Support Resources for Penetration Tests

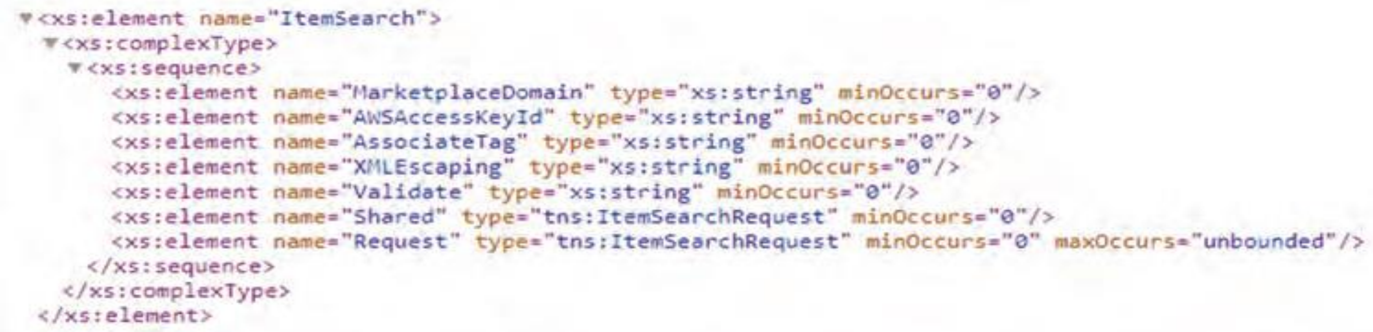
Penetration testers can take advantage of internal documentation to help plan their testing (and black box testers may manage to acquire this documentation during their work!) While there are a multitude of possible documents that each organization may have, documentation, accounts and access, and budget are all specifically described in the Pentesting objectives.

Documentation

The documentation that an organization creates and maintains to support its infrastructure and services can be incredibly useful to a penetration tester. While there are a multitude of possible documents that each organization may have, a few of the most common are described in the PenTest+ objectives, including these:

XML documentation like Web Services Description Language (WSDL), Web Application Description Language (WADL), SOAP, or other XML-based schema definitions. There are a multitude of XML-based standards that penetration testers may encounter. Fortunately, XML code is usually reasonably human-readable, and you should be able to get a general idea of what the definition or documentation describes by reading through it. Figure shows an example of Amazon's Product Advertising WSDL (found a http://webservices.amazon.com/AWSECommerceService/ AWSECommerceService.wsdl), which shows value types, operation definitions, and request/response formats.

An example of an API WSD:



```
▼<xs:element name="ItemSearch">
  ▼<xs:complexType>
    ▼<xs:sequence>
       <xs:element name="MarketplaceDomain" type="xs:string" minOccurs="0"/>
       <xs:element name="AWSAccessKeyId" type="xs:string" minOccurs="0"/>
       <xs:element name="AssociateTag" type="xs:string" minOccurs="0"/>
       <xs:element name="XMLEscaping" type="xs:string" minOccurs="0"/>
       <xs:element name="Validate" type="xs:string" minOccurs="0"/>
       <xs:element name="Shared" type="tns:ItemSearchRequest" minOccurs="0"/>
       <xs:element name="Request" type="tns:ItemSearchRequest" minOccurs="0" maxOccurs="unbounded"/>
     </xs:sequence>
   </xs:complexType>
 </xs:element>
```

Application programming interface (API) documentation describes how software components communicate. While APIs can be described in many ways, including via the Web Services Description Language (WSDL), tools such as Swagger, Apiary, and RAML are some of the most popular ways of developing and documenting the REST full APIs that are part of many modern service stacks. So, access to a Swagger document provides testers with a good view of how the API works and thus how they can test it.

Software development kits (SDKs) also provide documentation, and organizations may either create their own SDKs or use commercial or open-source SDKs. Understanding which SDKs are in use, and where, can help a penetration tester test applications and services.

Internal documentation may also include examples like sample application requests, API examples, or other useful code that testers can use to validate or improve their own testing. This is particularly useful for penetration tests that are directed at web applications or APIs.

Architectural diagrams, dataflow diagrams, and other system and design documentation can provide penetration testers with an understanding of potential targets, how they communicate, and other configuration and design details.

Configuration files can be treasure troves of information and may contain details including accounts, IP addresses, and even passwords or API keys.

Access and Accounts

White box assessments will provide direct access to the systems that are being tested. This may include permitting penetration testers past defenses that are normally in place. A black box assessment team won't have that luxury and will have to make their way past those defenses. Common security exceptions for white box tests are as follows:

Whitelisting testers in Intrusion Prevention Systems (IPSs), Web Application Firewalls (WAFs), and other security devices will allow them to perform their tests without being blocked. For a white box test, this means that testers won't spend time waiting to be unblocked when security measures detect their efforts. Black box and red-team tests are more likely to result in testers being blacklisted or blocked by security measures.

Security exceptions at the network layer, such as allowing testers to bypass network access controls (NACs) that would normally prevent unauthorized devices from connecting to the network.

Bypassing or disabling certificate pinning.

what is certificate Pinning?

Certificate pinning associates a host with an X.509 certificate (or a public key) and then uses that association to make a trust decision. That means that if the certificate changes, the remote system will no longer be recognized, and the client shouldn't be able to visit it. Pinning can cause issues, particularly if an organization uses data loss prevention (DLP) proxies that intercept traffic. Pinning can work with this if the interception proxy is also added to the pinning list, called a pinset.

Access to user accounts and privileged accounts can play a significant role in the success of a penetration test. White box assessments should be conducted using appropriate accounts to enable testers to meet the complete scope of the assessment. Black box tests will require testers to acquire credentials and access. That means a strong security model may make some desired testing impossible a good result in many cases, but it may leave hidden issues open to insider threats or more advanced threat actors.

Physical access to a facility or system is one of the most powerful tools a penetration tester can have. In white box assessments, testers often have full access to anything they need to test. Black box testers may have to use social engineering techniques or other methods we will discuss later in this book to gain access.

Network access, either on site, via a VPN, or through some other method, is also important, and testers need access to each network segment or protected zone that should be assessed. That means that a good view of the network in the form of a network diagram and a means to cross network boundaries are often crucial to success.

Budget

Technical considerations are often the first things that penetration testers think about, but budgeting is also a major part of the business process of penetration testing. Determining a budget and staying within it can make the difference between a viable business and a failed effort.

The budget required to complete a penetration test is determined by the scope and rules of engagement (or, at times, vice versa if the budget is a limiting factor, thus determining what can reasonably be done as part of the assessment!). For internal penetration testers, a budget may simply involve the allocation of time for the team to conduct the test. For external or commercial testers, a budget normally starts from an estimated number of hours based on the complexity of the test, the size of the team, and any costs associated with the test such as materials, insurance, or other expenditures that aren't related to personnel time.

In Short:

Documents: These explain how systems work, like API documentation (how software components talk to each other) or network diagrams (showing how devices are connected).

Access: Testers might need different levels of access depending on the test type. In some cases, they get direct access (white box testing), while in others, they need to find ways in like a real attacker (black box testing).

Accounts: Testers may use different accounts to test systems, including regular user accounts or special privileged accounts (white box testing). Black box testers might need to find their own credentials.

Budget: Penetration testing costs money. The price depends on how complex the system is, how long it takes to test, and the team's size.
Here's a real-time example based on the information provided:

RealTime Example :ONLINE STORE APP

Imagine you're a penetration tester tasked with evaluating the security of an online store (let's call it "Sneaker Haven").

Using Documentation:

- You might start by looking at Sneaker Haven's developer documentation. This could be in the form of a Swagger document that explains how their website interacts with its mobile app (through an API). Understanding this API helps you identify potential weaknesses in how the app communicates with the website.
- Additionally, you might find network diagrams showing how Sneaker Haven's servers are connected. This helps you understand the overall structure of their system and identify potential entry points for attacks.

Access and Accounts:

- In a white-box scenario, Sneaker Haven might grant you a temporary account with limited privileges to test their website and app. This allows you to explore the system from a user's perspective and identify vulnerabilities like weak password requirements or exploitable features.
- In a black-box scenario, you might try finding ways to gain unauthorized access, like attempting to guess a weak login password or exploiting a known vulnerability in their website software.

Physical and Network Access:

- In a white-box test, you might be granted physical access to Sneaker Haven's server room (if they have one) to examine their network configuration more closely.
- In a black-box test, your network access might be limited to the public internet. You would then need to use your skills to find ways to connect to Sneaker Haven's network and explore it for vulnerabilities.

Budget Considerations:

- The scope of your testing depends on the agreed-upon budget. A longer and more in-depth test will naturally cost more than a quick scan. In this case, Sneaker Haven might decide on a budget based on the complexity of their website and app.

By using a combination of these resources, you can identify potential security weaknesses in Sneaker Haven's systems and recommend ways for them to improve their defenses.

Real-Time Example: Network Penetration Test for a Coffee Shop

Here's a scenario where we apply the concepts from the content to a real-time example of a network penetration test for a coffee shop named "Brews & Bytes":

Documentation:

- We request access to Brews & Bytes' network diagrams. These diagrams will show how devices like routers, firewalls, and point-of-sale (POS) systems are connected. This helps us understand the overall network layout and identify potential entry points for attacks.
- We also request any security policies they have in place. These policies might outline password complexity requirements, acceptable use guidelines for employee devices connecting to the network, and vulnerability patching procedures.

External Resources:

- We use vulnerability scanners to scan Brews & Bytes' publicly accessible IP addresses. These scanners identify known vulnerabilities in the software versions used by their devices.
- We search online for any reported vulnerabilities specific to the coffee shop industry or the POS systems they might be using. This helps us focus our testing efforts on areas with a higher likelihood of success.

Network Access:

- We attempt to exploit any identified vulnerabilities in their publicly accessible devices. This might involve techniques like SQL injection attacks against their website or attempting to gain unauthorized access to their wireless network.
- We perform social engineering techniques (**with prior permission** during the engagement) to see if employees might accidentally disclose sensitive information or click on malicious phishing links.

Internal Access (White-Box Testing Only):

- If it's a white-box test, Brews & Bytes might grant us temporary access to their internal network with limited privileges. This allows us to test the security of their internal systems, like the POS system, and identify vulnerabilities that wouldn't be accessible from the public internet.
- With internal access, we might use tools to analyze network traffic and identify any suspicious activity or unauthorized communication.

Accounts (White-Box Testing Only):

- During a white-box test, Brews & Bytes might provide us with credentials for different user accounts, including employee accounts and potentially an administrator account. This allows us to test the effectiveness of their access controls and identify weaknesses in user permission levels.

Budget:

- The scope of our testing is determined by the agreed-upon budget with Brews & Bytes. A more limited budget might focus on scanning for publicly known vulnerabilities, while a larger budget could allow for more in-depth testing, including social engineering attempts and internal network assessments (if applicable).

By combining these techniques, we can identify potential security risks in Brews & Bytes' network and recommend solutions to improve their overall security posture. It's important to note that ethical hacking and penetration testing should always be conducted with the client's permission and within a defined scope to avoid any legal issues.