to ensure all data and systems are adequately protected against intrusion by malicious attackers.

Areas of Testing

All systems within the specific network ranges are tested from an unauthenticated and, if requested, authenticated, perspective. The following aspects will be examined:

- Complete inventory of networks, with a focus on identifying all live hosts
- Enumeration of services that are hosted on servers
- Discovery of vulnerable or out of date software
- Network segregation

Common vulnerabilities can include:

- Outdated operating systems
- Default or weak credentials
- Vulnerable software due to being out of date or poorly configured
- Active Directory misconfigurations
- Weak or no encryption found on services
- Legacy or previously unknown hosts, protocols, or services presenting unnecessary risk

Example Attacks from Previous Engagements

To illustrate a real-world attack, the following is an example of a domain compromise achieved by leveraging vulnerabilities found on a previous test:

The consultant was provided with access to a laptop that was connected to a corporate VLAN of a network. During initial enumeration, the consultant identified a file server. This server hosted an SMB share that did not require authentication for access. The share was found to contain backups of virtual machines hosted within multiple VLANs of the network. By extracting these backups, the consultant was able to access sensitive information stored within the machines, including extracting passwords from the Windows SYSTEM and SAM hives.

Using a pass-the-hash attack, the consultant was able to access the live servers and extract further information from the domain. This included access as an administrator to the backup server. By extracting the encrypted authentication material used by the backup server, along with its keys, the consultant was able to recover the plaintext password for the default domain administrator, allowing total domain compromise to be achieved.

Additionally, spraying this password along with other hashes identified earlier allowed the consultant to gain access to other machines that were not joined to the Active Directory domain, indicating that password reuse was prevalent within the network.

Testing Scenarios

PTP offers three main scenarios for an internal infrastructure test:

Blackbox Testing: The consultant will receive no assistance from the client beyond that which is required to connect to the network. This testing will emulate a malicious attacker who has gained access to the network but does not have additional access such as valid user accounts. The consultants will attempt to exploit systems to gain further, more privileged access to the network.

Greybox Testing: The consultant is given assistance in accessing the network and provided with additional information, such as a set of valid user credentials. This allows the consultant to simulate a malicious insider, or an attacker who has successfully phished or otherwise gained legitimate credentials.

Whitebox Testing: The consultant is provided full access to all resources they require, including network diagrams, physical access, console access, user account, and administrator access.

Testing Process

The following steps are taken during the testing process:

Setup: The consultants will connect to the network; this is generally achieved either via a VPN, or by the consultants conducting testing from the client site. In most scenarios, the consultants will conduct testing from their own devices. Once connected to the network and initial connectivity tests have been conducted, the consultants can begin further testing.

In some networks, the consultants may need to move between various subnets or VLANs to gain access to in scope systems. In these scenarios, the full testing chain (with the exception of reporting) will be conducted multiple times, once for each network in scope.

Discovery and Enumeration: Consultants will conduct discovery and enumeration against the internal network using a variety of techniques, including host discovery, port scanning, domain and user enumeration, and enumeration of connected networks. This allows the consultants to build an overall view of the network and services that are available within it, allowing them to ensure the full network is tested effectively.

Key and critical systems such as domain controllers, databases, and file stores are assessed as high priority, as these often contain sensitive information or act as “keys to the kingdom” and would be of high interest to malicious attackers.

Vulnerability Assessment: Several vulnerability assessment tools will be used to assess hosts for common weaknesses, including outdated software, weak or default credentials, or poor access controls. The results of these assessments are then verified and, if applicable, taken forward for manual testing, such as exploitation of issues to gain further access to a network.

These scans will be conducted from an unauthenticated perspective external to the devices, and, where required, conducted using provided credentials to allow authenticated scanning.

Manual Testing: Once all hosts and services have been identified and initial vulnerability assessments have been completed, the engagement enters the manual testing phase. During this phase of testing, the consultant will conduct testing against services that have been identified in previous phases, which may include tailor-made attacks and exploitation of services. Such a targeted attack requires time and expertise to determine the most efficient route of compromise, if any.

Particularly important is the identification of improperly configured services, as these can allow an attacker remote access without system owners being aware that any intrusions have taken place.

The client will be made fully aware if privilege escalation is found to be possible on a specific host. This is because further access to the network maybe gained by exploitation of this vulnerability. Any potential impact of a vulnerability will also be communicated at this stage.

Results from the vulnerability scanners will be verified and false positives removed from the final results.

Reporting

A report is written detailing the processes carried out and the issues found. Generally, this will contain a prioritised list of vulnerabilities discovered grouped by functional area. Remediation advice will be provided, both in terms of

an immediate fix and any defence-in-depth measures that could be taken to mitigate risk. Attack-chains, alongside their impact, will be documented. Any higher-level findings that can be abstracted from the testing will be provided. Any architectural or design weaknesses will be highlighted so that these can be avoided in the future. Finally, an executive summary is produced to allow the most severe issues to be communicated quickly to stakeholders.

CHECK Testing

In some cases, testing may need to be conducted under the NCSC CHECK scheme. When this occurs, certain requirements must be met:

- Testing led by a CHECK Team Leader – Infrastructure (CTL INF).
- On large or complicated tests, the CTL may be supported by one or more CHECK Team Members (CTM)

It is important to consider whether testing will need to be conducted under the CHECK scheme at the scoping phase.

Considerations for SECRET Networks

Where testing will be conducted on a network rated for SECRET information, additional requirements must be met:

- Two CTL INFs are required at all times.
- No storage devices may leave the facility upon testing being completed, requiring the use of new drives.
- All testing and reporting must be conducted on-site.

Requirements

The following will be required for internal infrastructure tests:

- Accurate scope of the networks to be tested and an authorisation form
- Information regarding connecting to the network, e.g. if it is Ethernet, Wi-Fi, Fibre, network access control (NAC) etc.
- Any network diagrams available
- Any credentials for authenticated scans
- Any information required regarding the site for onsite work

Access

The consultants will require access to the networks in scope for testing; this can be facilitated in several ways:

- Remote via VPN: Consultants are given access to the client's standard VPN setup.
- Remote via shipped laptop: PTP will ship a laptop to the client that will be attached to the required networks.
- Virtual machine in client network: The client will configure a virtual machine or jump box for the consultants to test from within the network.
- Onsite: The consultant will attend the client site and conduct testing by connecting their own device to the client network.

In all testing scenarios, the client will need to ensure that the consultants' testing device is connected to the relevant networks and VLANs. Remote assistance may be needed to move the device as required if multiple networks or VLANs are to be tested.

5.10. Kubernetes Testing Methodology

Kubernetes, a widely adopted container orchestration system, is integral to managing containerised applications in various environments. Vulnerabilities in Kubernetes clusters can lead to severe security breaches, including unauthorised access, data theft, and service disruption. Regular testing of Kubernetes configurations and deployments is essential to ensure the security and resilience of these systems. It is recommended to perform these tests at least annually and following significant updates or changes to the infrastructure.

The PTP Kubernetes security assessment methodology draws inspiration from established frameworks like the CIS Kubernetes Benchmark and the NSA-CISA Kubernetes Hardening Guidance. However, we understand that relying solely on these resources is insufficient. Our approach blends these standards with our expertise and awareness of the latest security research, forming a dynamic and comprehensive testing methodology.

For thorough testing of Kubernetes clusters, it is crucial to engage with experienced security professionals. Automated tools, while helpful, often fall short in thoroughly evaluating complex Kubernetes environments. They might miss intricate vulnerabilities or generate false positives and negatives. PTPs manual testing, leveraging the insights and experience of our consultants, ensures comprehensive coverage and the discovery of subtle, yet critical security issues.

What is covered in a Kubernetes Security Assessment?

A Kubernetes security assessment should encompass a broad range of checks, including but not limited to:

- **Configuration and Security Posture:** Reviewing Kubernetes cluster configurations for best practices in security, including RBAC (Role-Based Access Control) settings, network policies, and pod security policies.
- **Secrets Management:** Ensuring sensitive data like passwords and tokens are securely managed and encrypted.
- **Workload Analysis:** Examining deployed applications for security risks, including container vulnerabilities and runtime security.
- **Network Security:** Testing network policies and ingress/egress controls to prevent unauthorised access.
- **API Server Security:** Ensuring the API server, the central control entity in Kubernetes, is properly secured against unauthorised access and attacks.
- **Storage Security:** Assessing the security of persistent storage configurations and data encryption.

Common vulnerabilities in Kubernetes environments can include:

- Misconfigured RBAC leading to excessive permissions.
- Inadequate network policies leading to potential data breaches.
- Poor secret management practices exposing sensitive data.
- Use of outdated or vulnerable container images.

High and Critical Risk Vulnerabilities

In line with our policy, any discovery of high or critical risk vulnerabilities, or those posing immediate threats, will prompt direct contact with the designated client representative. This communication, through email or phone, will detail the nature of the vulnerability, its potential exploitation, and remediation steps.

Example Attacks from Previous Engagements

Examples of real-world attacks based on our past Kubernetes security assessments include:

- A pod hosting an exposed service had the ability to communicate with a cloud metadata service, resulting in high level privileges being gained.
- Secrets were stored in environment variables that were available to any process on a container.
- An untrusted third-party container was deployed into a sensitive namespace, potentially allowing a supply-chain attack.

Testing Process

The testing process for Kubernetes security assessment involves several phases:

Review and Analysis: Understanding the cluster architecture, configurations, and deployed workloads. Examining Kubernetes manifests, Helm charts, and any relevant documentation.

Reconnaissance: Identifying the Kubernetes version, API endpoints, and any exposed services. This phase includes both internal and external reconnaissance.

Enumeration and Vulnerability Discovery: Assessing each component of the cluster for misconfigurations, vulnerabilities, and security best practices. This involves testing the security of the control plane, worker nodes, and deployed workloads.

Automated Scanning and Manual Testing: Using specialised tools to identify common vulnerabilities, followed by in-depth manual testing to uncover deeper security issues.

Attack Simulation: Crafting realistic attack scenarios based on identified vulnerabilities. This may involve simulating privilege escalation, unauthorised access, or data exfiltration.

Reporting

A report is written detailing the processes carried out and the issues found. Generally, this will contain a prioritised list of vulnerabilities discovered grouped by functional area. Remediation advice will be provided, both in terms of an immediate fix and any defence-in-depth measures that could be taken to mitigate risk. Attack chains, alongside their impact, will be documented. Any higher-level findings that can be abstracted from the testing will be provided. Any architectural or design weaknesses will be highlighted so that these can be avoided in the future. Finally, an executive summary is produced to allow the most severe issues to be communicated quickly to stakeholders.

Requirements and Prerequisites

From the client, we require:

- Access to the Kubernetes API to connect to the service.
- A valid method of authentication such as a pre-generated kubeconfig file, or credentials with which to generate one.
- Read only access to all resources in the cluster, including custom resources.
- Ideally, “exec” and “port-forward” permissions on the pods API endpoint to spawn shells on containers and forward internal services.

5.11. Mobile Application Security Assessment Methodology

Our mobile application testing methodology is in line with the OWASP Mobile Application Security project which provides guidelines for mobile application assessments via the OWASP Mobile Application Security Verification Standard (MASVS) and OWASP Mobile Application Security Testing Guide (MASTG).

As mobile applications continue to gain relevance, it is increasingly important to security test them to identify and address security vulnerabilities and weaknesses before they can be exploited by attackers, potentially leading to data breaches, loss of sensitive information, financial loss, and reputational damage. A mobile penetration test can help to uncover vulnerabilities such as insecure data storage, weak authentication and authorization mechanisms, insecure communications, and other security issues that could be exploited by attackers. By identifying and fixing these vulnerabilities, mobile applications can be made more secure and better able to protect users' sensitive information.

The greater areas of mobile application testing should include:

- Authentication and session management
- Network communications
- Code inspection
- WebView implementation
- Tampering and reverse engineering
- Cryptography implementation
- Application configuration
- Data storage analysis

Common vulnerabilities can include:

- Biometric authentication bypass
- Sensitive information in application bundles
- Insecure data storage
- Improper user session handling
- Access control bypass
- Hardcoded cryptographic keys
- Weak authentication mechanism
- Insecure exported activity
- Sensitive information written to device logs

Example Attacks from Previous Engagements

To provide examples of potential attacks, some vulnerabilities exploited in previous tests are described below:

A mobile application had hardcoded details of a cloud service which provided identity and access management. These details included identification keys, which were uncovered by decompiling the application binary. Once acquired, these keys were used to generate temporary access keys which allowed access to part of the client's cloud infrastructure, including databases and storage buckets. This was also made possible due to misconfigurations on their cloud infrastructure, which allowed the creation of temporary access keys.

Poor implementation of biometric verification can often lead to the bypassing of an application's local authentication. With access to the device, it was possible to circumvent the biometric login of the application and access sensitive user's data.

Similar to web applications, mobile applications also communicate with the application server and its backend via the HTTP protocol or cutting-edge technologies like QUIC. Vulnerabilities are often found in these interactions with the server's API. An example of this was a mobile payment system that was tricked into accepting a payment of a certain value and crediting the user's account with more money than it was paid. This attack used interception techniques to tamper with communication by changing the request's parameters.

Testing

The following steps are taken during the testing process:

Architecture, design and threat modelling: Testing starts with an analysis of the mobile applications to gather information on the application functionalities and frameworks used. It is important to consider the nature of the application, as general-use applications have different security requirements than apps that handle highly sensitive data. In this phase, vulnerabilities in the application's components are verified and potential attack paths are identified.

Static analysis and reverse engineering: In this phase, the applications' packages are extracted and examined to identify misconfigurations in transport security, permissions, and exported components. The application binaries are decompiled, and the code is inspected to detect any form of sensitive data such as private keys, passwords or certificates. In addition, code inspection provides further insight into the application's logic, which helps to exploit issues found in subsequent stages of the testing process.

Dynamic analysis and network communications: With the application installed, the consultant proceeds by interacting with the application's functionalities and analysing the network traffic generated by communications between the app and server's API. In this phase, extensive security checks are performed to cover the main areas of the assessment. To mention a few of them, application components such as WebView are inspected to ensure no tampering or code injection is possible, user authentication and authorisation are assessed to uncover vulnerabilities such as authentication bypass, and applications' logs are monitored to ensure no sensitive data is leaked through them.

Data storage analysis: The consultant verifies which data is stored in the local container and external storage, and how it is stored. When the application is required to store sensitive data, this must be done securely by using encryption and security components of the system, such as Android's Keystore and iOS Keychain, which have their

implementations on the mobile apps verified. The use of application's cache and temporary files are also inspected, as they can unintentionally store sensitive information.

Reporting

All the vulnerabilities found during testing are explained in detail in a report document, which contains risk ratings for every issue discovered. Recommendations for resolving or mitigating the identified issues are provided and suggestions are made to harden the overall application's security. The report contains a business risk summary to ensure that the most significant issues are conveyed to relevant parties.

Requirements

Application builds should be provided for each platform in their respective formats: APK for the Android operating system and IPA for iOS. Alternatively, apps could be downloaded from the App Store and Google Play, where they will be extracted from testing devices for further analysis.

Some applications implement root and jailbreak detection, as well as certificate pinning, which are used in mobile application security to prevent attacks such as tampering and reverse engineering. Although such measures can be circumvented most of the time during testing, it is recommended that application builds without these protections are provided to maximise use of testing time.

5.12. Mobile Device Testing Methodology

Introduction

Mobile phones and tablets are increasingly issued by businesses to their staff to improve productivity and mobile working whilst still ensuring the security of data. These devices, however, can provide a gateway for an attacker to access internal corporate resources.

A penetration test of your mobile devices can help identify any misconfigurations or other vulnerabilities, giving you assurance that they won't be used to compromise the confidentiality or integrity of your company.

Testing Overview

There are three main activities:

- **Device Inspection: This is a check of the mobile device itself to ensure it is up to date and securely configured.**
- **Connectivity Inspection: The device will be checked to see if it connects to rouge wireless and mobile networks, if a private SIM is in use.**
- **MDM Assessment: When applicable, any MDM solution installed on the device will also be assessed, both the MDM configuration and the MDM application itself.**

Vulnerability and extended manual testing will explicitly identify where security holes lie and remove false positives. When applicable, other methods will be utilised.

Testing Breakdown

PTP will assess and thoroughly test the device and all associated resources within the scope. Any security issues identified will be highlighted and appropriate recommendations will be made to ensure all risks are appropriately

minimised. Any high-risk issues, or issues that could be exploited easily, will be reported immediately as they could pose an imminent threat, if discovered by an attacker. Testing is an iterative process. Each step feeds information into the next one to identify and exploit targets.

Device Inspection

Testing will be performed against an actual mobile device. This will include checks to ensure the device is up to date with the latest operating system, has a good security policy defined, and strong passwords or PINs are set.

Removable storage options are also assessed to test whether encryption is in use and that no sensitive data could be extracted should a device be lost or stolen, including via debug interfaces.

The device will also be assessed to ensure that any asset stickers or branding cannot directly identify the corporate owner of the device, leading to reputational damage if lost.

Connectively Inspection

Corporate mobile devices often have pre-defined wireless configuration settings installed on them. This assessment will review the device's wireless configuration and where applicable attempt to convince the device to connect to a rouge wireless access point, and intercept traffic.

In some instances, private data SIMs can be used (often in automotive, mobile tablet, and IoT scenarios) to provide mobile connectivity back to a corporate data centre. Efforts will be made to extract private APN configuration data with a view to accessing other corporate devices or servers that may be visible.

Any user VPNs will also be assessed for secure configuration and encryption.

MDM testing

Most companies employ some kind of MDM solution. Whilst the major providers in this field have regular penetration tests and generally deliver secure solutions, they rely on the server configuration being well-secured. Pen Test Partners have also found flaws in MDM solutions in the past that meant that a jailbroken or rooted device could be registered on a network whilst the device management server saw it as a standard device which could impact on security.

In general, when testing MDM solutions, it is prudent to start by assessing the configuration being sent to the end device before performing application testing against the actual MDM application.

Tools

For testing, PTP uses a combination of commercial and open-source tools that facilitate some procedures.

Full black box testing can be carried out, however, to maximise the efficiency of testing, a "break glass" methodology is recommended in order that escalating access to the device can be obtained if required.
Unless specifically requested, denial-of-service (DoS) attacks are not carried out.

Extended Manual tests

In most cases, issues highlighted by testing tools are validated manually to ensure that they are either present (and explored further) and valid or are a false positive.

Services that require authentication may be assessed for susceptibility to brute-forcing and hybrid dictionary-based attacks may be used.

Version numbers of detected operating systems and other software components are checked against known exploits.

Considerations for Testing

Testing should be performed on a representative corporate mobile device, on or off the corporate site. In cases where rouge wireless access points or mobile base stations are set up, it is strongly advised that this is performed offsite at PTP facilities to avoid disruption to normal business activities.

Deliverable

After the engagement has been completed, PTP provides a report detailing the security posture of the mobile devices under review.

5.13. Remote Access / VPN Assessment Methodology

Remote access solutions or VPNs are commonly used to provide access to remote working staff. To review these, we work alongside our other methodologies such as our external infrastructure and build review methodologies when reviewing VPN endpoints from an external perspective or reviewing VPN software from an on-host build review perspective.

VPN Assessment Approaches:

There are three types of approaches when reviewing VPNs or remote access solutions:

- External Perspective
- Software Deployment Perspective
- VPN Device Configuration Perspective

External Perspective:

In addition to our standard external infrastructure testing methodology (*PTP - Methodology - External Infrastructure.pdf*) additional checks are conducted which are VPN specific:

- Fingerprinting using tools such as “iker.py” or “ike-scan”
- Transforms Enumeration using tools such as “iker.py”
- Group ID Enumeration using such tools as “ike-scan”
- Cracking PSKs using such tools as “hashcat”
- Brute Force XAUTH using such tools as ikeforce.py

Software Deployment Perspective:

In addition to our standard build review methodology (*PTP - Methodology - Build Review.pdf*) where the software has been deployed to, additional checks are conducted which are VPN specific:

- Reviewing VPN software vulnerabilities
- Reviewing VPN software logging
- Reviewing VPN software permissions
- Reviewing VPN Software authentication

VPN Device Configuration Perspective:

In addition to our standard firewall and network device methodology (*PTP - Methodology - Firewall&Network Device Review.pdf*) where the VPN device has been configured, additional checks are conducted which are VPN specific:

- Reviewing VPN specific protocols
- Reviewing VPN specific authentication
- Reviewing VPN specific algorithms in use
- Reviewing VPN industry best practice settings

5.14. Web Application Assessment Methodology

Web applications are often critical to business and, if not properly secured, can lead to loss of sensitive information, damage to the brand's image, denial-of-service, and loss of revenue. It is recommended that all web applications are tested at least once a year and after major code changes.

The primary source of the Pen Test Partners web application assessment methodology is the OWASP Web Security Testing Guide (WSTG). However, reliance on a single static resource would result in vulnerabilities being missed. As a result, we use a combination of our own experience and techniques while keeping up to date with cutting-edge research, resulting in a hybrid testing methodology.

The testing process is driven by the application and how it functions. With many sites having complex flows and multiple user roles, it is essential that both automated and manual testing are used to discover deeply hidden issues. We search for weaknesses within the application, looking to chain vulnerabilities together to maximise their impact. Due to the rich and varied nature of web applications, we take an approach that combines frameworks such as OWASP and combine it with years of web application testing experience.

When a high level of assurance is required, source code can be used to augment web application testing. With the ability to directly observe the inner workings of a web application, it is possible to uncover weaknesses that would remain hidden as part of a normal web application test.

What should be covered as part of a web application security assessment?

Although using common web application frameworks and content management systems is becoming the norm for web application development, advanced customisation of these environments and in-house developed web applications means no two web applications are the same. Pen Test Partners recommends testing web applications from all available user perspectives, including unauthenticated perspective, to simulate the attack surface an attacker would have upon finding the site on the Internet, and an authenticated approach to simulate the perspective a genuine user would have.

Areas of testing should include:

- Access control (through all user roles including unauthenticated)
- Injection vulnerabilities (both client and server side)
- Cryptographic failures and data exposure
- Design flaws in application functionality
- Authentication flows, including the use of Multi-Factor Authentication (MFA) mechanisms
- Rate limitation techniques or race conditions
- Software and data Integrity (e.g. third-party dependencies and software supply chain)
- Enumeration of the web application and its files, directories, and sub directories to ensure no legacy, administrative, or backup files are present on the server

Common vulnerabilities can include:

- A lack of effective access control, where a lower privileged or unauthenticated user may horizontally or vertically escalate their privileges to access other users' information or higher privileged functionality.
- Injection vulnerabilities, in which an attacker can exploit vulnerable code to execute client or server-side code through the web application front-end. Examples of this can range from Cross-Site Scripting or client side template injection to SQL injection or direct host-level command injection.
- Vulnerable and outdated software components used by the web application may be exploited through use of publicly known vulnerabilities.
- Broken authentication flows, e.g. where lax Single-Sign On (SSO) or OAuth scopes may expose authentication tokens through insecure redirects.

Authenticated testing with client-provided user accounts will be conducted by default.

PTP will deploy consultants based on the size of the web application. In most cases, a single consultant will be assigned to an engagement; however, with large or complex web applications, multiple consultants may be used to deliver the project.

High and Critical Risk Vulnerabilities

At PTP, it is our policy to directly contact the designated client contact upon the discovery of a high or critical risk vulnerability, or one which poses an immediate threat to the environment or its users. Direct contact is made through email or phone with a detailed description of what the vulnerability is, how it would be exploited by an attacker, and how to remediate it as a temporary or permanent fix.

Example Vulnerabilities from Previous Engagements

To illustrate the type of real-world attack that a client can expect, these are some possible attacks based on previous tests carried out across a range of web applications.

From an unauthenticated perspective, the consultant was able to identify a vulnerability in the server software that exposed the partial names of all files within the web root. Using this vulnerability, they were able to enumerate a list of valid file names and directories on the application using a word list matching the partially exposed six-digit file name prefix. By visiting each of these files, a handful were found to be accessible prior to authenticating to the web application, one of which was an endpoint used to package resources into a compressed format. The consultant identified a valid parameter through fuzzing this endpoint which was found to accept a full URL scheme and filename.

The consultant then contacted the client regarding the endpoint and their findings. The client was happy for the consultant to replicate the process of what an external attacker would do and attempt to escalate the finding further. Using this parameter, the consultant attempted to query a listener service hosted on PTP's infrastructure with a HTTP request to see if the endpoint was vulnerable to Server-Side Request Forgery (SSRF). Upon observing a connection being made from an IP address belonging to the client's web server, the presence of an unauthenticated SSRF vulnerability was confirmed.

Through leveraging the SSRF to read local files using a specially crafted URI scheme, the consultant was able to retrieve encryption keys used by the application to preserve page and control values between round trips. Using these keys, the consultant was able to craft a serialised payload to demonstrate code execution by writing a static file to the web root which contained server-side code. Overall, this demonstrated the ability for an unauthenticated attacker to completely compromise the web application and its server from a black box perspective with no prior knowledge of the web application or its internal purpose. Constant communication with the client was carried out throughout the process of identifying the endpoint to creating a proof of concept for the code execution.

Testing

The testing is expected to follow the following phased approach:

Functionality Review and Analysis: The consultant will navigate to the URL(s) as described in the designated scope and familiarise themselves with the web application and environment. If a demonstration call or extensive documentation has been provided prior to the testing commencement date, this phase will be significantly shorter, as the consultant will already have knowledge of the web application and its functionality. Any further documentation provided by the client, such as associated web service documentation, will also be reviewed by the consultant to ensure they are familiar with what they are testing.

Reconnaissance: Time will be taken to perform a reconnaissance phase against the web application and its environment. This phase is used to identify the server software, CMS, and/or the framework in use by the web application. If the web application is hosted in a production environment that has, for example, been cached by search engines, publicly available information may also be collected by the consultant during this phase of testing. This phase may also include identifying which JavaScript files are visible prior to authenticating in an attempt to see what an attacker from an unauthenticated perspective can identify regarding the web application.

Enumeration and Site Map Generation: Information regarding the site structure and layout is passively collected by the consultant to work with throughout the engagement by browsing the application as a normal user would. During this phase, the consultant will also perform enumeration against the web application to identify hidden files, directories, or functionality which is not visible at the application layer. The purpose of this is to ensure no legacy functionality, administrative areas, or backup files are present on the server which an attacker may use as an indirect route to compromise the environment or its users.

Vulnerability Assessment: Based on previous stages, parts of the web application will undergo light vulnerability scanning. This identifies any obvious vulnerabilities that may be present in the web application to be detected early in the engagement. This follows a similar pattern to most vulnerability assessments, with asset discovery, application mapping, software and library identification, and manual testing. This aims to find weaknesses in either the design or implementation of the web application.

Attack Scenario and Demonstration: With knowledge of vulnerabilities and the design of the web application, viable attacks scenarios will be planned against the web application or its users. It may be possible at this stage to

build an example realistic attack scenario that an attacker may formulate. Sometimes, one or more vulnerabilities identified during the engagement may be chained together to create an advanced proof of concept attack. Ultimately, the goal is to promote defence-in-depth. Security controls should be present on many components of the web application to prevent attacks against the web application itself or its users.

Iteration: At this stage, it may be necessary to perform further vulnerability assessment. This could involve more in-depth testing of individual components or vulnerabilities and chaining of attack vectors together to create an example of a realistic real-world attack scenario. This phase is also used to ensure no false positive findings are present and to take necessary evidence used for the reporting phase.

Reporting

A report is written detailing the processes carried out and the issues found. Generally, this will contain a prioritised list of vulnerabilities discovered grouped by functional area. Remediation advice will be provided, both in terms of an immediate fix and any defence-in-depth measures that could be taken to mitigate risk. Attack-chains, alongside their impact, will be documented. Any higher-level findings that can be abstracted from the testing will be provided. Any architectural or design weaknesses will be highlighted so that these can be avoided in the future. Finally, an executive summary is produced to allow the most severe issues to be communicated quickly to stakeholders.

Requirements and Pre-Requisites

The following is required from the client:

User Testing Accounts (where applicable): This is used to permit authenticated testing. The more information that can be provided regarding each user role or privilege, the better. Information relating to the authentication process (e.g. OAuth, OpenID Connect, and SAML) and use of Multi-Factor Authentication (MFA) setup is also required where applicable.

If privilege escalation is a concern for the client, then accounts of differing user privileges should be supplied, e.g. Administrative account, normal user account, etc.

Should the web application be internally accessible only, a VPN solution or jump-box should be set up (if possible) ahead of testing for the consultant to access the web application.

Pre-access checks will be conducted ahead of testing by PTP and confirmed by the consultant to ensure testing runs smoothly on the commencement date.

Environment and Setup details

- A summary of what the web applications purpose is and brief description of functionality
- The web application stack details, examples of which may include:
    - Server software
    - Framework type
    - CMS type
    - In-house developed
- Whether the web application is hosted in a production or testing environment
- A non-production environment is recommended for testing as, despite all care being taken not to disrupt the performance of the server, the infrastructure might behave unexpectedly with the additional load that comes from penetration testing a web application.

- Whether the web application shares any backend services with neighbouring services (e.g. test environment's backend DBMS is shared with production)
- Whether there is sensitive administration functionality that should not be tested using automated tools (e.g. Administration Panel on production with the ability to modify pages or users)
- Where any third-party Web Application Firewalls (WAFs) are in place, such as Akami, Cloudflare, or F5 BIG-IP
    - PTP testing IP addresses should be added to an allow-list prior to testing to ensure testing of the actual application takes place.
- API documentation for web services that may be used by the web application as part of its overall functionality
    - Swagger documentation, Postman collections

Documentation and Demonstration (Optional but recommended)

- Design documents relating to the web applications architecture, including a network diagram of backend services in use and/or a wireframe of the web application's design.
- A demonstration call ahead of testing to provide a brief tour of the application to the consultant(s) to ensure when testing begins, they are familiar with the application and its functionality.

5.15. Wireless Testing Methodology

Wireless networks are commonplace in both home and corporate environments. They are more flexible and can be configured to a more secure standard than Ethernet-based local area networks. However, wireless signals cannot easily be constrained within the perimeter of a building or campus in the way that Ethernet can.

Wireless networking is nonetheless susceptible to attack due to configuration weaknesses or known or unknown security flaws in the software that implements the solution, including controllers, base stations, or access points and client stations.

Normally in a corporate environment there would be several wireless networks with different purposes, which would require adequate settings depending on the risk profile that they represent to the business. For example, a guest network that provides no connectivity to the corporate LAN, and a wireless network that provides access to corporate services would have very different risk profiles and would have different security requirements. Unlike Ethernet-based LANs, wireless networks are often accessible from locations outside office spaces, which increases their physical attack surface.

Several approaches could be taken to test wireless networks, depending on the risk profile of the network and the perceived threats to the solution. Perceived threats could be, for example, attacks from a public space with a high gain antenna or gaining access to a network by stealing and breaking into a mobile device to extract authentication information.

Assessment

As part of wireless network assessments, we conduct one or more of the following tests:

Onsite survey of access points (Aps) and connected client stations and detection of rogue access points within the client's premises:

- Surveying is conducted by tracking power levels using a low to average gain omni-directional antenna to find the general location of authorised APs, followed by a low to average gain directional antenna to obtain a more precise location.
- Particular attention is given to finding unauthorised APs located within the same space, and additional effort is made to physically locate them and ensure (within what is possible and allowed) that they do not bridge connectivity into the corporate LAN.

NOTE1: Surveying by tracking power levels can be limited by the fact that APs have varying average and peak power levels.
NOTE2: We do not normally provide maps or diagrams with locations of base stations, especially if the survey is conducted indoors.

Assessment of authentication methods against the scope of each specific network if more than one is in scope for testing: Attempts will be made to break into networks in scope by using technical means (WEP key cracking, WPA/PSK handshake capture and key cracking, evil twin attacks, etc.). This test would determine how likely would an attacker be able to gain unauthorised access into a wireless network from a public location.

Assessment of client station configuration on corporate laptops, workstations, or mobile devices: Single or dual factor authentication, PKI enforcement and server authenticity validation, provisioning process, locking access in case a device is stolen, etc. This test would determine the resilience of the networks against the compromise of a mobile device (inc. laptops).

Testing of traffic segregation features including client-to-client, traffic between wireless networks and wireless networks to Ethernet segments, traffic egress, etc.: We aim to propose a combination of the above techniques in a way that suits the purpose of each specific engagement carried out.

Example Scenarios

The scenarios described below are examples of how weaknesses found in wireless networks can translate into business risk by allowing attackers to affect users and data.

Access to Corporate LAN from the Guest Wireless Network: Unless segregation is correctly implemented between the multiple wireless networks and the wired corporate LAN, there is a possibility that a less protected guest wireless network permits access to a wired network that contains business-critical data. In some extreme cases, it would have been possible for attackers on the street to anonymously gain access into the corporate LAN which they could attack and potentially compromise.

Wireless Network Technology: All software is potentially affected by vulnerabilities and today all software requires regular maintenance and patching. Unless the wireless network infrastructure is kept up to date and has its configuration hardened to adhere to best security practices, it could have or develop weaknesses that attackers can exploit to gain levels of access that would normally not be available.

Client Stations on a Wireless Network: Clients of a particular wireless network could be seen as friendly or hostile. Typically, a well-protected wireless network that merges into the corporate LAN is meant to have friendly

(authorised) stations. Guest networks, on the other hand, will have stations connected of users with variable intentions and skill levels, which should be seen as hostile by default.

Stations connected to the guest wireless network could be legitimate trustworthy visitors with vested interest in the business and who may carry sensitive information. Other stations on the same network could have been anonymous attackers who guessed or decrypted access keys (if any). Therefore, it is possible that a client-to-client attack could cause damage to the business hosting the wireless network, even if it happens on the guest network.

Most wireless infrastructure technology support important security features that aim to defeat this type of scenario, such as client isolation, intrusion detection (and prevention) modules, and protection against rogue access points.

Rogue Access Points: Rogue access points are ones that are setup with the purpose of merging access to a wired network or to actively attack nearby client stations. These can be setup by employees to facilitate connectivity to specific devices, thus without malicious intent. Simultaneously, the strength of their configuration could mean that they become a weak point in the network and could be exploited by attackers to gain a level of access that would not have been possible on the business's wireless infrastructure.

In a more sinister scenario, 'evil twin' attacks consist of setting up access points configured to appear to be a legitimate infrastructure access point that aims to trick stations to connect to it, and either intercept traffic or obtain enough information to enable attacking of staff passwords. Some wireless infrastructure technologies can detect and accurately locate rogue access points within a defined area provided that sufficient access points are available.

Susceptibility to Denial-of-Service: Because of the way wireless technology works, it is possible for an attacker to attempt to de-authenticate connected stations, an attack which is usually successful.

The distributed nature of the access points would make it difficult but not impossible to implement a coordinated attack that drops all stations off the network.

Some wireless infrastructure technologies can detect and, to a limited extent, mitigate this type of attacks.

Authentication Mechanisms: Encryption and authentication of some kind should be required on all wireless networks. The strength of authentication however should be weighed against the risk posed by each network being broken into.

In a network where encryption is non-existent, even if a captive portal is used, attackers can simply eavesdrop on all information that any station sends and receives. With encryption enabled, specifically any type of WPA, each client station encrypts data using their own keys, thus rendering attackers (unauthenticated to the target network) unable to observe traffic in plaintext.

Any network that merges into the corporate LAN should be configured to require very strong authentication, ideally multi-factor authentication using certificates. However, care should be taken to ensure that all client stations are configured to strictly validate the integrity of the broker authentication server they connect to. Otherwise, they may be susceptible to 'evil twin' attacks.