Though this module focuses on nine specific applications, there are still many different ones that we may encounter in the wild. I have performed large penetration tests where I ended up with an over 500-page EyeWitness report to go through.

The module was designed to teach a methodology that can be applied to all other applications we may encounter. The list of applications we covered in this module covers the main functions and most of the objectives of the vast number of individual applications to increase the effectiveness of your internal and external assessments during your penetration tests.

We covered enumerating the network and creating a visual representation of the applications within a network to ensure maximum coverage. We also covered a variety of ways that we can attack common applications, from fingerprinting and discovery to abusing built-in functionality and known public exploits. The aim of the sections on osTicket and GitLab was not only to teach you how to enumerate and attack these specific applications but also to show how support desk ticketing systems and Git repository applications may yield fruit that can be useful elsewhere during an engagement.

A big part of penetration testing is adapting to the unknown. Some testers may run a few scans and become discouraged when they don't see anything directly exploitable. If we can dig through our scan data and filter out all of the noise, we will often find things that scanners miss, such as a Tomcat instance with weak or default credentials or a wide-open Git repository that gives us an SSH key or password that we can use elsewhere to gain access. Having a deep understanding of the necessary methodology and mindset will make you successful, no matter if the target network has WordPress and Tomcat or a custom support ticketing system and a network monitoring system such as Nagios. Ensure that you understand the various techniques taught for footprinting these applications and the curiosity to explore an unknown application. You will come across applications not listed in this module. Similar to what I did with the Nexus Repository OSS application in the introduction section, you can apply these principles to find issues like default credentials and built-in functionality leading to remote code execution.

**Honorable Mentions**

That being said, here are a few other applications that we have come across during assessments and are worth looking out for:

| **Application** | **Abuse Info** |
| --- | --- |
| [Axis2](https://axis.apache.org/axis2/java/core/) | This can be abused similar to Tomcat. We will often actually see it sitting on top of a Tomcat installation. If we cannot get RCE via Tomcat, it is worth checking for weak/default admin credentials on Axis2. We can then upload a [webshell](https://github.com/tennc/webshell/tree/master/other/cat.aar) in the form of an AAR file (Axis2 service file). There is also a Metasploit [module](https://packetstormsecurity.com/files/96224/Axis2-Upload-Exec-via-REST.html) that can assist with this. |
| [Websphere](https://en.wikipedia.org/wiki/IBM_WebSphere_Application_Server) | Websphere has suffered from many different [vulnerabilities](https://www.cvedetails.com/vulnerability-list/vendor_id-14/product_id-576/cvssscoremin-9/cvssscoremax-/IBM-Websphere-Application-Server.html) over the years. Furthermore, if we can log in to the administrative console with default credentials such as system:manager we can deploy a WAR file (similar to Tomcat) and gain RCE via a web shell or reverse shell. |
| [Elasticsearch](https://en.wikipedia.org/wiki/Elasticsearch) | Elasticsearch has had its fair share of vulnerabilities as well. Though old, we have seen [this](https://www.exploit-db.com/exploits/36337) before on forgotten Elasticsearch installs during an assessment for a large enterprise (and identified within 100s of pages of EyeWitness report output). Though not realistic, the Hack The Box machine [Haystack](https://youtube.com/watch?v=oGO9MEIz_tI&t=54) features Elasticsearch. |
| [Zabbix](https://en.wikipedia.org/wiki/Zabbix) | Zabbix is an open-source system and network monitoring solution that has had quite a few [vulnerabilities](https://www.cvedetails.com/vulnerability-list/vendor_id-5667/product_id-9588/Zabbix-Zabbix.html) discovered such as SQL injection, authentication bypass, stored XSS, LDAP password disclosure, and remote code execution. Zabbix also has built-in functionality that can be abused to gain remote code execution. The HTB box [Zipper](https://youtube.com/watch?v=RLvFwiDK_F8&t=250) showcases how to use the Zabbix API to gain RCE. |
| [Nagios](https://en.wikipedia.org/wiki/Nagios) | Nagios is another system and network monitoring product. Nagios has had a wide variety of issues over the years, including remote code execution, root privilege escalation, SQL injection, code injection, and stored XSS. If you come across a Nagios instance, it is worth checking for the default credentials nagiosadmin:PASSW0RD and fingerprinting the version. |
| [WebLogic](https://en.wikipedia.org/wiki/Oracle_WebLogic_Server) | WebLogic is a Java EE application server. At the time of writing, it has 190 reported [CVEs](https://www.cvedetails.com/vulnerability-list/vendor_id-93/product_id-14534/Oracle-Weblogic-Server.html). There are many unauthenticated RCE exploits from 2007 up to 2021, many of which are Java Deserialization vulnerabilities. |
| Wikis/Intranets | We may come across internal Wikis (such as MediaWiki), custom intranet pages, SharePoint, etc. These are worth assessing for known vulnerabilities but also searching if there is a document repository. We have run into many intranet pages (both custom and SharePoint) that had a search functionality which led to discovering valid credentials. |
| [DotNetNuke](https://en.wikipedia.org/wiki/DNN_(software)) | DotNetNuke (DNN) is an open-source CMS written in C# that uses the .NET framework. It has had a few severe [issues](https://www.cvedetails.com/vulnerability-list/vendor_id-2486/product_id-4306/Dotnetnuke-Dotnetnuke.html) over time, such as authentication bypass, directory traversal, stored XSS, file upload bypass, and arbitrary file download. |
| [vCenter](https://en.wikipedia.org/wiki/VCenter) | vCenter is often present in large organizations to manage multiple instances of ESXi. It is worth checking for weak credentials and vulnerabilities such as this [Apache Struts 2 RCE](https://blog.gdssecurity.com/labs/2017/4/13/vmware-vcenter-unauthenticated-rce-using-cve-2017-5638-apach.html) that scanners like Nessus do not pick up. This [unauthenticated OVA file upload](https://www.rapid7.com/db/modules/exploit/multi/http/vmware_vcenter_uploadova_rce/) vulnerability was disclosed in early 2021, and a PoC for [CVE-2021-22005](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22005) was released during the development of this module. vCenter comes as both a Windows and a Linux appliance. If we get a shell on the Windows appliance, privilege escalation is relatively simple using JuicyPotato or similar. We have also seen vCenter already running as SYSTEM and even running as a domain admin! It can be a great foothold in the environment or be a single source of compromise. |

Once again, this is not an exhaustive list but just more examples of the many things we may come across in a corporate network. As shown here, often, a default password and built-in functionality are all we need.