FORTRA

GUIDE *(Cybersecurity)*

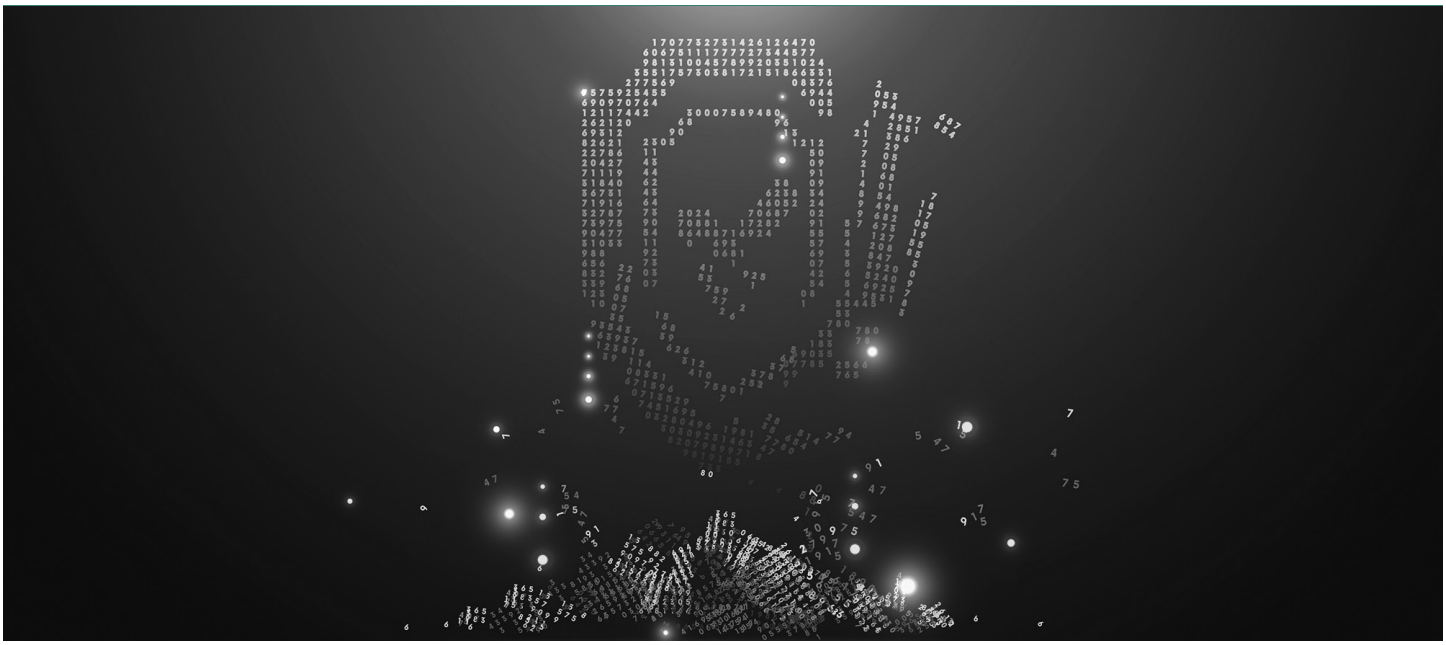
# How Black Box Fuzzers Protect Against the Unknown

## Unpublished vulnerabilities are the unsung menace threatening organizations today.

When it comes to vulnerabilities, what you don't know can hurt you. Cyber attackers are looking for new ways to exploit software and devices every day. Just because a system doesn't have any known vulnerabilities does not mean it is secure. There are almost infinite possibilities for input into a product, and any one of them might lead to an undiscovered security flaw.

Vulnerability analysis is crucial for detecting known issues and exploits, however, it is not equipped to find bugs in software or identify unknown vulnerabilities in a product. Research has shown that 80% of exploits are published almost 23 days before an official CVE is released. This gives attackers a considerable head start on the organizations that focus only on detecting and remediating known vulnerabilities.

How do organizations take the next step to advance the security of their products without disrupting productivity? In this e-book, we explore the ways companies can leverage the power of black-box fuzzing, a method of dynamic application security testing (DAST), and other security tools to efficiently and effectively secure their software, products, and devices.



## Expanding Attacks

The rate of cyber attacks in 2021 increased 50% to an all-time high. With the cost of a breach skyrocketing to $4.24 Million in 2021, companies need to take control of their products and infrastructure security. To do this, they need to use the same tools that the attackers use to unearth security defects before they can be abused and wielded as a weapon for criminal activity.

## Exploits Lead to Escalation

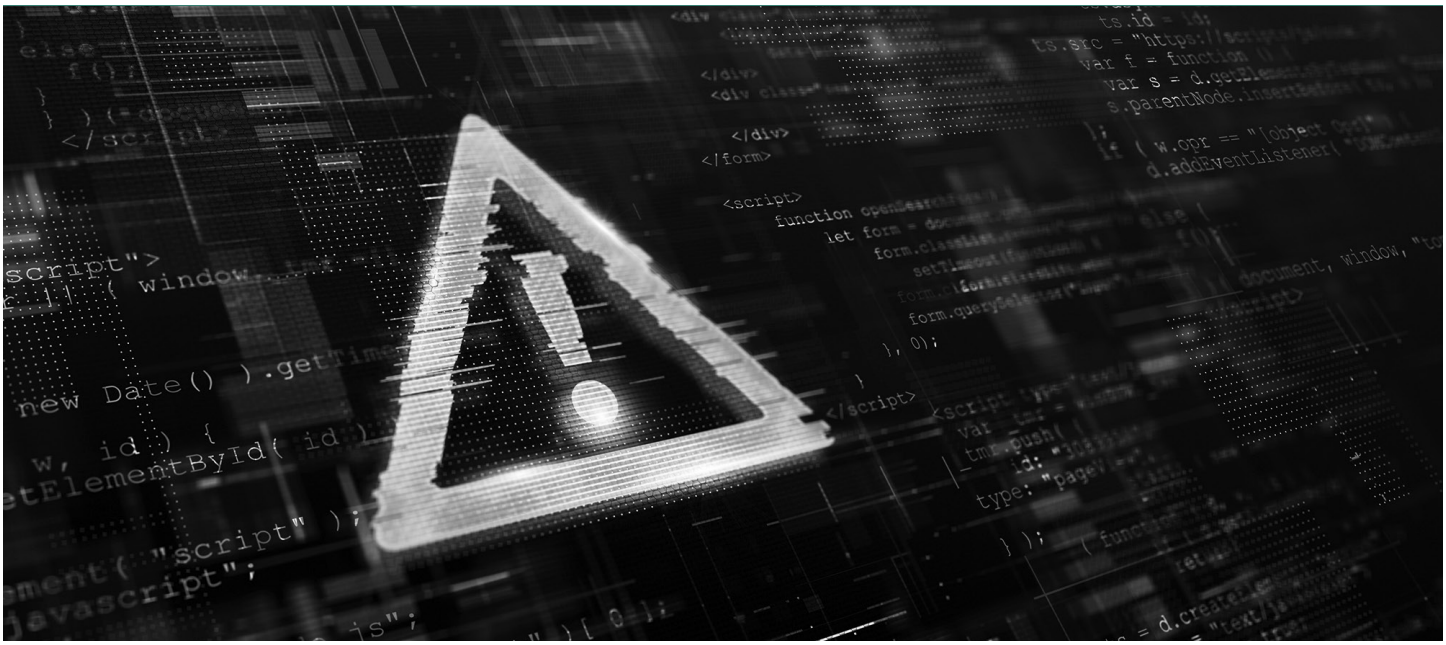
Software exploits do not happen in a bubble. Once a piece of software is exploited, there is the potential for the attacker to escalate privileges to gain increased access and control. Escalated privileges can allow the attacker to monitor other users' usage, exfiltrate sensitive information from target systems, and move laterally throughout the network. This step is rarely the end of the attack, rather it is the beginning - establishing a beachhead for the next phase of compromise.

Once an attacker has some level of access, even if there is nothing on the machine they compromised, it is often used as a pivot point to gain access to protected parts of the network. In many cases, networking rules are less stringent for trusted systems, allowing those that have taken advantage of a security hole in a system to access other connected systems that would otherwise be inaccessible to the outside. This can rapidly escalate into a chain of compromises, with attackers gaining widespread access across the network infrastructure.

***"Software bugs cost the U.S. economy an estimated $59.5 billion each year"***

## Abundance of Targets

When looking at an entire organization, there exist an abundance of options for cybercriminals to target. These targets range from traditional server infrastructure to internet of things (IoT) devices and cloud solutions. Many of these systems may have external interfaces for administration and access to internal networks. This set of potential targets creates a broad threat landscape that administrators need to protect by identifying security gaps and remediating them before the attackers can use them. Attackers are most commonly assumed to target servers and other internet-accessible computers. While the first step for attackers is identifying known vulnerabilities (or low-hanging fruit), sometimes they come up with little to nothing of use. In those instances, attackers have to adopt more uncommon tactics.



## Fuzzing to Find Flaws

Many well-known and established tools for detecting exploitable vulnerabilities can be leveraged for exploitation or prevention, depending on the user. Vulnerability scanning tools are tailored to detect known issues and vulnerabilities that have been published. Exploit frameworks can help launch an attack on or validate an existing vulnerability.

While these tools are suitable for detecting more common issues, some conditions are not as easily detected, or may utilize an obscure protocol that is not supported. In these cases, fuzzers can be used to identify possible flaws. Attackers can used fuzzers when they come up against a hardened infrastructure or device to root out odd programming errors for potential exploitation. Conversely, organizations can employ DAST tools and fuzzers to preempt these efforts and secure their software.

## How Fuzzing Works

A fuzzer can systematically iterate through millions of different input combinations, valid/expected, invalid/unexpected, malformed, semi-malformed, and random, against all of the various input interfaces available. Over time this can exhaust every theoretical combination exposing any programming errors, implementation bugs, or security loopholes. Fuzzing gives insight into more covert vulnerabilities that have not made the known exploit list. The importance of fuzzing grows as resource constrained development teams cut corners to beat deadlines and competitors. Assessing systems, devices, applications, and software for vulnerabilities can be time-consuming, and developers don't always take the time to audit every aspect of a product before release. So they test the most likely scenarios and edge cases, hoping that will be sufficient. This leaves the potential for vulnerabilities to be discovered later, unless a fuzzer is employed.

## Unconventional IOT Targets

Internet of Things (IoT) devices are crucial for a wide array of business functions, from network monitoring to physical access security. These devices are often connected to the internet and have a reputation for having security gaps. This reputation is not unfounded, as over half of all IoT devices are vulnerable to acute cybersecurity attacks. These vulnerabilities exist, even if they are not in published CVEs.

The true range of IoT devices goes well beyond the traditional "things" that many would consider. Operational Technologies (OT), a category of hardware and software used to monitor and control the performance of physical devices, is one area where IoT convenience is needed for the manufacturers of buildings. Manufacturers may have SCADA (supervisory control and data acquisition) systems controlling industrial equipment and manufacturing. There are also environmental control devices


that maintain the heating and cooling of buildings. Attackers targeting these types of systems create real physical-world consequences. They could shut down production lines or make offices unfit for working, thus impacting the company's ability to function or introducing a health and safety hazard.

While many of these systems may, at first glance, appear secure as they have proprietary interfaces for management that are more complex than simply connecting to a web page, they are not immune to attack. Attackers can communicate across a wide variety of protocols and APIs to target systems with obscure and proprietary interfaces. This allows them to target systems that may not have existing known vulnerabilities due to their obscure nature or cash in by corrupting less secure support devices, like IoT. Black box fuzzers test for these types weaknesses, so even lesser-known systems can be protected.

## Cloud Systems

Cloud network adoption is exploding, now accounting for 67% of enterprise infrastructure. These systems can range from platform as a service (PaaS) or infrastructure as a service (IaaS) that operate similar to traditional server infrastructure to software as a service (SaaS) solutions that are hosted applications.

There is a shared responsibility between the host and the customer with cloud computing systems for operating the infrastructure. This can leave a customer blind to a portion of the implementation. Attackers understand this gap and are increasingly pursuing flaws in the infrastructure through fuzzing attacks. These attacks can expose issues in the host systems and the management protocols controlling them. Organizations that proactively test their own cloud systems can discover these gaps before the criminals and find solutions before they get exploited.

## Fuzzing Advantages

Fuzzing is not just a tool for attackers but can also be leveraged by the defender. Using static application security testing (SAST) to identify existing software code defects before attackers do, dynamic application security testing (DAST) uses fuzzing to allow organizations to test deeper by inspecting applications and protocol behavior, protecting the integrity and security of the application or protocols leaving no security questions unanswered. Discovering defects proactively gives companies a distinct advantage over attackers. It gives them time to prioritize and remediate vulnerabilities discovered, dealing with them on their timetable rather than responding to an attack.

## Find the Right Fuzzer

Choosing the right fuzzing solution is crucial for effectively leveraging fuzzing in your organization. Not all fuzzers are created equal. Like DAST tools, Black Box Fuzzers do not require knowledge of a program's source code and are usually a faster way of finding defects without an in-depth understanding of the application being tested.


To be most effective in a commercial environment, a fuzzer needs to be autonomous enough to work with minimal monitoring and flexible enough to adapt to the vast array of technologies in the organization. It needs to produce actionable findings that can be consumed by different teams in the organization to correct flaws and patch potential holes. Be sure the black box fuzzer you choose offers the following characteristics and functionality:

## Automated Operation and Testing

One essential characteristic of a fuzzer for business use is the ability to operate with minimal fuss. Once the fuzzer is configured and launched, it needs to work with minimal oversight and management as it iterates through various interfaces and combinations of inputs. As the search space for complex products can be huge, the actual operation could take days or weeks to complete. During this time, team members need to be able to accomplish other tasks around the organization rather than babysitting the fuzzer.

For a fuzzer to meet this criterion, it also needs to be tracking what it has tested intelligently. This prevents wasted effort on the part of the program and excessive runtime before results are attained. When findings are discovered, they should be automatically tracked to be played back for verification. This helps to eliminate false positives and rapidly validate test results.

## Protocol Flexibility

When selecting a suitable fuzzer for use, you also need to consider the broad range of protocols that might be used. While many off-the-shelf and open source fuzzers support a limited set of standard protocols such as HTTPS, FTP, and SMTP, this only scratches the surface of possibilities. A robust solution needs to accommodate these common internet protocols, support less common protocols, and be able to be customizable for proprietary protocols, application, or hardware.

Not every product will have a publicized or common enough protocol that is readily available to use. This is where a customizable black box fuzzing solution is crucial for carrying out effective testing. Advanced fuzzing solutions can auto-learn the protocol and how it works, allowing testing of secret and proprietary protocols. With this functionality, there are no devices that will be untestable.

## Detailed Reporting

Culminating any fuzzing effort should be actionable reporting. It is essential to know what was discovered so analysts can conduct further testing to confirm the finding and discern how impactful it might be. To adequately confirm a result, the reported information needs to contain enough details that the testing can be repeated. This information is also vital for validating that the flaw has been completely eliminated after remediation.

## Trusted Black Box Fuzzer

Beyond Security's beSTORM is a trusted name in enterprise security fuzzing. It combines the best of both DAST and Black Box Fuzzing test approaches, creating a powerful tool designed to identify security flaws in protocol implementations. beSTORM utilizes intelligent automation to efficiently scan through likely test cases to determine potential vulnerabilities quickly and exhaustively iterating through the existing space to find more hidden flaws.

beSTORM is not restricted to standard interfaces but has a wide range of pre-defined protocols, eliminating complex configuration tasks and allowing teams to get to testing quickly. With beSTORM, organizations can go beyond basic known vulnerability assessment and identify flaws in all varieties of products, even if they are currently undiscovered or unpublished.

Schedule a demo today to discover how beSTORM can help your organization get ahead of attackers.

FORTRA

Fortra.com

**About Fortra**

Fortra is a cybersecurity company like no other. We're creating a simpler, stronger future for our customers. Our trusted experts and portfolio of integrated, scalable solutions bring balance and control to organizations around the world. We're the positive changemakers and your relentless ally to provide peace of mind through every step of your cybersecurity journey. Learn more at fortra.com.