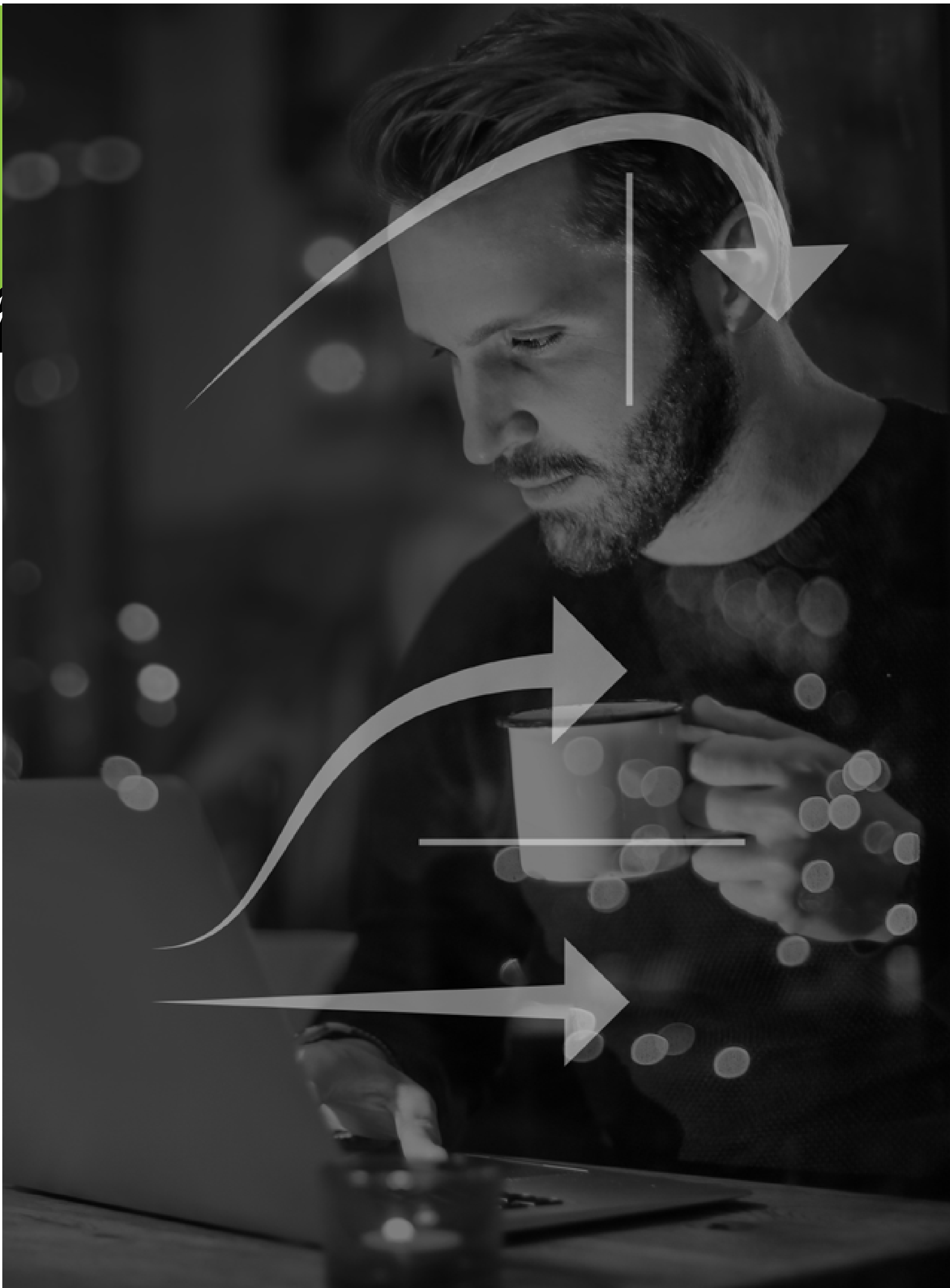
WHITE PAPER

RidgeHAD

Ridge Horizontal Access Detection

RIDGE SECURITY



Background

Broken Access Control and Horizontal Privilege Escalation

Broken Access Control refers to the ability of an end-user, whether through tampering of a URL, cookie, token, or contents of a page, to virtually access data where they shouldn't have access. Broken access controls are commonly encountered critical vulnerabilities. These vulnerabilities rank #5 in OWASP's (Open Web Application Security Project) 2017 Top 10 most critical web application security risks.

The design and management of access controls can be complicated and dynamic. Web applications are constantly evolving, and we find that access control rules become inserted in various locations at different times. It is an insidious challenge to detect flawed access controls by relying on a developer's discretion.

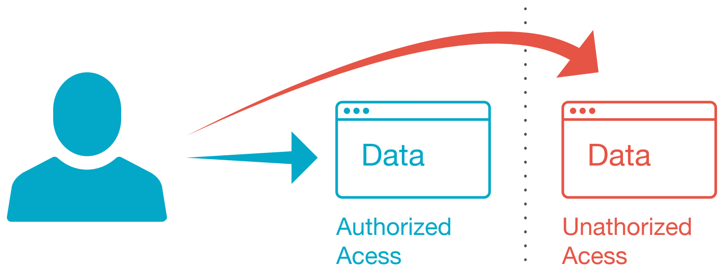
Flawed access controls are exploited for attacks of:

- **Vertical Privilege Escalation**
  Vertical Privilege Escalation refers to the access violation between different levels of user privilege, such as an ordinary user to execute as an administrator user.
- **Horizontal Privilege Escalation**
  The horizontal privilege escalation refers to an unauthorized operation between users at the same level. For example, if an employee should only be able to access their employment and payroll records, but can, in fact, also access the records of other employees, then this is horizontal privilege escalation.

Both vertical and horizontal privilege escalations result in devastating consequences. As the vertical privilege escalation is an imperative step for most sophisticated attacks, its detection is explored extensively and covered by many threat detection systems. The horizontal attack is worth its special treatment as flaws are common in the code, and it is easy to exploit and is blamed for many significant data breaches and financial losses.

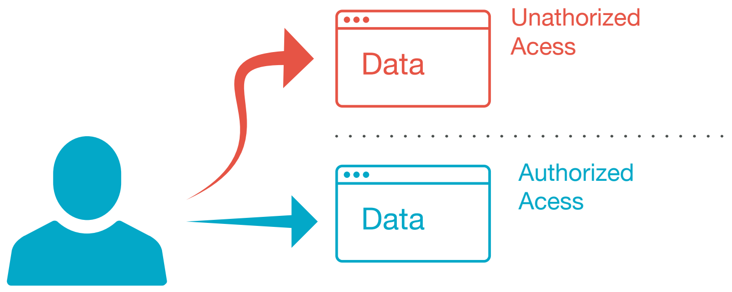
The RidgeHAD is a system that mainly aims to detect and stop horizontal privilege escalation attacks and help organizations to enforce the horizontal access control.

User A—Horizontal Privilege Escalation



User A and User B belong to the same role and have the same permission level. Still, the payroll system only verifies the role that can access the data, without binding it to a specific user identity

User B—Vertical Privilege Escalation



Causes and Consequences

Usually during development, most web application developers deliberately design a strict permission verification for each function. However, as the code simply evolves, and the app nears deployment, the ad hoc collection of rules becomes so unwieldy that it is almost impossible to understand. Once there is an omission, unauthorized access can occur.

Privilege vulnerabilities cause devastating damage from both organizations and users.

Quite a few well-known data breaches in the past blamed flawed privilege protection. For example, Home Depot's breaches in 2014 started when a 3rd party partner's credentials were stolen. The real damage happened when the attacker gained privilege escalation through this stolen account, allowing the attacker access to the same kind of data through the entire organization.

Other common cases are with e-commerce applications, where promotions use coupon or point redemption. If web sites use some form of ID, key, or index as a way to reference a user, but fail to validate the ID belonging to that current user, the attacker can guess these IDs, access stolen data, and steal points, coupons and other information. If this occurs, the website cannot achieve the goal of their promotions, suffers financial losses, and puts their customers at risk.

For end-users, the impact is direct and disastrous. Once their identity, home address, telephone, and other sensitive information leak out to the black market, criminals could do all kinds of damages with it: tempering/deleting accounts and information, even stealing their identities.

The three most common reasons that result in a "horizontal privilege escalation"

1. Web function solely based on "user identity ID". When performing a function, the corresponding data is accessed or manipulated through a unique ID based on a user's identity, such as user ID, account number, mobile phone number, identification number, etc.

2. Web function solely based on "the object ID". When performing a function, the corresponding data is accessed or manipulated through the object ID such as order number, record number etc.

3. Web function solely based on "file name". When performing a function, a file is accessed by using its file name, most commonly seen when a user is uploading a file.

Examples of Horizontal Access Detection

An Email Service Provider—Email Password Reset Vulnerability

This email service provider sent a verification code to a user's mobile phone to reset the user's mailbox password as many web sites do. However, in their access control mechanism, they omitted to verify the binding relationship between the mailbox and the mobile phone number. So an attacker reset the mailbox password by modifying the email address in the request URL and received the verification code with a mobile phone number they specified. With the password reset exploit, an attacker quickly gained access to the user's emails. Nowadays, most users bind their email address with their often used e-commerce websites, via a "Forgot Your Password" function. The attacker then resets the user's passwords on all E-commerce websites with the compromised email address.

A Recruiting Website—unauthorized Access to All CVs

On this website, by traversing the URL request parameters, an attacker obtained other users' resumés, where they found sensitive, personal information like identification, residential address, mobile phone number, email address, education etc.

A Movie Ticket App—unauthorized access to other users' booking orders

After purchasing a ticket online, when a user quits his/her order, the ID shows in the URL parameters as a clear text. In the login state, by traversing the IDs, an attacker got all other user's booking information and the QR code of the tickets.

A Smart Watch Official Website—unauthorized points redemption

This web site provides users with a function to redeem points for goods. However, the access control of this function had a flaw, and an attacker was able to get all the user's points by tampering with the user ID in the URL parameters.

RidgeHAD System

Horizontal privilege escalation is a business logic vulnerability that is difficult to avoid during development. Extensive security testing before deployment is the best tool recommended by security experts to detect the violations and protect access controls.

The access control mechanism should be tested broadly to be sure that there is no way to bypass it. This testing requires a variety of accounts and extensive attempts to access unauthorized content or functions.

Currently, many organizations adopted manual testing; however, manual testing is not only very costly but also not feasible for complex and large-scale applications. Horizontal privilege escalation detection is relatively time-consuming and labor-intensive compared to other tests, which inevitably affects the rapid delivery of application systems. An efficient and reliable automated detection system is keenly needed.

The RidgeHAD is such a system, freeing security testers from heavy manual testing and improve productivity.

Detection Mechanisms

The RidgeHAD uses two detection mechanisms: crawling and proxy.

The **crawling mode**, configured for WEB applications, utilizes Smart Crawler and Brute-Force engines to obtain all URLs and interactive data of the tested applications.

The **proxy mode** is used mainly toward mobile applications. A proxy is configured to capture all URLs and interactive data that is generated by users' from their mobile APPs.

After collecting all data via Crawler or Proxy, RidgeHAD turns on its vulnerability inspection engine, in conjunction with its AI-Powered threat detection engine, discovering hidden vulnerabilities in the horizontal access controls of applications under test. Security teams go through all findings from RidgeHAD to validate. The development team repairs validated vulnerabilities before deploying.

The RidgeHAD is black-box testing. Testers do not need to understand the application implementation details or any particular programming languages. Testing is simple and convenient, and the results are persistent. This frees the testers from the heavy workload of documentation and reporting preparation as required in manual testing.

Technical Advantages

Five core technologies power the RidgeHAD system

- Smart Crawler
- Brute-Force Engine
- Traffic Proxy
- Vulnerability Inspection Engine
- AI-Powered Threat Detection Engine

Smart Web Crawler

The Ridge Security Smart Crawler simulates "click" action, efficiently scraping any reviewable contents on Web Pages, such as Text, Image, Email, Address, Phone number, Search Result… There are two key modules in the Ridge Crawler: the **Control Module** and the **Execution Module**. The Control Module controls crawling strategy, login authentication data, and customizes what to and what not to scrape, and the Execution Module extracts a page's links and manipulates the Document Object Model (DOM), a programming API for HTML and XML documents.

The Ridge Smart Crawler differentiates itself in the following ways

1. High efficiency—checks and filters out duplications, significantly shortening the time to scraping.
2. Rich control strategies to filter out invalid URLs.
3. Support JavaScript interpreter, able to get URLs from JavaScript code.
4. Support extracting links from a flash file.
5. Support crawling through a proxy.
6. Support scanning via authentication.
7. Support granular control of the scanning scope. Users can choose to scan the entire domain, subdomains, or the current directory.
8. Multi-thread design in execution to improve the crawling speed; and, in the meantime, able to flexibly configure the number of execution threads in the running to avoid excessive pressure on a single application by a large number of concurrent sessions.

Brute-Force Engine

The Ridge Security Brute-Force Engine is cloud-based, supporting URL path traversal and brute-forces a web page to locate hidden links. The engine supports a customized brute-force dictionary and automatically identifies URL redirect pages, invalid directories, invalid pages, and pan-links to avoid wasted efforts. It also achieves efficiency with high concurrent sessions support.

Traffic proxy

In the case that the Smart Crawler cannot recognize all parameters or accurately simulate input parameters, the Traffic Proxy is useful. As a middleman between the client and the server, the Traffic Proxy precisely captures all user's input values while the user is doing a standard functional test. It intercepts, stores, and analyzes the original data flow in bi-direction.

The Ridge Security Traffic Proxy fully supports both HTTP and HTTPS protocols. It can be applied to specific targeted applications (specify by web or IP address) while allowing other applications to flow freely without going through the proxy. The configuration is simple and straightforward, with no network knowledge required for admins.

Vulnerability Inspection Engine

The Ridge Security Vulnerability Inspection Engine intelligently identifies the to-be-inspected points in a web request. Requests usually include URL paths, URL parameters, parameters in the request body and header, and key-value in a cookie, etc. while analyzing all possible checkpoints.

The inspection engine, based on multi-user, cross-check technology, utilizes various algorithms, such as fuzzy matching algorithms and similarity algorithms to perform cross-inspection and identify privilege vulnerabilities, and generate test reports.

AI-Powered Threat Detection Engine

The foundation of the AI-Powered threat detection engine is machine learning plus deep learning technologies under a supervised learning model. The AI-powered engine determines whether a web page has horizontal privilege escalation that is based on the algorithm and knowledge map built with expert experience. It then submits the suspected pages to the penetration testing team. The penetration testing team further validates and confirms the findings. In this way, it dramatically improves the productivity of the penetration test and reduces the cost. This method abandons the traditional way of detecting horizontal privilege escalation, based on the return value of different user logins. The AI-powered engine can pinpoint exact pages where privilege vulnerability exists in a much faster and more accurate fashion.

Summary

With an organization's increased security awareness and extensive use of various security systems (such as security development framework, intrusion prevention system, security protection software etc.), common vulnerabilities such as SQL injection, XSS, command execution, and CSRF are predicted to be minimal. However, "vulnerabilities in the code logic" show no signs of decline. It may even become the main battlefield, as the code logic is a product of human thinking, which is error-prone.

Currently, the detection of horizontal privilege escalation is covered solely by manual penetration testing, during which a tester has to manually process all URLs and request parameters to find the problem pages. Manual testing has the following limitations:

1. Require highly skilled professionals who can perform penetration testing as well as have a good understanding of programming logic. Also, manual testing requires a lot of documents and reports preparation. All of the above eventually translate to higher costs and longer times for organizations.

2. Often time, it isn't feasible to conduct a comprehensive test because of software release date, development cost, personnel resources etc. factors.

3. Test quality is inconsistent. In many cases, the tester has to trade off the rigor of the test with the time-to-market.

RidgeHAD, the automated testing system to horizontal privilege escalation, is a viable solution. It's essential to use computers to handle repetitive work and free up testers' time. It helps organizations reduce 2 or 3 man-week work to 1.5 computer-hour. Moreover, RidgeHAD is highly portable. It can perform various applications' tests in an organization, as it is agnostic to the program's language.

Notes

Company Profile

Ridge Security is transforming Security Validation with automated intelligent systems modeled using the techniques utilized by literally millions of hackers that penetrate systems. When deployed within a system, Ridge Security tools are relentless in their quest to locate, exploit, and document their findings. They work within a defined scope and instantly replicate to address highly complex structures. Ridge Security enables enterprise and web application teams, ISVs, governments, education, anyone responsible for ensuring software security to affordably and efficiently test their systems.

Ridge Security Technology Inc.
2010 El Camino Real PMB 3017
Santa Clara, CA 95050
www.ridgesecurity.ai