Hack the Pentagon For DoD Partners

October 2022

We appreciate your interest in executing a Bug Bounty with us! We bring expertise, long standing relationships with industry partners and security researchers, and execution authority from the Secretary of Defense, consistent with Vulnerability Disclosure Program principles.

Below you will find information answering common questions and important facts in the running of a Bug Bounty with us.

Policy Docs: Security Researchers And The Handling Of PII & PHI

DoD in all policy references the standing OMB Memo M-17-12 that defines PII/PHI and handling.
View M-17-12 →

OMB Memo M-20-32 to clarify how vulnerability discovery programs impact existing guidance to include M-17-12 and PII/PHI. The clearest guidance comes from the first section, specifically bullet point five, which DDS believes exempts activity done within the scope of authorized testing from the consequences of M-17-12.
View M-20-32 →

DOJ Prosecution Guidelines for CFAA
Learn More →

Bug Bounties Within The DoD

Why Bug Bounties

A. What Are They

A Bug Bounty is an event where ethical hackers are awarded monetary compensations for successfully discovering and reporting a vulnerability through the appropriate channels.

Bug Bounties have been around for several years and are used by some of the biggest companies in the world such as Apple, CISCO, Facebook, Google and Twitter.

B. Why Are They Done

A Bug Bounty program is a cost-effective way for an organization to identify security risks and vulnerabilities. The program allows organizations to have experienced ethical hackers from diverse backgrounds proactively identifying weaknesses so they can be remediated.

C. How Are They Valuable

Bug Bounties allow the DoD to find and address vulnerabilities in a rapid, cost-effective way - before the adversary does.

Difference In Bug Bounties And The VDP

A. Bug Bounties vs. VDP

A Bug Bounty program gives ethical hackers permission to test an organization's applications for certain types of vulnerabilities resulting in monetary payment. In comparison a Vulnerability Disclosure Program (VDP) relies on hackers to "see something, say something" where anyone can submit security vulnerabilities to help mitigate risks and is not paid.

For more information on the VDP, visit: www.dc3.mil

B. Classification Within Bug Bounties And The VDP

Most bounties are unclassified, however if vulnerabilities are encountered with a higher risk, they will be escalated appropriately.

Our Vendors And Researchers

A. Our Vendor Partners

Currently, we work with a small collection of 3rd Party vendors who source the researchers from their extensive databases. Additionally, they manage the platforms for communication, validating submissions and payment to the researchers.

B. Our Security Researchers

The researchers for our Bug Bounties are vetted with an extensive background screening from our approved vendors. They do not have a preset schedule and can access an active program based on predetermined scope from the system owners. We pride ourselves on user researchers from a broad background so in addition to U.S. Nationals, they can be from NATO and FVEY nations.

The Basics Of Running A Bug Bounty

The Bug Bounty Lifecycle

A. Timeline

DDS Bug Bounties follow a 4 week x 4 week x 4 week model that starts after the Task Order is approved by Procurement Officers.

B. Phase 1 - Bounty Preparation

Bounty prep will include defining the scope and ensuring backend connection to systems.

C. Phase 2 - Bounty Launch

Bounty launch will include kick off, training on the vendor's portals, monitoring of the vulnerabilities and payments from the vendor directly to the researchers..

D. Phase 3 - Bounty Wrap-Up, Report, and Remediation

Bounty wrap up will include review of metrics, out-briefs and final remediation.

Scooping & Cost

A. Scoping

Scoping for a bug bounty effort is a critical step in hosting an effective bounty program. Tuning your scope to fit the security needs of your organization can make a significant difference in researcher engagement, meeting your program goals, and creating a lasting positive effect on your cyber security.

Mapping Scope to Goals:
One of the most effective ways to scope an engagement is to begin by defining the goals or outcomes you wish to see from a bounty program. If you aim to see code development improvement then scoping in source code, deployment pipelines, and production applications may make for an effective choice. If you are hoping to identify weaknesses in your deployed application then scoping in the application and any supporting services (cloud hosting, web application firewall, load balancing) may be preferable.

Incentivize Worst Case Scenarios:
Another way to best identify scope is to table top worst case scenarios. If an attacker got in, what actions or systems would be the worst possible outcomes of a breach? Using these "nightmare" scenarios as a basis, it is possible to add incentives to vulnerabilities that would lead directly (or chain indirectly) to these outcomes. If a specific database holds sensitive information such as Personally Identifiable Information (PII) or Protected Healthcare Information (PHI), incentivizing vulnerabilities against these specific assets would be prudent.

Test/Staging vs. Production Environments:
Many organizations implement a testing environment which may be referred to as staging, test, development, user acceptance testing, or pre-production environments. Organizations may be inclined to offer these environments for scope to avoid negative impacts to production. The standing recommendation for scoping is to test the production environment if at all possible. While many test environments claim to be exact copies of production, in all prior experience nuances between these environments leads system owners to pay for vulnerabilities that do not exist in the production instance. Additionally, vulnerabilities may exist in production environments that do not exist in any pre-production instances. Asset owners sometimes profess concern about the risk of researchers accessing production environments. In our experience, the detailed rules of engagement put in place for bug bounties, combined with detailed logging of researcher actions and researchers' need to maintain their professional reputations, provide more than sufficient risk mitigation.

Legal Protections for Vulnerability Disclosure:
Organizations routinely fear perceived legal and policy consequences for vulnerabilities found during a bug bounty or a vulnerability disclosure program. Federal policy has been updated to provide protection to security researchers operating in good faith. M-20-32 provides the latest guidance on vulnerability research performed during bug bounties and vulnerability disclosure programs. It is clearly stated in that memo that, "Good-Faith Security Research is Not an Incident or Breach". However, in the process of assessing and responding to the reported vulnerability, it is possible to discover that an incident or breach occurred prior to the vulnerability report. Any discovered incident or breach must be handled according to the OMB Memo M-17-12. *Both Memo's are linked on page 1.*

Scoping for a bug bounty effort is a critical step in hosting an effective bounty program. Tuning your scope to fit the security needs of your organization can make a significant difference in researcher engagement, meeting your program goals, and creating a lasting positive effect on your cyber security.

Mapping Scope to Goals:
One of the most effective ways to scope an engagement is to begin by defining the goals or outcomes you wish to see from a bounty program. If you aim to see code development improvement then scoping in source code, deployment pipelines, and production applications may make for an effective choice. If you are hoping to identify

weaknesses in your deployed application then scoping in the application and any supporting services (cloud hosting, web application firewall, load balancing) may be preferable.

Incentivize Worst Case Scenarios:
Another way to best identify scope is to table top worst case scenarios. If an attacker got in, what actions or systems would be the worst possible outcomes of a breach? Using these "nightmare" scenarios as a basis, it is possible to add incentives to vulnerabilities that would lead directly (or chain indirectly) to these outcomes. If a specific database holds sensitive information such as Personally Identifiable Information (PII) or Protected Healthcare Information (PHI), incentivizing vulnerabilities against these specific assets would be prudent.

Test/Staging vs. Production Environments:
Many organizations implement a testing environment which may be referred to as staging, test, development, user acceptance testing, or pre-production environments. Organizations may be inclined to offer these environments for scope to avoid negative impacts to production. The standing recommendation for scoping is to test the production environment if at all possible. While many test environments claim to be exact copies of production, in all prior experience nuances between these environments leads system owners to pay for vulnerabilities that do not exist in the production instance. Additionally, vulnerabilities may exist in production environments that do not exist in any pre-production instances. Asset owners sometimes profess concern about the risk of researchers accessing production environments. In our experience, the detailed rules of engagement put in place for bug bounties, combined with detailed logging of researcher actions and researchers' need to maintain their professional reputations, provide more than sufficient risk mitigation.

Legal Protections for Vulnerability Disclosure:
Organizations routinely fear perceived legal and policy consequences for vulnerabilities found during a bug bounty or a vulnerability disclosure program. Federal policy has been updated to provide protection to security researchers operating in good faith. M-20-32 provides the latest guidance on vulnerability research performed during bug bounties and vulnerability disclosure programs. It is clearly stated in that memo that, "Good-Faith Security Research is Not an Incident or Breach". However, in the process of assessing and responding to the reported vulnerability, it is possible to discover that an incident or breach occurred prior to the vulnerability report. Any discovered incident or breach must be handled according to the OMB Memo M-17-12. *Both Memo's are linked on page 1.*

B. Basic Questions For Your Initial Scope

➔ What are your goals?
➔ What's your nightmare scenario?
➔ What assets are you testing?
➔ What's your timeline?
➔ Do you have funding?
➔ What kind of bounty do you want to run? (public, private, classified, not sure)

C. Detailed Information We Typically Request After Approval

➔ URL's (wildcards are acceptable but please approximate size of infrastructure)
➔ IP Addresses, if not covered by URLs
➔ Code bases, if applicable (approximate size by lines of code)
➔ Technology types or appliances
➔ Nightmare scenarios or goals for testing if you have a scenario you're particularly interested in testing (permission escalations, etc.)
➔ Will you be linking the JITC certificates (Functionally emulates CAC authentication) to identities within your infrastructure?
➔ Are you interested in testing multiple authenticated security roles within the same application?
➔ Short unclassified (publicly releasable) descriptions of what each URL or code base does to help researchers understand its role or purpose.

D. Cost
When you begin the process of a Bug Bounty, you and the vendor set the price and scope of the bounty in advance, which means you have control over the process. If researchers submit duplicates or vulnerabilities that are not valid, they will not be paid.

However, it's important that you do not exploit this rule because if your reputation as a system owner is equally important in the community, you want to prevent gaining the reputation that you do not pay then it becomes difficult to attract top researchers to future bounties.

In general system owners can expect to pay $250K—800K per bounty, with most of the money going to researchers.

<u>You're A Good Candidate For A Bug Bounty If...</u>

What We Look For

A. Significance Of The App/System To Be Tested

Since we can only support so many bounties per year, it's important to prioritize our efforts on systems that are of significance to DoD or our partners' mission.

B. Executive Buy-In
Goes without saying: it is critical that we have a champion for the bounty both at the leadership/decision-maker level, but also a reliable champion at the technical & program management levels to ensure a smooth effort.

C. Technical Maturity
To reap the most benefit from a Bug Bounty, the DoD partner should have a technical shop, enough to (a) quickly & effectively support the technical aspects of a bounty, like allowing researcher aspect to the network and (b) effectively remediate the discovered vulnerabilities and learn from trends after the bounty completes.

D. Attitude/Vision
An ideal HTP partner program is one with leaders and folks who understand the value of security assessment, including bounties, and don't suffer from the mindset of 'this will be bad because it will show my superiors how vulnerable my product is'.

E. Identifying Funds
Ideally the DoD 'customer' will fully fund the bounty. For Army & Air Force groups that can't fund the bounty, reach out to us since DDS receives funding from Army/AF that we may be able to direct towards a bounty. Worst case, we can petition DDS leadership to direct some DDS funds towards the bounty.