vmware® Carbon Black

WHITEPAPER

Re-designing Linux Security: Do No Harm

Introduction

Cloud is the dominant computing trend of our time, and Linux powers the cloud. Linux has been taking over the world for some time now. If you haven't noticed, it can be found behind the scenes powering web servers, mobile devices, the top 500 supercomputers, watches, thermostats, televisions, and cars. As businesses become more savvy to the benefits of Linux, we see accelerated adoption of the operating system in the form of both migrations of existing applications and greenfield application deployments. Linux is here, we best secure it.

Unfortunately, the approach to securing this growing fleet of production servers has not evolved at the same pace. Your typical security solution is finely tuned to the issues specific to Windows operating systems and then ported over to run on Linux machines.

However, simply making the "Windows approach" to security available on Linux doesn't acknowledge real and significant differences between the two operating systems. Despite the lack of customization to different operating systems, these security tools have been pushed by security vendors and accepted by organizations.

We believe this is due to a neglect of the unique characteristics of highly performant Linux machines, and therefore a lack of effective criteria with which to evaluate Linux security tools.

The goals of this whitepaper are to bring light to the flaws with porting Windows security approaches to Linux, identify unique challenges with securing Linux infrastructure, introduce a list of questions one can use to better evaluate a Linux security offering, and propose a core set of design principles on which strong Linux security offerings can be built.

Flaws in Evaluating Security Tools for Linux

We can define our collective perception of what security is by investigating the questions we ask about the tools we use. To evaluate Windows security tools, two questions rise to the top:

- Does this tool protect my system from viruses and other malware?
- Does this tool support my version of Windows, now and in the future?

These certainly are important questions and highly valuable when assessing Windows security tools. When a tool designed to meet these criteria on Windows is ported to Linux and presented as an equivalent solution, we assume, often implicitly, that the questions we want answered are also equivalent. However, there are several flaws in using these questions to evaluate Linux security tools.

1 Flaw

PROJECTING THE IMPORTANCE OF CHARACTERISTICS IN WINDOWS SOLUTIONS TO LINUX.

The primary question we need to ask to evaluate a tool for Windows is: Does this tool protect my system from viruses and other malware? The reason this question is important to answer for Windows machines is because we know viruses and malware are a significant threat for them. If malware is a significant threat to a system, it makes perfect sense to heavily weigh malware protection in a security solution. However, due to the nature of Linux machines being critical assets that don't regularly leave your environment, malware can be managed without the need for cumbersome tools . Malware certainly exists on Linux, but nowhere near the volume or variety as malware on Windows. 'Bad things' on Linux are almost exclusively fileless, meaning traditional solutions are ineffective and introduce unnecessary performance impact. By accepting a Windows security model on Linux, we implicitly weigh malware protection disproportionately high, even though we may know Linux malware is relatively scarce compared to Windows.

2 Flaw

A SUFFICIENT ANSWER ON WINDOWS IS NOT NECESSARILY SUFFICIENT ON LINUX.

On Windows, we also want to ask: Does this tool support my version of Windows, now and in the future? This question can be answered sufficiently if the security tool releases updates at the same pace as a fairly small selection of Windows versions you might use, typically quarterly or semi-annually. This is a perfectly appropriate answer for Windows because the release schedule is regular and relatively infrequent. With a quarterly release schedule, the tool would almost always be up-to-date. This is not the case with Linux.

There are many different distributions with wide variation and independent release schedules, some of which include new releases every night. Given the pace of Linux development, a quarterly or semi-annual release schedule would result in a security tool that is almost always out of date.

3 Flaw

THE IMPACT OF SECURITY SCANS ARE NOT EQUAL ON WINDOWS AND LINUX.

By promoting the Windows approach to security on Linux, vendors are saying that the impact of running the tool on both environments is acceptable. On Windows, the effects of a security scan range from unnoticeable to somewhat bothersome. Typically, users on Windows desktops are not utilizing all of the machine's resources. A security scan can temporarily borrow some of the unused resources without significantly impacting the productivity of the user. However, Linux is not typically running on desktop machines.

It's powering servers, which have likely been tuned and optimized to run mission-critical applications and workloads. On production servers, unused machine resources are a waste of money and most organizations will do everything they can to maximize resource utilization. With little-to-no unused resources, a security scan will encroach on resources already in use and result in degraded performance of those mission-critical applications. In a fileless world scans have no effect.

Building Better Evaluation Criteria for Linux Security

Given the flaws in applying these questions to Linux security tools, we need to identify a new set of questions we can use to evaluate the suitability of a security tool for Linux. In order to do this, we must first understand the unique security context of Linux and build out questions from that context. Linux security threats tend to be variations on the same theme: exploiting vulnerabilities in trusted software. Vulnerabilities in this context usually refer to oversights or errors in the implementation of software, such as improperly handled error scenarios or insufficient input validation. These errors and oversights can happen anywhere from the low-level Linux kernel to commonly installed system packages to popular application frameworks and plugins. Several properties of these Linux exploits present challenges unrelated to the common malware attacks found on Windows.

Challenge #1:

FRIEND OR FOE

Windows machines face the unique challenge of large amounts of commodity malware, and a variety of fileless based attacks. A large class of Windows malware achieves persistence by installing new software at specific locations. The mere presence of a file known to be installed by Windows malware is a red flag. Fending off commodity malware, and addressing fileless attacks is table stakes on Windows. However, most Linux attacks are fileless and are executed through software that was installed on purpose, albeit for another function, making identification of potential vulnerabilities a different task than merely looking for files. This means that Linux based security tools must be designed to identify and remediate fileless attacks first. We have to identify which versions of software are susceptible to exploit and determine if any of those software versions are installed. For vulnerabilities that haven't yet been identified, we need to be able to monitor and flag unexpected behaviour or use of software. Identifying unexpected behavior doesn't always point to exploitation, so we need to be able to investigate further. Again, if your leveraging legacy tools which aren't designed 'fileless first' those tools

will prove to be ineffective. Questions that can be used to help evaluate security tools against this challenge include:

- Can the tool help detect software installed with known vulnerabilities and exploits?
- Does this tool allow for operational visibility in order to identify and investigate unexpected behavior?

Challenge #2

HIGH-VELOCITY AND BROAD CONFIGURATION SPECTRUM

There are many Linux distributions, and rapid update cadences are common. Security tools must support both the breadth of distributions and keep pace with the rate of change. If not, security becomes a blocker, not an enabler for operations. Additionally, Linux distributions and installed software packages have unique release timelines, some releasing as often as one or more times a day. Keeping track of independent, fast-moving releases of installed software can be problematic for security teams. The problem is multiplied when you take into account the fact Linux machines across a fleet will not necessarily be running the same versions of software or even the same distribution. Questions that can be used to help evaluate security tools against this challenge include:

- Does the tool update frequently enough that it can cover new vulnerabilities and exploits soon after being made public?
- What is the impact of the tool being out of date?
- Can the tool perform well across a fleet of machines with diverse configurations?
- Can the tool clearly report findings from scans across a fleet of machines with diverse configurations?

Challenge #3

LINUX IS DESIGNED TO RUN AT REDLINE

Linux administrators trust Linux for predictability and high performance for critical workloads. In these types of environments high-performance security tooling is a must. Security tools that have a material resource footprint mean breaking workloads and forcing expensive expansion. Linux is more commonly installed on servers than desktop machines, which changes the impact of scanning for vulnerabilities. As discussed in Flaw #3 above, the same security scan may have a relatively low impact on a Windows machine and a potentially high impact on a Linux machine. In addition to the hardware being fully utilized, the "user" of a server is very different than that of a desktop machine. In fact, a Linux server running production applications is likely serving many users at a time, multiplying the impact of a single failure across each user affected. Instead of blindly over-allocating hardware to account for security scans, we should have the power to balance our specific security concerns against the impact to the system. Questions that can be used to help evaluate security tools against this challenge include:

- Does this tool significantly impact the ability for my instances to serve traffic?
- Does the tool give me the opportunity to tune it for the security scenarios I care about?

Discovering Design Principles

Through the process of examining the context around Linux security, we have assembled the following list of questions to help evaluate Linux security offerings:

- Can the tool help detect software installed with known vulnerabilities and exploits?
- Does this tool allow for operational visibility in order to identify and investigate unexpected behavior?
- Does the tool update frequently enough that it can cover new vulnerabilities and exploits soon after being made public?
- What is the impact of the tool being out of date?
- Can the tool perform well across a fleet of machines with diverse configurations?
- Can the tool clearly report findings from scans across a fleet of machines with diverse configurations?
- Does this tool significantly impact the ability for my instances to serve traffic?
- Does the tool give me the opportunity to tune it for the security scenarios I care about?

This should not be considered a comprehensive list of questions for evaluating Linux security tools, but it is a good starting point to help make smarter decisions around the security and performance of your Linux machines. As we reflect on appropriate answers to these questions, we believe that there are four key design principles that all vendors must take into account when building solutions for highly performant systems.

DO NO HARM

First and foremost, solutions must be optimized for performance and stability. As vendors, we need to recognize that security tools that negatively impact a business's ability to make money are at fundamental odds with the reality of doing business.

SPEED UP DELIVERY CADENCE

Today companies are forced to choose between upgrading their operating systems or waiting for their security tools to support the latest OS version.

As vendors, we need to remove this decision all together and drive to 0-day support.

VALUE BREADTH OF COVERAGE OVER DEPTH OF FUNCTIONALITY WITHIN A SPECIFIC DISTRIBUTION

Increasingly we are seeing multiple Linux distributions supported within a single environment. As vendors, we need to ensure that our solutions are able to provide comparable value across a wide variety of distributions. It is not acceptable to focus on a single distribution and not cover our customer's
entire footprint.

GIVE CUSTOMERS OWNERSHIP OF THE RISK/REWARD DIAL

Security needs to work in service of the business, not in spite of it. Security practitioners need to be in tune with their business context so that they can appropriately manage their security solutions while avoiding disruptions to the business. As vendors, we need to build solutions that enable tuning and management of security solutions based on business needs. In order to do this successfully, solutions need to be implemented thoughtfully from the ground up, starting with the architecture.

A Way Forward

Until now, there hasn't been a security tool that sufficiently answers the list of questions we assembled. In fact, there hasn't even been a security vendor that asks these questions. Despite basic flaws in reasoning, the approach of solving for Windows and porting to Linux has been presented by vendors across the security space as a reasonable strategy.

We believe it is time, as vendors, security professionals, and security-conscious organizations, to collectively take a step back and rethink the status quo. To start that conversation, we've uncovered some assumptions about the existing approach and identified how they are flawed. We've also examined the context around Linux security challenges in order to construct a sample set of questions that can be used to more effectively evaluate security offerings. Based on our list of questions, we derived four key design principles that must be present in an effective Linux security offering, namely:

- **DO NO HARM**
- **SPEED UP DELIVERY CADENCE**
- **VALUE BREADTH OF COVERAGE OVER DEPTH OF FUNCTIONALITY WITHIN A SPECIFIC DISTRIBUTION**
- **GIVE CUSTOMERS OWNERSHIP OF THE RISK/REWARD DIAL**

Our time of accepting fundamentally flawed security tools needs to come to an end. We believe the only way to move forward is to take a careful, inquisitive approach. Starting from the ground up, we must first examine the context around what we wish to secure. Within that context, we need to identify driving questions and design principles that fulfill those questions.

Lastly, we need to build and adopt modern security tools that adhere to these design principles. Only then can we enable organizations to effectively secure their ever-growing landscape of Linux machines.

vmware® Carbon Black

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 vmware.com