USF | Muma Case Review
A publication of the Muma College of Business | University of South Florida

Volume 1 Number 4 3 SEPTEMBER 2016

GRANDON GILL, JONI JONES

MULTI-FACTOR AUTHENTICATION AT JAGGED PEAK[1]

Would it make sense to implement multi-factor authentication? If we do, how can we minimize the costs and disruption it might cause?

Jeffrey Stiles pondered these seemingly straightforward questions. As IT Director of *Jagged Peak, Inc.*, a developer of e-commerce solutions located in the Tampa Bay region of Florida, it would be his responsibility to oversee the implementation of security measures that went beyond the existing user name and password currently required for each user. Recent events suggested that a move towards increased security might be inevitable. In just the past year, highly publicized security failures at the U.S. Department of Defense, major healthcare providers and large companies, such as Sony and JP Morgan Chase, had made executives acutely aware of the adverse consequences of IT system vulnerabilities. In fact, a study of business risk managers conducted in 2014 found that 69% of all businesses had experienced some level of hacking in the previous year.

The nature of *Jagged Peak's* business made the security of its systems a particular concern. The company, which had grown rapidly over the years, reporting over $61 million in revenue in 2014, provided its customers with software that supported web-based ordering, fulfillment and logistics activities, built around a philosophy of "buy anywhere, fulfill anywhere, return anywhere". To support these activities, the company's Edge platform needed to handle a variety of payment types, including gift cards (a recent target of hackers), as well as sensitive personal identifying information (PII). Compounding the security challenge: each customer ran its own instance of the Edge platform, and managed its own users.

When only a single customer was being considered, the addition of further layers of security to authenticate uses was an eminently solvable problem. A variety of alternative approaches existed, including the use of various biometrics, key fobs that provided codes the user could enter, personalized security questions, and many others. The problem was that where multiple customers were involved, it was much more difficult to form a consensus. One customer might object to biometrics because it users lacked the necessary hardware. Another might object to security keys as being too costly, easily stolen or lost. Personalized questions might be considered too failure-prone by some customers. Furthermore, it was not clear that adding additional layers of authentication would necessarily be the most cost-effective way to reduce vulnerability. Other approaches, such as user training might provide greater value.

Even if Stiles decided to proceed with additional authentication, questions remained. Mandatory or a free/added-cost option? Developed in house or by a third party? Used for internal systems only, customer platforms only, or both? Implementation could not begin until these broad questions were answered.

[1] Copyright © 2016, *T. Grandon Gill*. This case was prepared for the purpose of class discussion, and not to illustrate the effective or ineffective handling of an administrative situation. This case is published under a Creative Commons BY-NC license. Permission is granted to copy and distribute this case for non-commercial purposes, in both printed and electronic formats.

Editor: Matt Mullarkey

Jagged Peak

Jagged Peak, Inc. (OTCBB: JGPK) was founded in July, 2000 through a merger of two companies--an emerging e-commerce software company that focused on the development of real-time, web-based order management applications and the other--a stalwart marketing and logistics firm that had fine-tuned fulfillment optimization.

Background

Jagged Peak President and Co-founder Paul Demirdjian's vision for the company was inspired by his dealings with an online computer electronics retailer. He realized that there was a need for a transparent order fulfillment process after he purchased a new laptop online and waited for almost a month for its arrival. He contacted the retailer's customer service call center to investigate the delivery status of his order only to be informed that they could not locate the order, or even determine if the product was in stock. This was quite surprising as he felt the successful delivery of products to customers should have been among the top priorities of the company. He realized then that the retailer's web portal could not communicate with their order management system for inventory information and this gap inspired him to create a solution.

Demirdjian envisaged a software application that would allow companies to merge different software applications to create a comprehensive supply chain software platform that could be used as a single multi-channel system.

To fulfill his vision, Demirdjian created IBIS (Internet Business Integrated Solutions) in 1999 to serve as a custom web development services provider. He believed that web sites would move away from simply being informational presentation layers to becoming business layers that allowed companies to more closely interact with their customers (see Exhibit 1). Working from home, he built *Jagged Peak's* flagship software application known as EDGE®.

Jagged Peak established itself as a pioneer in the development of e-commerce solutions and supply chain services using EDGE®. It provided demand and supply chain management, CRM execution and e-Fulfillment solutions and services to a diverse group of companies. *Jagged Peak* set up its first office in Tampa, FL providing e-commerce solutions to companies and located its fulfillment center in St. Petersburg.

As it continued to grow at a rapid rate, *Jagged Peak* served an increasing number of global clients, providing web based applications in multiple industry segments including financial services, insurance, pharmaceutical, travel and tourism, automotive, manufacturing, and consumer goods such as Tag Heuer, Nabisco U.S., Pfizer Pharmaceuticals and South African Airlines. By spring 2015, the company had well over 100 employees, and according to its most recent 10K filing with the SEC, had 2014 revenues in excess of $60 million.

Technology Offerings

The core technology developed by *Jagged Peak* was EDGE® (Enterprise Dynamic Global Engine). This flagship product served as a web-based software application that enabled companies to control and coordinate multi-channel orders, catalogs, multi-warehouse inventories, and fulfillment across multiple customers, suppliers, employees, and partners in real-time. EDGE® and its related tools also included functionality that enabled clients to: build and operate custom branded portals such as e-commerce, implement incentive and rebate programs, provide customer service, repair and reverse logistics, manage marketing materials, and automate other business processes.

The company also provided outsourced e-commerce and supply chain solutions that gave manufacturers, distributors, and consumer brand companies the ability to establish and operate a direct to customer online business that was integrated with their offline sales channels. FlexNet was a flexible and configurable warehouse network that was fully integrated with the EDGE® platform, and employed complex demand-management rules to route orders to the optimal fulfillment location, thereby improving customer delivery service and reducing supply chain costs. The company's Chief Sales and Marketing Officer, Vincent Fabrizzi, stated:

> Using FlexNet Advanced Logistics, we are able to get the products our customers sell online closer to their customers on the ground. We do this by configuring a custom warehouse network in the markets our clients need them in order to better serve their customer base.

The design of the EDGE® application contained modules logically built around the business process components of e-commerce solutions which provided users with a perceptive and customized control interface to manage their online businesses (see Exhibit 1). Its collaboration with a company's existing systems extended the value of technological investments as it took on the role of a global order management system. EDGE® had a built-in integration layer using the Java Platform Enterprise Edition (J2EE) architecture, allowing it to connect seamlessly with a client's back-end Enterprise Resource Planning (ERP) system.

Jagged Peak's most integrated offering was TotalCommerce®. As illustrated in Exhibit 2, it provided all the capabilities required for a customer to implement a complete global e-commerce and order fulfillment system. From a practical standpoint, customers adopting this service represented one end of *Jagged Peak's* spectrum of offerings—where it provided a complete and self-contained managed IT solution. At the other extreme of the spectrum were those customers to whom *Jagged Peak* supplied a limited set of modules and who often managed their own IT without substantial reliance on *Jagged Peak's* full range of capabilities. Potentially, the impact of any identity authentication system could be quite different for these two classes of clients.

User Authentication

User authentication was an important element within the broader area of identity and access management. It specifically referred to the activities or processes involved in determining that a particular user was "authentic". In other words, that the user's true identity was confirmed. Establishing a confirmed identity was critical to any secure system because a variety of privileges—such as access to confidential data or the computer's operating system—were typically tied to an individual's identity. If an intruder was able to acquire those privileges, the entire integrity of the system could be compromised and substantial data, reputational and economic damages could be incurred. Where large firms were involved, the costs of unauthorized intrusions could potentially be in the hundreds of millions of dollars, or even more.

At the conceptual level, user authentication was often described as involving one or more of three key factors. These factors were:

- Something you know
- Something you have
- Something you are

Something You Know

Authentication through “something you know” was the most widely used form of authentication. By far the most common implementation of this type of authentication was the familiar user name and password combination. Although this form of authentication could be, in theory, quite secure, it was vulnerable to a number of potential threats. Three of the most significant of these were password cracking, password theft and password misappropriation.

Password Cracking

Password cracking was the process of figuring out a user’s password. Cracking could be accomplished in a number of different ways. Individuals that knew the user, or could acquire information about the user, might be able to crack a password through intelligent guesswork. There were also automated methods that could be employed. To understand these, we need to consider how passwords would typically be implemented.

On secure systems, passwords would almost never be stored as plain text—that would create a huge security vulnerability. Instead, they would normally be transformed into a coded value through an algorithmic process referred to as hashing. When a user logged in, the password he or she entered was then put through the same process. If the value produced matched the stored code, then the correct password had been entered. If not, the login was rejected. Because the hashing process was effectively one way—you could easily generate the code from a password, but you could not reproduce the password from the code—storing the codes limited the vulnerability.

Limiting vulnerability, however, was not the same as eliminating it. For example, if an intruder gained access to the file of password codes, he or she might copy it then, and using a dictionary file, try putting every word in the dictionary through the hashing process. If *any* user happened to use a regular word as his or her password, it would produce a code that matched the code for that user in the copied password file. The intruder could then login using that user’s identity. Another approach might involve the default passwords that were established for many software applications, such as databases and network router interfaces. Although a security conscious user would always change such default passwords immediately, many users did not. Moreover, these passwords for a variety of systems were widely available on the Internet. As a consequence, automated spiders could seek out systems and try default passwords that have not been changed; so eventually they would find them.

A variety of system-enforced techniques have been developed to address the password cracking threat. These included requiring strong passwords—e.g., requiring a minimum length, a mixture of upper and lower case letters, use of numbers and symbols, and so forth. Systems may also require that passwords be changed at specific intervals, and may compare new password hash codes with prior password codes to ensure that the password was not being reused. Certain text, such as the user’s name, may not be permitted. Very commonly, a limit was placed on the number of times the same user could attempt to log in. This made it much harder for brute force methods to succeed. All of these methods tended to defeat the ability of cracking through brute force. They may also increase vulnerability to other threats, such as password theft. For example, when passwords became too complicated, users may write them down and keep them in their desks—increasing the vulnerability to theft.

Password Theft

As was the case with cracking, passwords could be stolen in a variety of ways, some automated and some not. One of the most common ways of acquiring a user’s password was watching them (or a video recording of them) as they typed it in. At ATM machines, for example, identity thieves have mounted cameras directed at the keyboard as a means of acquiring the PINs of unsuspecting bank clients. Other non-technological ways to steal passwords could involve searching a user's workspace (or trash) for a password list or coercion (e.g., threatening a user with injury or death if he or she does not relinquish the password to a system).

There were also many technology-enabled approaches to password theft. Since many users employ the same password on multiple systems—despite being told not to do so—if an individual could be convinced to establish a user ID and password on a system in order to log in, that same pair could be tried on many other systems. A user might also be tricked into installing malware on his or her system. That malware could include a key logger that records the user's every keystroke and sends it to the criminal's computer where passwords can be extracted. Using a packet sniffer on wireless networks could be an effective tool for detecting any passwords that were sent unencrypted.

Password Misappropriation

Password misappropriation could occur when a user voluntarily supplied his or her user name and password to another user. This could be done as matter of convenience, for an individual that did not have a user ID. It might also be done to avoid per-user licensing fees. It might even be done as a matter of policy (e.g., a group of instructors needed to use the same system in a classroom). The vulnerability misappropriation created was the loss of control over information and privileges. It could also reduce the organization's ability to determine the individual's involved in incidents after the fact. And, of course, limiting the damage it could cause depended entirely on the user's judgment of the trustworthiness of the individual being allowed unauthorized access.

Variations on "Something You Know"

In addition to passwords, there were a variety of other authentication techniques that fell under the "something you know" category. Some organizations required a PIN in addition to a password. The advantage of this approach was twofold: it reduced vulnerability to cracking and it provided an alternative approach to authentication that could be employed in contexts where a strong password would be cumbersome or impractical (e.g., an ATM keyboard). Another variation on a traditional password involved the use of gestures, such as drawing a pattern on a grid of dots that could be used to unlock some cell phones.

Other approaches involved acquisition of personal information. For example, some sites acquired a set of security questions from the user that could be randomly posed. There were also sites that generated questions from an individual's publicly available background and credit history; e.g., these were used in some online testing settings for identity verification. The advantage of this approach was that it would be virtually impossible for an imposter to prepare answers in advance. Unfortunately, it may also be difficult for authentic users to respond correctly, something that could apply to security questions in general.

Something You Have

A second category of authentication involved some type of object possessed by the user. There were a variety of examples of this approach:

- An identification card, such as a passport or driver's license, which contained coding that could be read electronically. This approach was often used at kiosks in airports where boarding passes were issued without a human agent.

- A card, such as a debit or credit card, with identifying information on a magnetic strip or an embedded chip.

- A one-time password (OTP) device that automatically generated single-use passwords, typically based on an algorithm that used a key shared only by the user and the system. Passwords may be generated in sequence or synchronized by time.

- A security certificate used for public key encryption, which may itself be embedded in a device.

- A USB device, often referred to as a token, containing information that uniquely identified the user.

- A specific piece of network hardware. Because every device manufactured had a unique identifier assigned by the manufacturer, known as a MAC address, presence of the device would signal that it was coming from the user's hardware.

- An email account or mobile phone under the control of the user. Normally, the user was authenticated through responding to a text or email message, a process that was sometimes characterized as challenge-response.

Generally speaking, the greatest vulnerabilities of authentication through something you had were the twin risks of theft and counterfeiting. As an example of the latter, the information on the magnetic strip of debit and credit cards was easy to duplicate. For this reason, many regions of the world (including Europe) only issued cards with built-in chips.

Something You Are

Authentication through "something you are" was dependent upon unique characteristics of the user, commonly referred to as biometrics. There were numerous examples of such characteristics, which could broadly be broken down into biological and behavioral traits. Examples of the former noted in a 2014 report on user authentication by the Gartner Group (Allan, 2014) included:

- Facial recognition

- Use of fingerprints and palm prints

- Retinal and iris scans

- Vein patterns on various locations of the body

- Voiceprints (falling into both biological and behavior categories)

Examples of behavior traits that uniquely identify the individual included:

- Typing patterns

- Signatures

- Mouse and swipe dynamics

While biometrics had the advantage of being highly personalized and difficult to steal or replicate, they had drawbacks as well. Some were intrusive and may have required highly specialized hardware (e.g., retinal scans). Others were subject to substantial inaccuracy (e.g., facial recognition), or may be hard to acquire for some individuals (e.g., some people had fingerprints that were very faint).

Multi-Factor Authentication

Given that every form of authentication had its own unique set of strengths and weaknesses, applications requiring a high level of security often reduced vulnerability by requiring more than one authentication approach. Where more than one factor was used to authenticate a user, the process was referred to as multi-factor authentication. Examples of two factor authentication in day to day living were very common, such as:

- An ATM machine would not work unless you supplied both your debit card (something you have) and typed in your PIN (something you know). In Europe and, increasingly in the U.S., users may be expected to type in a PIN when presenting a credit card to make a charge.

- To enter Disney World, you needed to both produce a ticket (something you have) and present your fingerprint (something you are), a tool used to prevent identity misallocation (e.g., an annual pass holder “lending” his or her pass to a guest visiting from out-of-state).

- When a customer made an online transaction using a credit card number (classified as something you know, as opposed to the card itself—which would be something you have) a confirming email was sent to the account address (something you have).

Three factor authentication was also used in situations requiring high security. For example, when travelling internationally, you may be asked to supply a passport (something you have), fingerprints (something you are) and you may be asked specific questions by the customs official (something you know).

Logically it made sense that confidence in correct authentication could be increased both by employing multi-factor approaches and by increasing the stringency of a particular factor (e.g., requiring both facial recognition and a signature in order to cash a check at a bank). This increased confidence came at a price, however. Biometric recognition hardware could be costly as were devices that proved difficult to counterfeit. False negatives, such as a user being locked out because he or she forgot a strong password, could be expensive to remedy. Furthermore, obsessive focus on user authentication could lead an organization to fail to recognize other threats to security that may pose a greater risk. For example, a disgruntled employee would authenticate every bit as accurately as a loyal one. Software back doors, installed for the purposes of maintenance or by malicious programmers, were put in place specifically because they bypassed authentication. With a limited security budget, the threat of user authentication failure needed to be weighed against other security priorities to determine the most cost-effective solution.

Authentication at Jagged Peak

As a provider of hosted e-commerce solutions, security was necessarily a top priority at *Jagged Peak*. For example, all transactions were conducted over a secure connection and all user accounts were password protected. Customers could also limit user logins to specific IP (internet protocol) addresses, which provided a level of assurance that they were coming from specific systems. A set of pre-defined user classes was used to control what information each user could access; each user belonged to one and only one of these classes.

Existing System

The nature of its business model, however, meant that individual customers needed to manage their own users (typically customer service representatives), who could number in the thousands. Given the variety of its customer base, a one-size-fits-all password policy did not make sense. Instead, each customer could configure its own password policies, with the ability to set values for important parameters such as minimum password length and expiration period. A screen capture used by the customer to establish password policy is presented as Exhibit 3, with lockout policy options being presented in Exhibit 4.

Recent Customer Requests

Perhaps as a consequence of recent highly publicized security failures in the government and major banks, some customers had expressed interest in going beyond single factor authentication. As Stiles put it, people "have begun circling the wagons".

One recent customer, for example, had expressed concerns about IP spoofing—a technique in which an intruder modified Internet data packets to create the appearance of coming from a known IP address. The customer had requested some form of two factor authentication. After studying the matter carefully, the IT staff at *Jagged Peak* had concluded that changes to such a fundamental security process could not be implemented haphazardly, but required careful planning. Instead, they negotiated a compromise solution with the client. That involved requiring user logins being conducted through an encrypted virtual private network (VPN). This addressed the concerns about IP spoofing, but added to the complexity of the login process. Stiles doubted that it would prove to be the best solution for all its customers. For example, where many of a customer's service representatives worked from home, IP spoofing tended to be less of a concern since IP addresses were likely to be dynamically assigned by the user's Internet service provider (ISP), and were therefore not particularly useful to authentication purposes anyway. It seemed likely that if that user's system login was compromised, for example through a hidden malware key logger, their VPN login could easily be compromised at the same time.

Stiles reported a request from another customer:

> One recent example was for one of our Health Care clients that had extraordinary requirements for handling PII [Personally Identifying Information] and controlling access. They also wanted our application to provide multi-factor integration as a more secured authentication procedure when using the portal.

Challenges of Multi-Factor Authentication

The diversity of users supported by *Jagged Peak's* organizational customers made it unlikely that any "optimal" authentication solution would be found. Among the obstacles:

- Different customers may have users with vastly different circumstances. For example, not all could be assumed to have a mobile phone, or even an external phone line.

- Some authentication techniques, such as those involving biometrics, might involve substantial costs when implemented across an entire organization. Not all customer organizations would perceive that the benefits would justify these costs.

- Different customer organizations may prioritize threats differently, in which case different vulnerabilities would be of concern. For example, an organization in which all users were physically connected to a network located in a secure facility may be unconcerned about identifying compromises resulting from theft of a user's system. The level of concern would (or should) be much higher for a customer whose users were teleworkers using laptops.

Examples of alternative authentication approaches and the challenges presented by each are listed in Exhibit 5. The final row of the table proposed offering options. But Stiles wondered if this was even feasible, and what it would cost.

Making the Multi-Factor Decision

As the summer of 2015 came around, it had become clear that a decision on how to move forward needed to be made. From Stiles' perspective, the first step in making the decision would be to determine what specific authentication technique, if any, would be employed. He expressed discomfort with the possibility of trying to implement more than one—offering the customers options—unless a customer was willing to underwrite the implementation of additional approaches. But even that assumption might require further consideration.

Centralized vs. Distributed Authentication

Once the decision on authentication approach (or approaches) had been made, there were a number of other decisions that needed to be made prior to implementation. One of these was whether the secondary authentication should be implemented as a centralized system, managed by *Jagged Peak*, or whether it should be embedded within the customers' installation (as was the case for the password system). The advantages of centralized authentication were largely derived from the reduced administrative overhead associated with a single point of management. That came with the risk of establishing a single point of failure: in the event the centralized authentication system went down, every customer's users would be unable to login. This would be a very serious problem for a system that was designed to take and fulfill orders from consumers and businesses. You could almost meter the lost sales with every minute of down time.

By implementing authentication within each customer's installation, the single point of failure was limited to a particular customer. It would also be the easiest to implement, since it mirrored the password system and was consistent with the general architecture provided by *Jagged Peak*. It might be slightly less secure, however, since it would likely be running on the same network as the customer's other systems, rather than on a centralized authentication network provided by *Jagged Peak*. It could also be more difficult to administer, as customers would likely seek to configure security differently (as they did with password policy).

User Configuration Policy

In parallel with the centralized-distributed decision, Stiles recognized that a number of policy decisions needed to be made regarding what accounts required multi-factor authentication. One approach would be to require the same authentication for all users. The only exception to this policy would be a tiny number of super-users, who could get into the system without an additional factor. This exception was necessary because if the entire authentication system failed, everyone would be locked out and repairs could not be made if programmers could not gain access.

An alternative would be to attach authentication policy to user classes. This would make sense from a practical standpoint. For example, the risk reduction benefits of multi-factor authenticating users who operated from a hard-wired system in a secure facility would be far less than the corresponding benefits for users who were teleworking using their laptops. The potential problem with making the policy flexible in this manner was that it would add to the complexity of the authentication system. In addition, *Jagged Peak's* existing user classes would not necessarily be a good fit with differing authentication needs. Customers might begin demanding that additional new classes be added. That could, in turn, lead to additional rework on existing systems.

Paying for Additional Authentication

Without knowing what authentication technology (or technologies) would be employed, it was impossible to provide an accurate estimate of implementation costs. Alfred Sarkis, the Associate VP of Research and Development described the range as follows:

> One vendor was asking $5000 fee plus $2000 monthly for internet based services. Application development internal estimates are roughly $9500.

Even once the implementation costs were accurately estimated, the question of who would pay those costs would remain. From a marketing standpoint there was a risk associated with passing such costs on to customers. Your customers generally did not like to hear that they were paying to make your system more secure; it seemed that acceptable security was something that you should be providing without charging extra for it. On the other hand, some customers had already made specific requests for additional authentication. Wouldn't it be fair if they underwrote some of the costs if capabilities were to be added? But would it be fair to later give those capabilities away to new customers as part of the base package?

Shortly, Stiles was going to attend a meeting with *Jagged Peak's* CEO, Paul Demirdjian, and other company executives regarding the authentication decision. At that time, he would be expected to make his recommendation. The time had come to move the decision to the front burner.

References

Allan, A. (2014). A taxonomy of user authentication methods. *Gartner Group*, Document 2697317, 1 April.

Acknowledgements

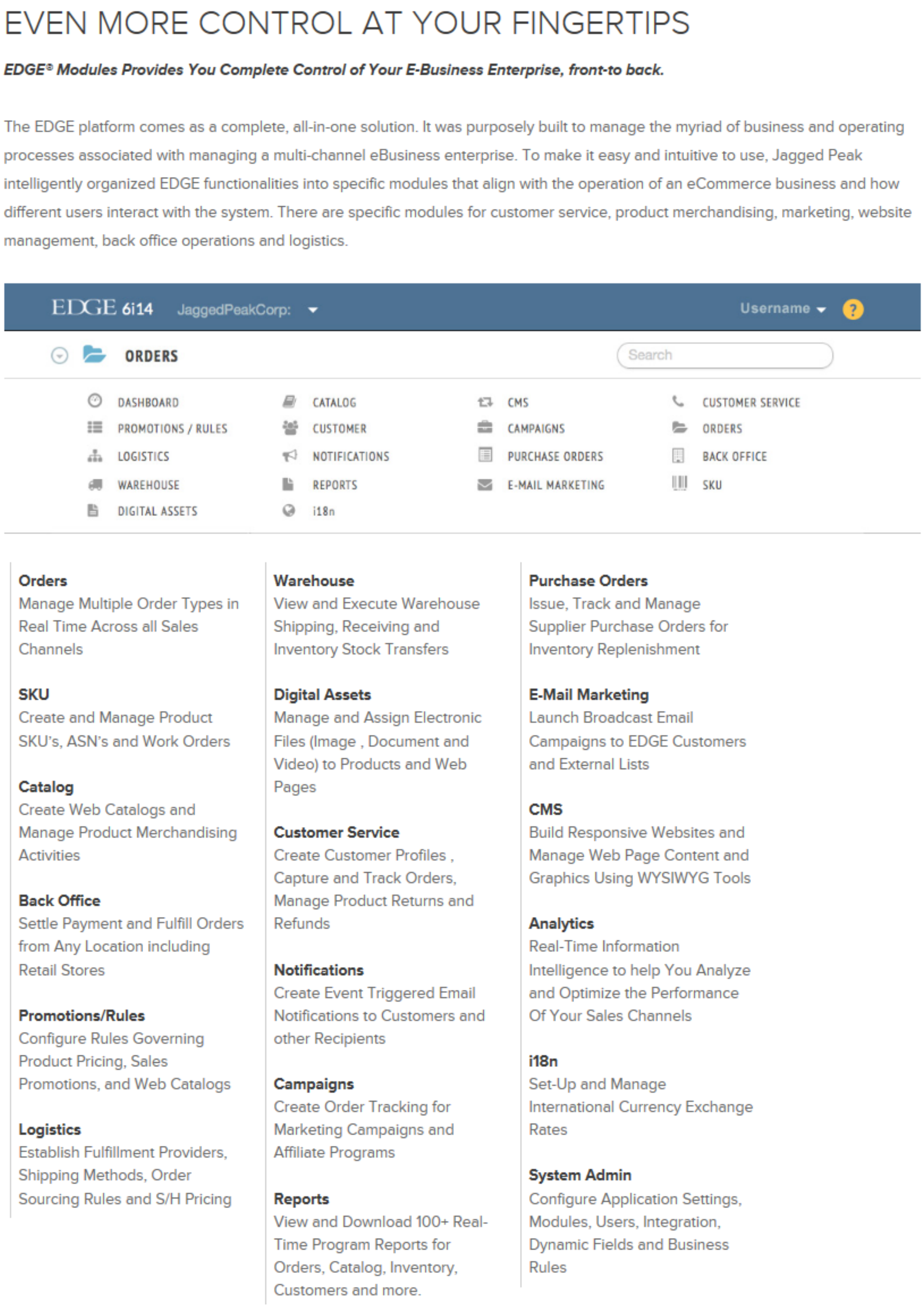
This case study is based upon work supported by the National Science Foundation under Grant No. 1418711.

Biographies

Grandon Gill is a Professor in the Information Systems and Decision Sciences department at the *University of South Florida*, where he also serves as the Academic Director of the Doctor of Business Administration program at the *Muma College of Business*. He holds a doctorate in Management Information Systems from Harvard Business School, where he also received his M.B.A. His principal research areas are the impacts of complexity on decision-making, the diffusion of academic research findings and applying the case method to STEM education. He is currently Editor-in-Chief of *Informing Science: The International Journal of an Emerging Transdiscipline* and an Editor of the *Journal of IT Education.* He is the founding editor of *Journal of IT Education: Discussion Cases*.

Joni Jones is an Associate Professor in the Information Systems Decision Sciences Department at the *University of South Florida*. She teaches graduate and undergraduate courses in systems analysis and design, business honors professional development, and research methods. She previously taught introductory courses in computing as well as courses in C#, managerial statistics, business system application and design, and software applications. Her research interests include electronic commerce, variable pricing mechanisms such as information and prediction markets, and social network use in organizations.

Exhibit 1: EDGE Modules



EVEN MORE CONTROL AT YOUR FINGERTIPS

EDGE® Modules Provides You Complete Control of Your E-Business Enterprise, front-to back.

The EDGE platform comes as a complete, all-in-one solution. It was purposely built to manage the myriad of business and operating processes associated with managing a multi-channel eBusiness enterprise. To make it easy and intuitive to use, Jagged Peak intelligently organized EDGE functionalities into specific modules that align with the operation of an eCommerce business and how different users interact with the system. There are specific modules for customer service, product merchandising, marketing, website management, back office operations and logistics.

Orders
Manage Multiple Order Types in Real Time Across all Sales Channels

SKU
Create and Manage Product SKU's, ASN's and Work Orders

Catalog
Create Web Catalogs and Manage Product Merchandising Activities

Back Office
Settle Payment and Fulfill Orders from Any Location including Retail Stores

Promotions/Rules
Configure Rules Governing Product Pricing, Sales Promotions, and Web Catalogs

Logistics
Establish Fulfillment Providers, Shipping Methods, Order Sourcing Rules and S/H Pricing

Warehouse
View and Execute Warehouse Shipping, Receiving and Inventory Stock Transfers

Digital Assets
Manage and Assign Electronic Files (Image , Document and Video) to Products and Web Pages

Customer Service
Create Customer Profiles , Capture and Track Orders, Manage Product Returns and Refunds

Notifications
Create Event Triggered Email Notifications to Customers and other Recipients

Campaigns
Create Order Tracking for Marketing Campaigns and Affiliate Programs

Reports
View and Download 100+ Real-Time Program Reports for Orders, Catalog, Inventory, Customers and more.

Purchase Orders
Issue, Track and Manage Supplier Purchase Orders for Inventory Replenishment

E-Mail Marketing
Launch Broadcast Email Campaigns to EDGE Customers and External Lists

CMS
Build Responsive Websites and Manage Web Page Content and Graphics Using WYSIWYG Tools

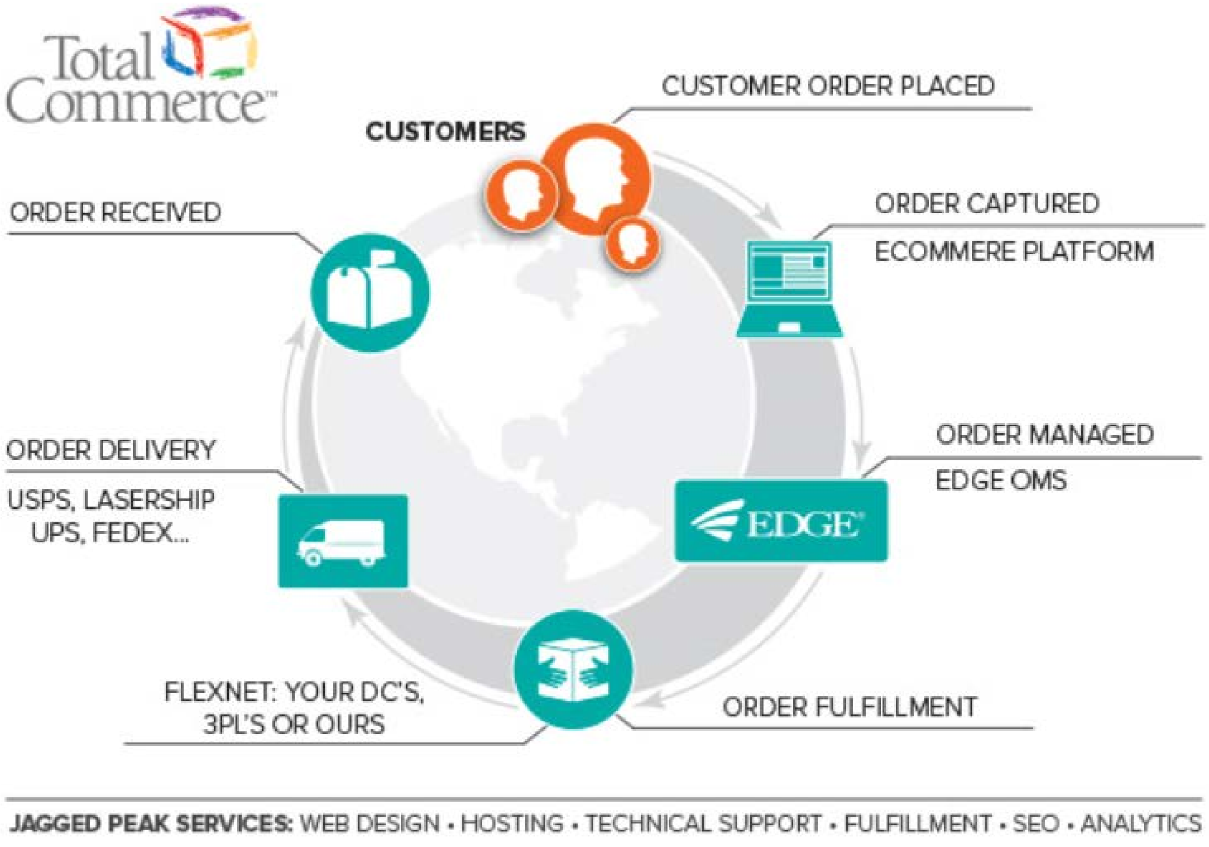
Analytics
Real-Time Information Intelligence to help You Analyze and Optimize the Performance Of Your Sales Channels

i18n
Set-Up and Manage International Currency Exchange Rates

System Admin
Configure Application Settings, Modules, Users, Integration, Dynamic Fields and Business Rules

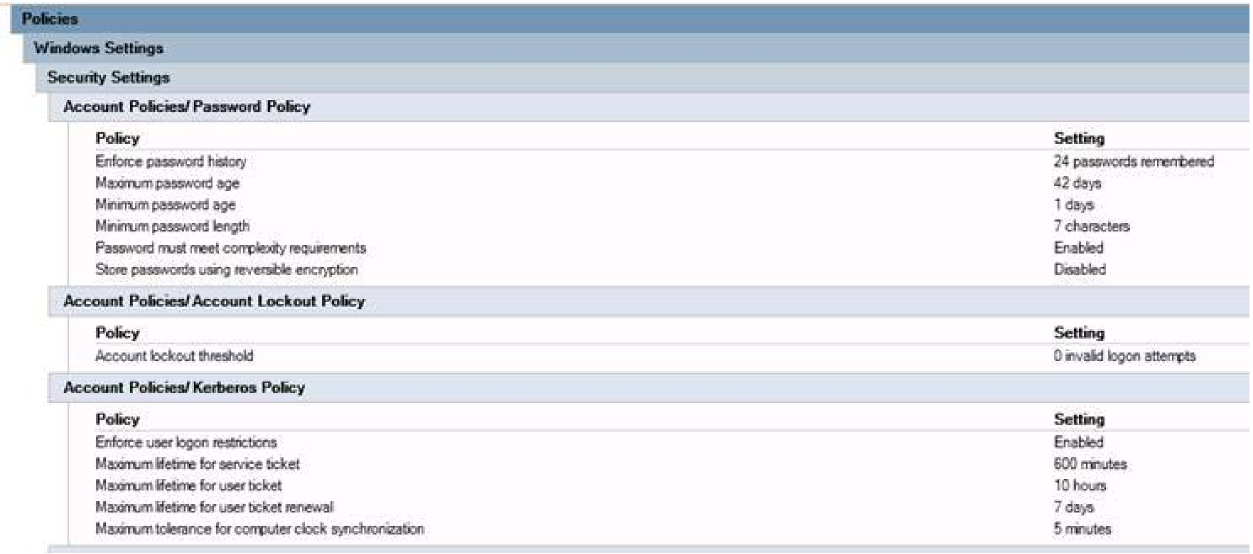
Source: Jagged Peak

Exhibit 2: Jagged Peak's TotalCommerce® Solution


Total Commerce™
CUSTOMERS
CUSTOMER ORDER PLACED
ORDER CAPTURED
ECOMMERE PLATFORM
ORDER MANAGED
EDGE OMS
EDGE
ORDER FULFILLMENT
FLEXNET: YOUR DC'S, 3PL'S OR OURS
ORDER DELIVERY
USPS, LASERSHIP UPS, FEDEX...
ORDER RECEIVED
JAGGED PEAK SERVICES: WEB DESIGN • HOSTING • TECHNICAL SUPPORT • FULFILLMENT • SEO • ANALYTICS

Source: Jagged Peak

Exhibit 3: Jagged Peak Password Policy



Policies

Windows Settings

Security Settings

Account Policies/ Password Policy

Policy	Setting
Enforce password history	24 passwords remembered
Maximum password age	42 days
Minimum password age	1 days
Minimum password length	7 characters
Password must meet complexity requirements	Enabled
Store passwords using reversible encryption	Disabled

Account Policies/ Account Lockout Policy

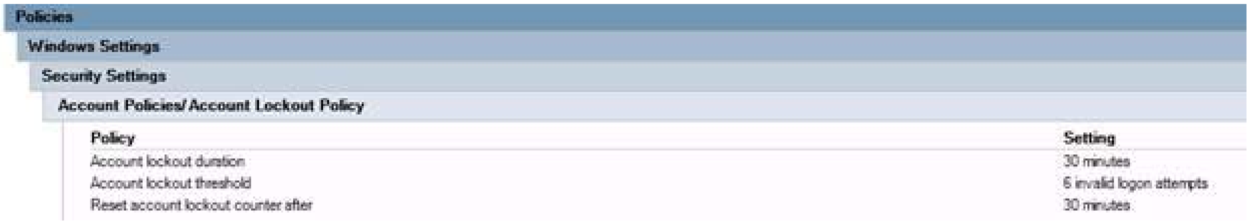
Policy	Setting
Account lockout threshold	0 invalid logon attempts

Account Policies/ Kerberos Policy

Policy	Setting
Enforce user logon restrictions	Enabled
Maximum lifetime for service ticket	600 minutes
Maximum lifetime for user ticket	10 hours
Maximum lifetime for user ticket renewal	7 days
Maximum tolerance for computer clock synchronization	5 minutes

Source: Jagged Peak

Exhibit 4: Account Lockout Policies



Policies

Windows Settings

Security Settings

Account Policies/ Account Lockout Policy

Policy	Setting
Account lockout duration	30 minutes
Account lockout threshold	6 invalid logon attempts
Reset account lockout counter after	30 minutes

Source: Jagged Peak

Exhibit 5: Examples of Authentication Challenges

Approach	Description	Challenges
Email validation	When user logs in, an email message is sent to the user's account and user must enter a code or click a link to confirm identity in order to complete the login.	Delays in email may slow down login. Spam filters may interfere with delivery. If a user's system is compromised or stolen, intruder may be able to login to user's email account as well.
SMS validation	Same as email validation, except code required to login is sent to mobile phone via text message.	Not all users may have cell phones. May incur a cost from text messaging.
Telephone validation	Same as email validation, except code required to login is delivered audibly, via a phone call.	May require an alternative phone line from the customer service line for support purposes. May involve a multi-lingual implementation. May incur a cost.
User-provided security questions	User provides responses to a variety of security questions and is prompted at random for a response.	Subject to guessing. Users often forget the precise response they previously established. Does not add a second factor (both password and security questions are "something you know").
System-provided security questions	System generates security questions based upon information gathered from external sources.	Subject to error. Can seem like an invasion of privacy to users. Does not add a second factor (both password and security questions are "something you know").
One time password (OTP) code book	User is given a book of passwords that apply to specific days that must be entered in addition to the personal password.	Code books must be distributed, a substantial cost for physical code books. If code books are distributed electronically, they can be accessed by intruders in the event a user's system is compromised.
OTP password token	A device that provides the user with an OTP that changes periodically or sequentially. It is entered in addition to the personal password.	Tokens add to costs. Owing to their small size, they may be lost or stolen easily. Would require additional decisions, as many token-based alternatives exist.
Fingerprint, handprint or eye scan	Biological biometric approaches that provide highly accurate identification.	Require specialized hardware for each user that is becoming increasingly available but remains costly when thousands of users are involved.
Facial and voice recognition	A less accurate form of biometric identification that can be accomplished using common hardware, such as a webcam or phone.	Not all users will have acceptable hardware. Both facial and voice recognition remains in its early stages and is subject to error and potential spoofing (e.g., using a photo or recording).
Typing patterns	The user is required to type a phrase and the system validates the pattern.	Subject to error; technology is in early stages. Requires training the system.
Customer-selected approaches	Customer is allowed to choose from a menu of authentication options based on their particular situation.	Costly and complicated to implement. Does not necessarily address challenges associated with a particular approach.

Source: Developed by case writer