Alyssa Herrera

Follow 1.5K Followers About

High Risk Vulnerabilities within the DoD - Exploiting Coldfusion, Dotnet Nuke, Oracle, and more

 Alyssa Herrera Jan 25, 2018 · 7 min read ★

Introduction

The Department of Defense Launched a bug bounty program on November 21st, 2016 on Hackerone. This allowed researchers to report vulnerabilities on any military domain, *.mil and DoD linked IP's as well. When the program launched, I wanted to use this program as an opportunity to help out the DoD's Website security but also as a chance to learn and sharpen my own skills.

The purpose of this post is to highlight unique and common place vulnerabilities that can be applied if you plan to look into the DoD program or in your own bug bounty hunt, and what I have learned from this engagement on the DoD's program. I am currently listed in the 8th spot on the leader board for the program and I will be disclosing my reports with an appropriate summary describing the vulnerabilities

Reconnaissance

Due to the quite large scope of the program it can be hard to pick a website or a sub domain to search for vulnerabilities. Lucky enough we can use a google dork to simplify this search. We can search through the entire website we selected and the sub domain for interesting files or potentially vulnerable end points with the following google dork, *site:*.*.mil* or *site:*.website.mil*.

If you don't know what google dorks are then here's a TL;DR we can use operators to specify what to look for, we can specify a domain name for our search to look at specifically, we can further narrow this down to file extension, a specific URL string, etc. We use site to specify to search the desired website specifically and with two wild cards to indicate to check sub domains as well, this is very broad search method. The other dork will look at the specific website and the subsequent sub domains.

We can use this to say search for flash applications that are known to be vulnerable, or look for a specific cms installation, this is how I found a majority of my vulnerabilities for the program. We can use tools such as Aquatone if we wanted to find all the sub-domains in a given website and then port scan them as well, a helpful tip is, if you notice that a sub-domain has a sub-domain within it self, such as dev.subdomain.website.com, then you should specify that sub-domain and check if there may be more sub-domains with in that sub-domain, this can help you find additional hosts.

These two recon techniques were extremely helpful overall and are typically applied to many bug bounty targets. Another technique that can help you find new end points to test or find sensitive information or hidden panel links is reading through JS files being used, there are tools that can be used to extract links in side of them such as linkfinder to make this process a lot quicker. I will go in depth about this in a later post if there's enough interest for it.

Vulnerabilities Discovered

The majority of vulnerabilities I discovered were low hanging fruit such as XSS, many of them unsurprisingly stemmed from Login forms, Error messages, and flash applications. Many of them aren't worth the time to explain since they're quite trivial to find, there's quite a good list of public information on known flash application XSS like flashmediaelement, video-js, jwplayer or plupload.

What I will focus on is primarily vulnerabilities that are high severity to medium issues since they're quite easy to find even for people who are quite new to bug bounties. Cold Fusion is heavily used for many military websites and they tend to have a treasure trove of vulnerabilities I would recommend Cold Fusion for pen testers to read about vulnerabilities in cold fusion and more in depth information on it.

One thing I have noticed about cold fusion websites is that SQL injection is quite pervasive and easy to find with in websites that uses cold fusion. With one website I discovered two separate instances of an SQL injection, one example is as followed, website.com/news/news.cfm?newsItem=1 and to check I used a single apostrophe which spit out an SQL error, I then proceeded to test the vulnerability using time based query which made the website pause for several seconds before responding.

These vulnerabilities were quite easy to spot especially with coldfusion's tendency to have verbose errors. In the same website I discovered the SQL injection I also discovered an Apache Solr Instance, which is similar to elasticsearch, that was publicly exposed, this was found by checking /solr/ directory. Recently an RCE was published for Apache Solr, unfortunately the instance I discovered wasn't affected but this provided a good trove of information about the website and I also found an XSS in one of the functions of the instance.

One resource useful for finding default files and potentially interesting files is this github . The other more pervasive issues I found within DoD websites were the result of DotNetNuke or DNN, which can be seen in websites with the /DNNCorp/ or /DesktopModules/ directories. My publicly disclosed reports on arbitrary file download were the result of DNN module that had a known exploit for arbitrary file download. The vulnerable endpoint looked like this xxx.xx.mil/desktopmodules/eventscalendar/downloaddoc.aspx?f=/file/test.doc which should set off alarm bells for anyone looking for arbitrary file download; it can be exploited by using a ~/file.

Oracle Instances are also quite popular with military websites, and many of them are running on outdated versions, one instance that I came across quite often would be Oracle People Soft. Oracle people soft has several vulnerabilities that were discovered by ErpSoft, their advisories usually include proof of concepts along with explanations of the vulnerabilities.

An easy way to search for oracle people soft is looking for /pspc/ or /psigw/ directories. They usually are vulnerable to a non authed XXE, SSRF and XSS. We can exploit the SSRF vulnerability as followed, first we visit *website.com/IMServlet?Method=CONNECT*

Then we visit
website.com/IMServlet?
Method=GOOGLE_PRESENCE&im_to_user=abc&im_server_name=GOOGLE&im_server=Host:Port/

You can also check if the end point is functioning by pointing it to your website first to see if you receive a connection from the website. This can of course lead to reading services and accessing sensitive internal instances on the website.

Notable Vulnerabilities discovered

I thought I should share some of the most notable vulnerabilities I discovered, either because they were unique, had high impact or other factors. The two most notable vulnerabilities that were discovered were two open FTP panels that allowed me to view files that were hosted on the website although the websites weren't used anymore thus received lower severity rating due to the lack of file creation. These vulnerabilities were discovered through Shodan.io and were quite surprising to find.

One of the most unique vulnerabilities I found let me harvest the email accounts of users who used the website. The vulnerability stemmed from an activate your account field, if you provide a valid username, the website will say "Activation email sent to user@email.com" and the field had no rate limiting either.

The first thing I did was to input "Admin" this provided the private email account of the administrator, now this PII leak could combined with a script that would run through a list of common usernames and we could then harvest emails of users. The consequences of this would mean an admin account could be targeted specifically by attackers this also meant users of the websites were also at risk.

I'll close this section with the most impactful vulnerability I discovered so far. This website demonstrated a quite classic file disclosure vulnerability, the website contained a feature to download pdf files and documents off the website, this was quite similar to the events calendar file disclosure vulnerability. The website looked as followed,

website.com/xxx/account/downloadfile?fileString=/pathtodocument/document.pdf. We could view the etc/shadow or other sensitive files by traversing the directory structure using ../ and then specifying a file.

I was able to retrieve the /etc/shadow file which contained an MD5 hashed root password which would be quite trivial to crack although I didn't crack it and simply reported my findings. Although the vulnerability was quite impactful, this subdomain was going to be taken down due to being a legacy site and thus was only a low severity issue.

Conclusion

The Department of Defense public program has demonstrated that security through bug hunters can prevent breaches of their assets by giving hackers a way to report vulnerabilities they encountered. What I learned through my continuous engagement on DoD assets helped me improve as a bug hunter and further improve my experience as one.

One of this hardest thing for many bug hunters is where to start for programs, simple reconnaissance through google dorking or using shodan.io can reveal trove of vulnerabilities to be discovered and reported. Through my blog post I demonstrated lesser known areas you could check for vulnerabilities, as I was surprised that not many people were familiar with dotnet nuke or oracle instances that had known vulnerabilities.

The biggest thing as a bug hunter is you should strive to keep learning and never stop learning, new exploits will be discovered, new attack surfaces might appear like James Kettle's Cracking the Lens: Targeting HTTP's Hidden Attack-Surface. I have made disclosure requests for the vulnerabilities I discussed here and I have made several of them public already, you can check them on my profile.

I'll be closing on this one final note, the bug hunter community is quite welcoming to new bug hunters, don't be afraid to ask questions or ask for help with exploiting vulnerabilities.

Security Information Security Bug Bounty Web Application Security

About Help Legal

Get the Medium app