Trustwave®
SpiderLabs℠

Dear Reader:

Over the last few years, Trustwave has published a data breach report. For 2010, we have broadened the report beyond compromise trends and statistics to address the growing threats in the global marketplace, those that we see in everyday practice.

In addition to the analysis of breach investigations you've come to expect from Trustwave and an actionable remediation plan, this report provides detailed technical information on the top vulnerabilities encountered during the penetration tests we performed in 2009. By identifying and sharing these vulnerabilities, we aim to provide information security departments and teams with the information necessary to protect their organization thoroughly.

As you may know, Trustwave is a leading global compliance and security firm. Four years ago, we added the SpiderLabs team to provide expertise and focus on penetration testing, application security and incident response to address our clients' needs. The vision of SpiderLabs is to be the specialty security team that organizations turn to because of our security knowledge, practical experience, quality and value.

The 2010 Global Security Report draws from this commitment, and is intended to provide readers with the current state of information security, as well as recommendations on how to secure their organization's sensitive data. We've tried to present these recommendations in a readily accessible format, requiring little to no capital expenditure for implementation.

As with each release of our SpiderLabs team's data, we're excited to share this with our customers and the industry at large. Our goal is to understand how breaches happen and share that knowledge with you, allowing us to work together to eliminate these threats for all businesses.

Nicholas J. Percoco
Senior Vice President, SpiderLabs

Trustwave®
SpiderLabs℠

Table of Contents

Introduction

Trustwave has conducted compromise investigations for over eight years, and in that time, businesses have made progress in protecting their data. Unfortunately, criminals are progressing just as quickly, adapting techniques to evade detection and obtain persistent access within target environments.

In 2009, the most notable trend is the continued use of existing attack techniques despite the security industry's awareness of these vulnerabilities. From our client work, we have evidence that major global companies have employed vulnerability chasers, searching out the latest threats and zero-day vulnerabilities, versus utilizing the work of security professionals to perform an impact analysis. Thus, basic security threats are being overlooked; our investigations and penetration tests frequently uncovered old vulnerabilities or vulnerabilities not considered a relevant threat to the overall security posture of a business. Analysis of these engagements revealed attackers were taking significant advantage of both new and old vulnerabilities.

Also in 2009, thousands of businesses (likely many more in 2010) were scrambling to define their policy and plans around such trends as social networking and cloud computing, while critical items like patching and password management fell by the wayside.

To help with the planning and security efforts of our readers, this report offers an analysis of compromise data, an analysis of proactive security testing and an actionable, global remediation plan. These three distinct sections are as follows:

1. Analysis of 2009 Incident Response Investigations: This section covers the analysis of results from more than 200 investigations performed in 2009 due to a suspected security breach identified by either the target organization or a third party (e.g., regulatory body, credit card brands, business partners, consumer, etc.). This type of approach is often required by various regulatory bodies to understand the extent of and continued risk of losses that need to be reported to interested parties.

2. Analysis of 2009 Penetration Tests: This section covers the analysis of results from more than 1,800 tests performed in 2009 that companies elected to carry out against their own environment. This approach is used to either validate a company's understanding of its security posture or uncover unknown shortfalls and ultimately improve its security posture. The tests result in what "could" happen should criminals attempt to target the company.

3. Global Remediation Plan: The analyses of investigations and penetration testing, revealing the actionable priorities for securing the data, systems and facilities for all organizations.

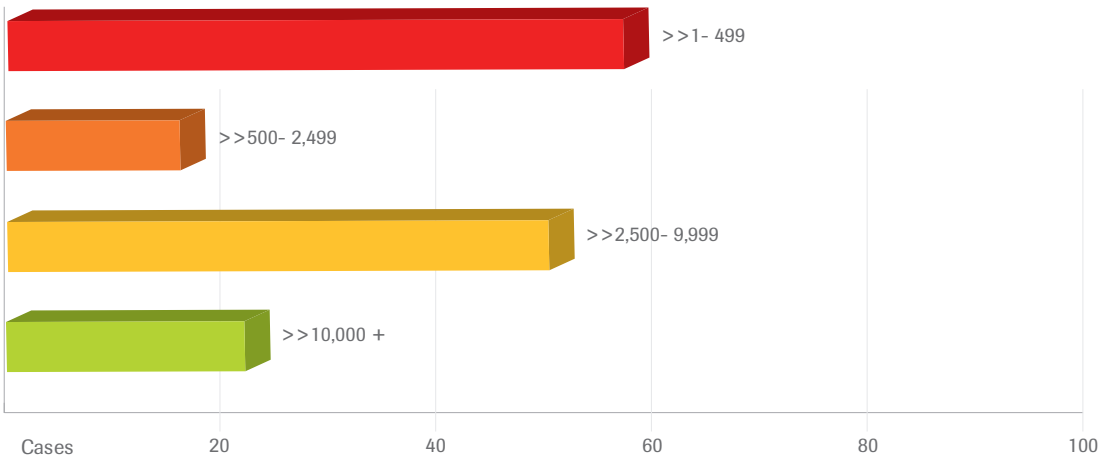
The overall methodology includes results based on firsthand evidence collected in 2009 by Trustwave's SpiderLabs advanced security team. Results were gathered during data breach investigations and penetration testing activities conducted for clients. All investigators use standardized tools to record data and relevant details for each case or test. Trustwave's SpiderLabs is committed to protecting the privacy of our clients, and the statistics within this report are presented in the aggregate only.

Analysis of 2009 Incident Response Investigations

Incident response is a collection of actions to stop an attack that is occurring and investigate how it occurred. Many technical and non-technical practices are applied to get the result that is considered acceptable by the interested parties. In some cases, the interested party is a single organization that has been compromised; in others, a regulatory body may also have an interest in the outcome of an investigation involving a breach.

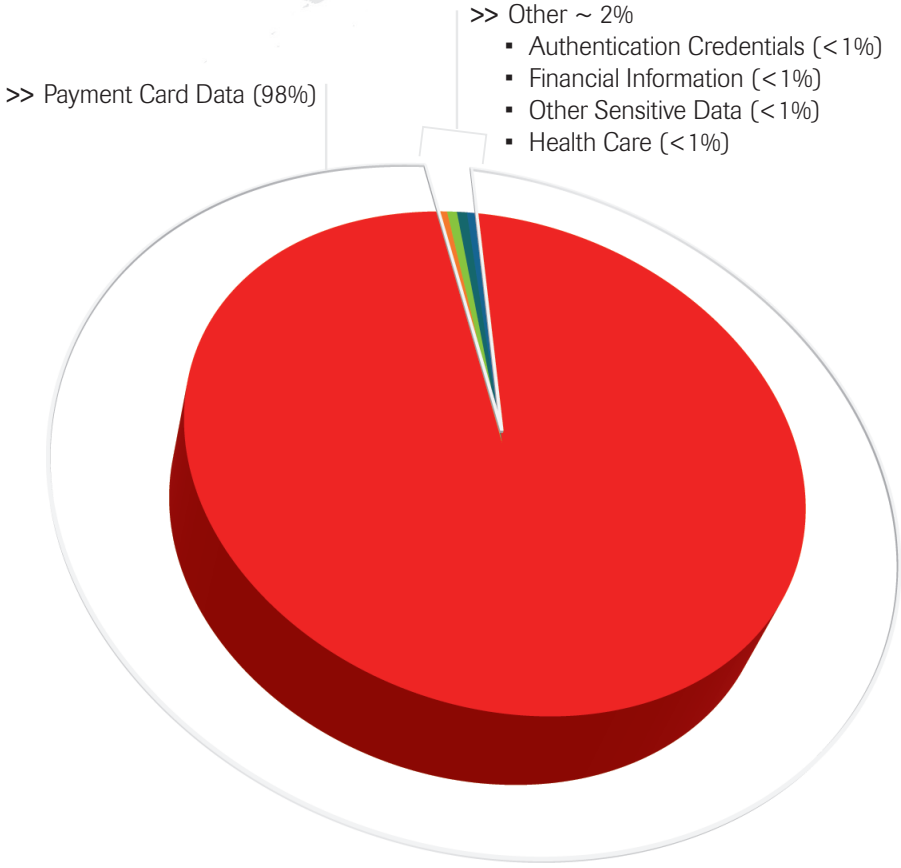
Company Size

Percentage of incident investigations by company size.



Countries Represented in 2009

Australia
Belgium
Canada
Chile
Cyprus
Denmark
Dominican Republic
Ecuador
Germany
Greece
Hong Kong
Ireland
Luxembourg
Malaysia
Puerto Rico
Saudi Arabia
South Africa
Sri Lanka
Switzerland
Ukraine
United Arab Emirates
United Kingdom
United States
Virgin Islands



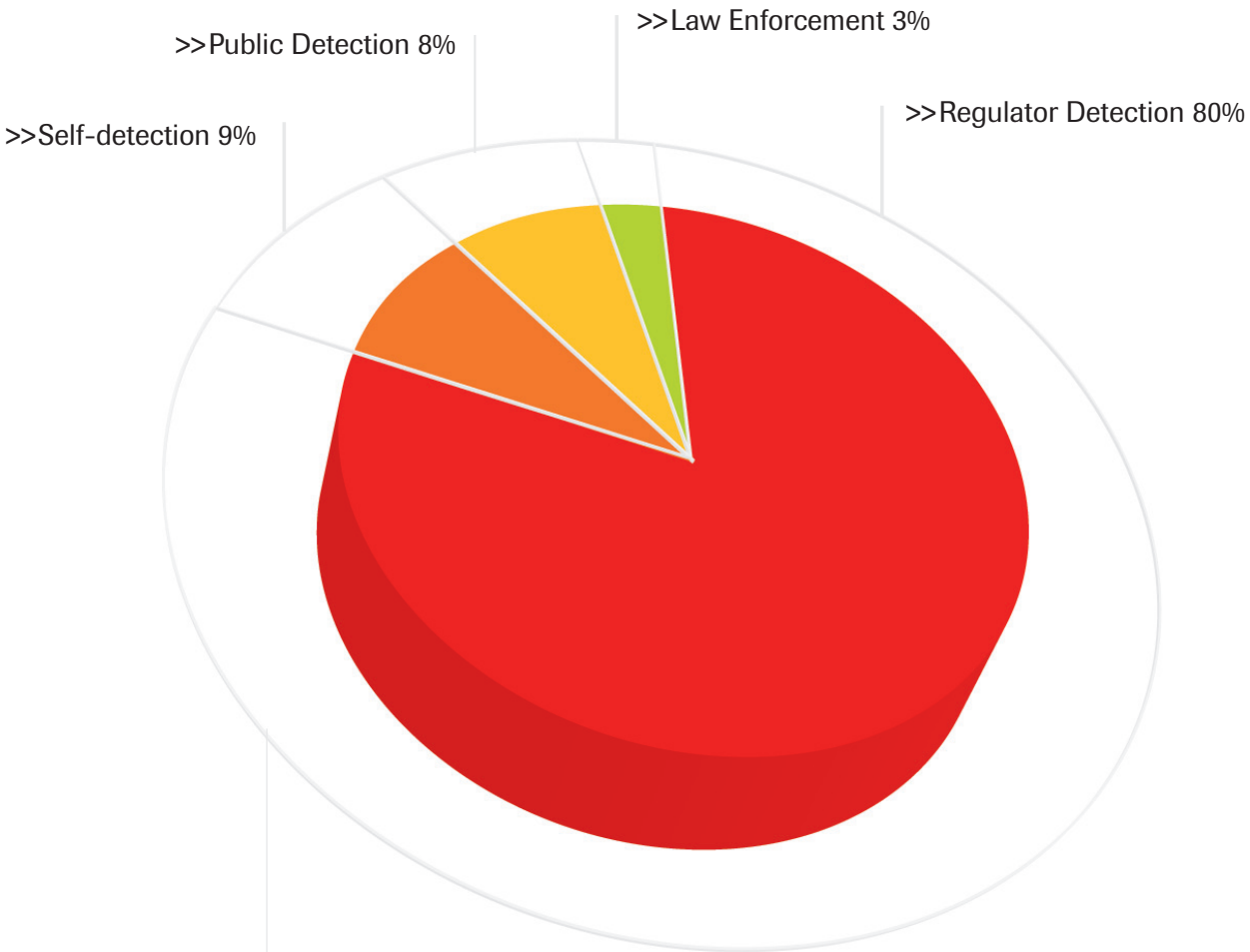
The 2009 incident response investigations targetted these types of data.

Data at Risk

Payment card breaches accounted for an overwhelming majority of our caseload in 2009. The targeting of payment data is to be expected, as this information can be sold or laundered through established black market networks for quick financial gain.

Additionally, Trustwave's SpiderLabs is one of only a handful of firms authorized to perform payment card breach investigations on behalf of the five major card brands (i.e., American Express, Discover Financial Services, JCB, MasterCard Worldwide and Visa Inc.).

In 2009, we conducted more than 200 security investigations globally. These investigations included both conclusive and inconclusive evidence of a data breach, although most were conclusive (18% inconclusive–no evidence of an attacker accessing or retrieving sensitive data). On average, a lapse of 156 days between the initial breach and detection was found across all investigations. The most common reason for the initiation of an investigation was regulatory detection.



Regulator Detection – an external regulatory body (such as a credit card company with access to the organization's data) detects a data pattern that indicates there may be some unauthorized access and usage of the organization's data.

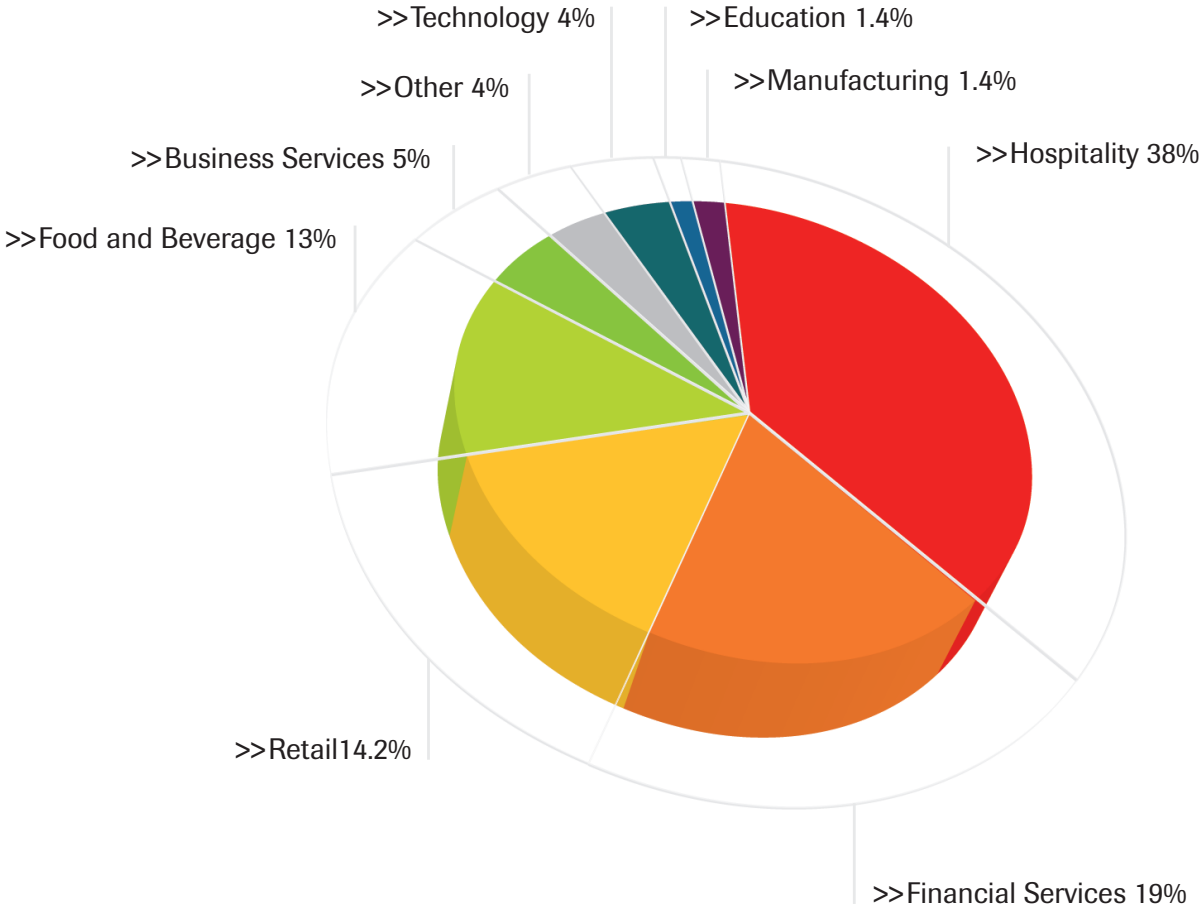
Self-Detection – the attacked organization has controls in place to detect an incident and appropriate personnel were alerted.

Public Detection – an organization or customer of the organization identifies an issue with his/her data and reports it directly to the organization or makes a public disclosure of the issue.

Law Enforcement – a law enforcement agency gathers intelligence regarding a potential data breach concerning a specific organization and notifies the organization or regulatory body.

Industries Represented

In prior years, the food and beverage industry shouldered the brunt of data breaches. In 2009, the hospitality industry was increasingly a target representing 38% of all breaches investigated by SpiderLabs. Insecure network connections, in conjunction with poor security controls, allowed unfettered network access between the multiple properties of several hospitality companies. The majority of our hospitality cases are interrelated; a single site breach resulted in attackers propagating to additional properties.



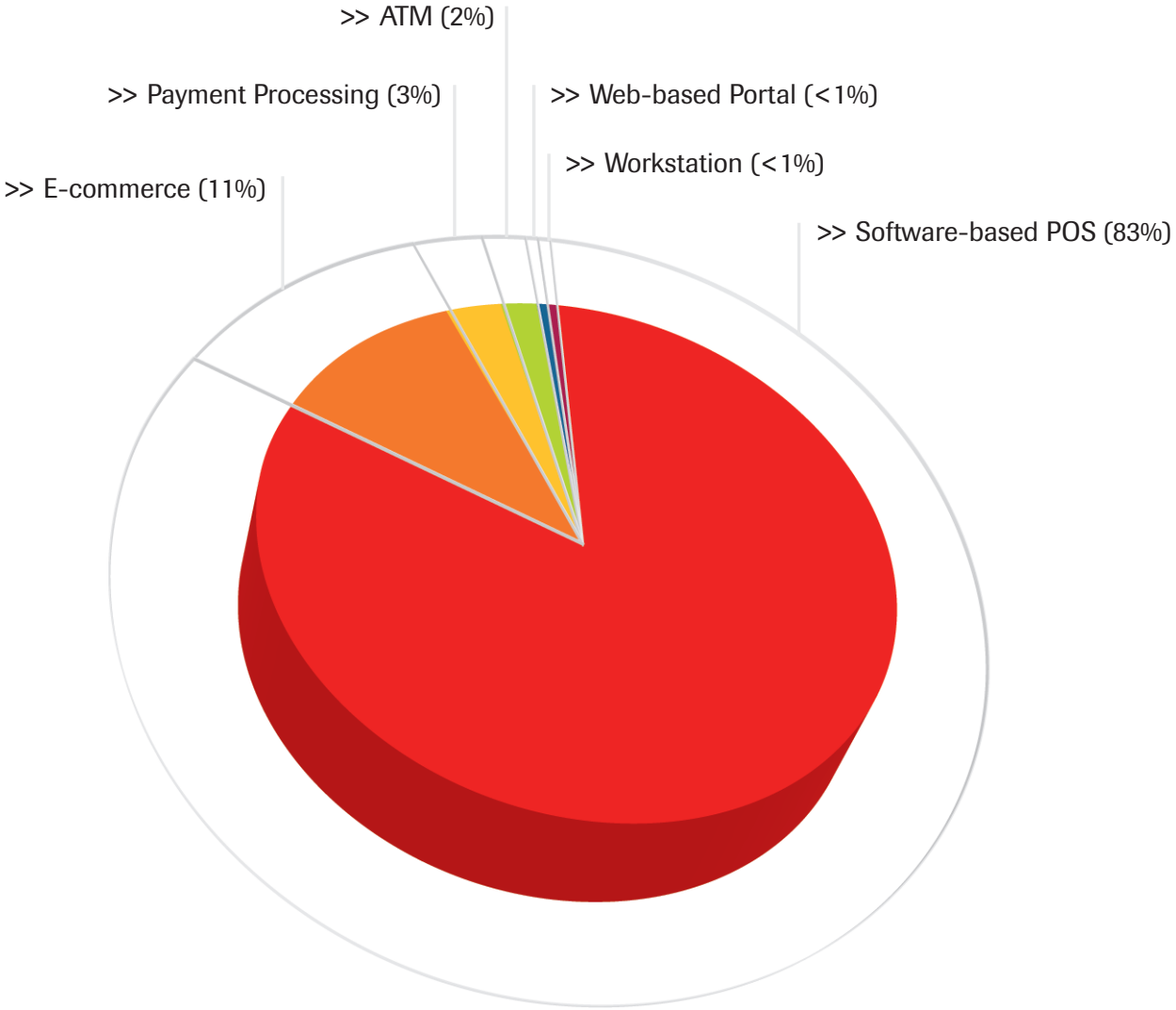
Breaches within the financial sector also resulted in significantly higher data loss when compared to certain other industries. As companies in this industry are often responsible for the aggregation, storage or transmission of millions of data records, higher numbers of attacks and data loss are to be expected.

While investigations within the food and beverage industry decreased, this industry still accounted for nearly 13% of our cases. Evidence suggested the majority of breaches within this sector, along with the retail industry, were not a result of explicit targeting, but were simply opportunistic.

Types of Target Assets[1]

The majority of breaches we investigated involved the targeting of systems responsible for the processing or transmission of payment card data. Software point of sale (POS) systems were the most frequently breached. POS systems represent the easiest method for criminals to obtain the magnetic stripe data necessary to commit card-present fraud. Due to common existence of well-known vulnerabilities and sheer volume of potential targets, software POS systems are considered low-hanging fruit to even the novice attacker.

E-commerce systems were involved in 11% of our cases. Criminals are limited in the fraud they can commit by infiltrating e-commerce systems because without the magnetic stripe data, they are limited to card-not-present fraud. As a result, the majority of e-commerce breaches we investigated were within countries that have adopted secure payment card technology, such as EMV (often called "chip and PIN" cards). With the adoption of EMV virtually eliminating the ability to commit card-present fraud, many criminals within these countries have turned their sights to e-commerce systems.

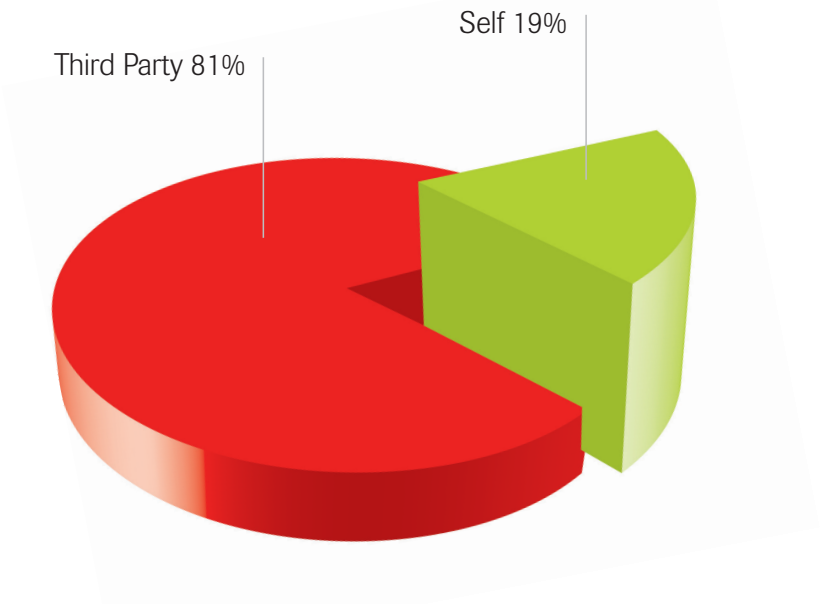


ATM and payment processing systems were again targeted last year. In most cases, the attackers successfully exploited the environment housing the financial transaction switch and related financial application programming interfaces. In these complex breaches, attackers obtained magnetic stripe data in conjunction with PIN, giving the criminals access to cash. In our 2009 cases, the breach of data from ATMs was mostly contained to hardware tampering. Criminals physically inserted skimming devices and hidden cameras to capture magnetic stripe data and PIN when entered. This year we also witnessed credentialed malware explicitly targeting several brands of ATM software (for a description of credentialed malware, please turn to page 11).

[1]Attacker Source Address Geography: Dozens of methods exist to hide an attacker's location; an attacker could be physically next door to the target environment and have his or her source IP from a location 10,000 miles away. We purposely excluded this data analysis from our report to attempt to limit sensationalizing this area. A false sense of security is created when people or organizations believe that attackers are located on the other side of the world. But anyone who does business in a connected way needs to be concerned about threats of all types and understand that threats can come from just a few feet away.

System Administration Responsibility

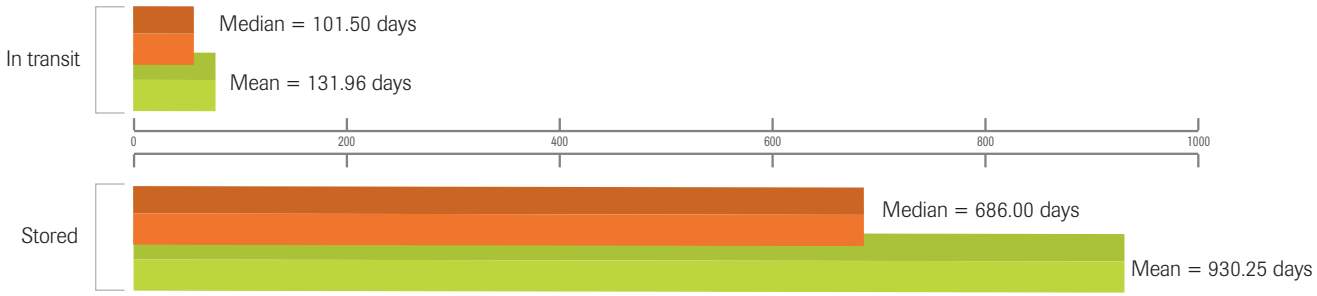
Most compromised systems we investigated in 2009 were managed by a third party. In the software POS system breaches we investigated, this third party often introduced many of the deficiencies exploited by the attacker, such as default vendor-supplied credentials and insecure remote access applications.



Window of Data Exposure

Based on our experience, an accurate accounting of compromised records cannot be determined by the investigating firm in most cases. This is especially true in breaches. After a breach, the impacted card brands, processing banks or health care organizations request the investigating firm to provide a "window of data exposure." All impacted accounts (transactions) processed by the compromised entity within this set window are supplied to the investigators, creating an extremely accurate accounting of the potentially compromised accounts. Unfortunately, this process can occur months after the completion of the investigation. Therefore, the investigating firm is rarely privileged to this information or present when an accurate final accounting of compromised accounts is finally determined.

By examining windows of data exposure, one trend is worth noting. Breaches involving the harvesting of data in transit averaged a window of data exposure of about 101 days (median), while those breaches targeting stored data averaged 686 days (median). This trend strengthens the argument that the elimination or protection of stored data is instrumental in minimizing the impact of a breach.



Anatomy of a Data Breach

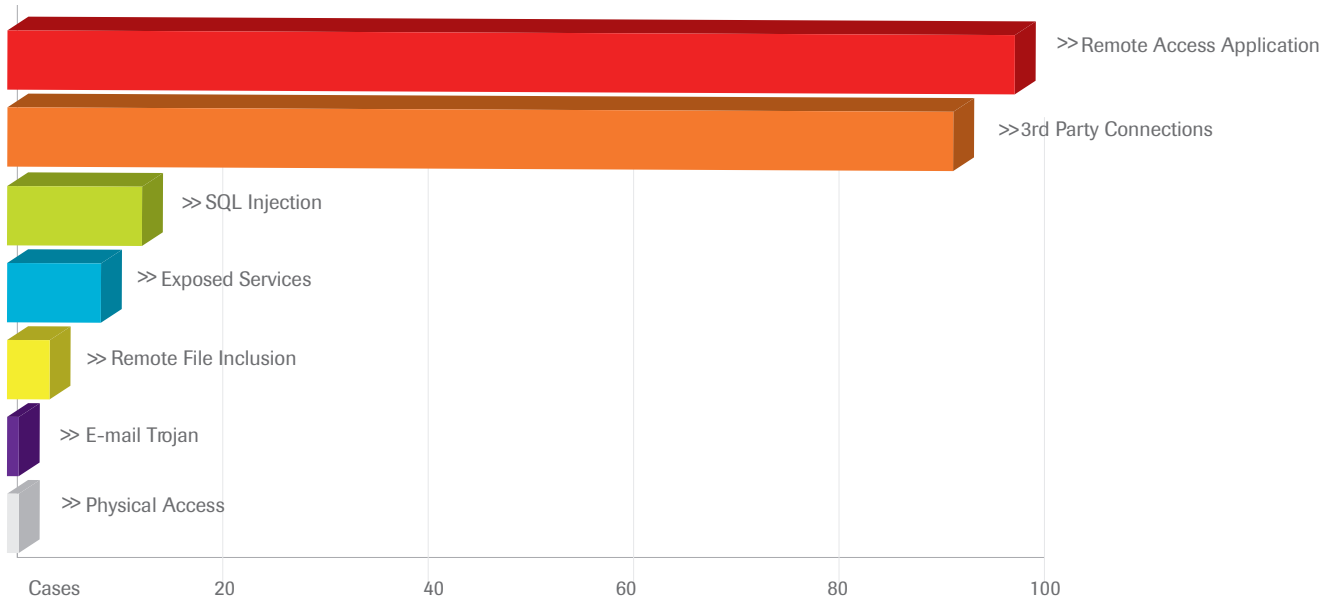
At its most basic level, a data breach consists of three components:

1. Initial Entry: The method utilized by the attacker to gain unauthorized access to a system.
2. Data Harvesting: The method utilized by the attacker to obtain data.
3. Exfiltration: The method utilized by the attacker to export data.

Only by differentiating between these components can we begin to clearly understand the anatomy of a data breach.

Initial Entry

The majority of criminals obtained system access by exploiting existing channels such as remote access applications or trusted internal network connections.



Defined Top 3 Attack Methods

Remote Access Application: A remote access application is any application used to administer a system remotely. While GUI-based remote access solutions are popular today, command-line remote applications, such as telnet, are still in use today. The telnet protocol standard was defined at UCLA on March 5, 1973.

This method will allow operating system level access. The level of attack difficulty is intermediate, but can be trivial when passwords are blank or left as the default value.

Third Party Connectivity: This could include any telecommunications line connecting two or more physically dispersed networks, such as multiprotocol label switching (MPLS), asynchronous transfer mode (ATM), and frame relay, a data transmission technique. This easy-to-use method could leave the entire network accessible via the connections, and therefore open to compromise.

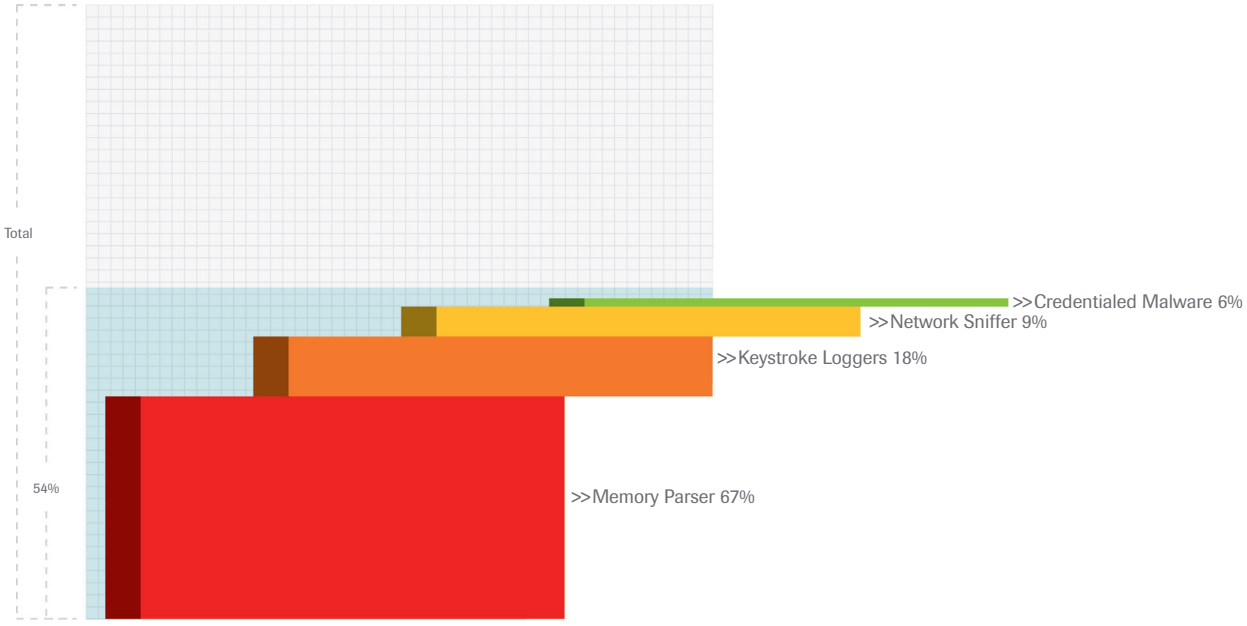
SQL Injection: Gaining mass popularity with the explosion of Web-based, database-driven applications, SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application. This method of attack has existed since 1998; on Christmas Day, a security researcher known as rfp detailed the attack in an article called "NT Web Technology Vulnerabilities" published in Phrack magazine, issue 54.

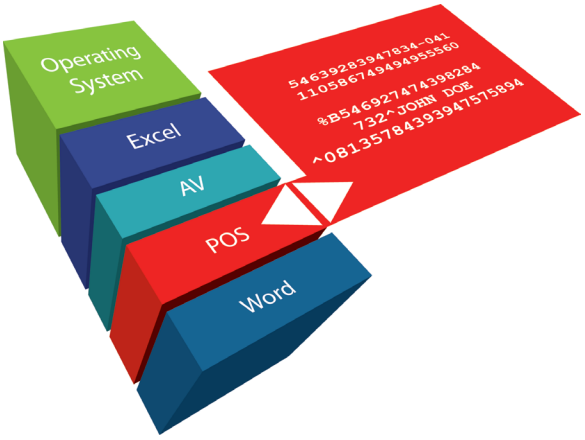
Traditional SQL injection attacks allow the extraction of data within the impacted application, although more complex methods can allow the shell access via stored procedures resulting in full system ownership.

Data Harvesting

Historically, in the majority of our cases, compromised systems were found to be storing large amounts of data that was simply not secured properly and, therefore, readily available to attackers once initial entry to the system or application was obtained. In 2009, we observed the continued shift away from basic "smash and grab" attacks to more complex methods of data harvesting.

As various application security standards and regulatory requirements (i.e., OWASP, PA-DSS) have begun to take hold, the elimination of insecure data storage practices has also been sustained. In response, attackers have devised methods to obtain data earlier in the application's data processing cycle, often by harvesting data in transit. Malware, in particular, was used to conduct these attack. In 54% of cases, attackers harvested data in transit with malware, using the four types described below. In cases where criminals gained entry via remote access applications (45%), conclusive evidence of a system breach was found. Not only were remote access applications exploited in high numbers, but default vendor-supplied or weak credentials expedited entry in 89% of these cases. The combination of insecure remote access applications and default vendor-supplied credentials frequently appeared to be standard practice for third parties providing system support for their clients.



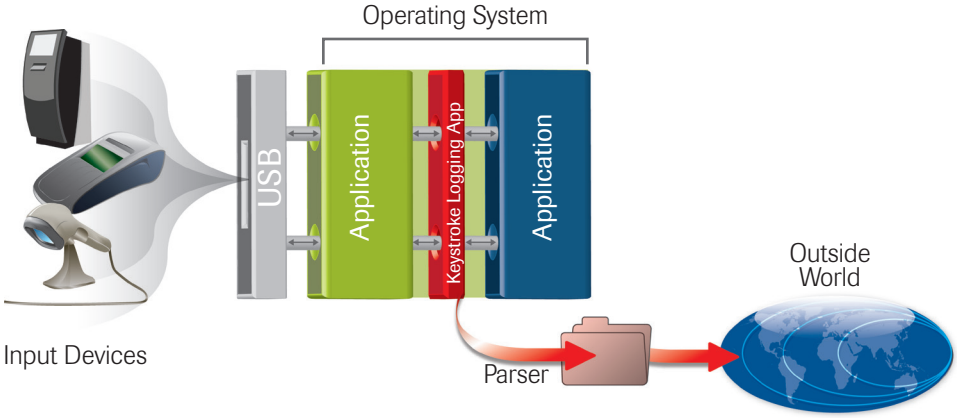


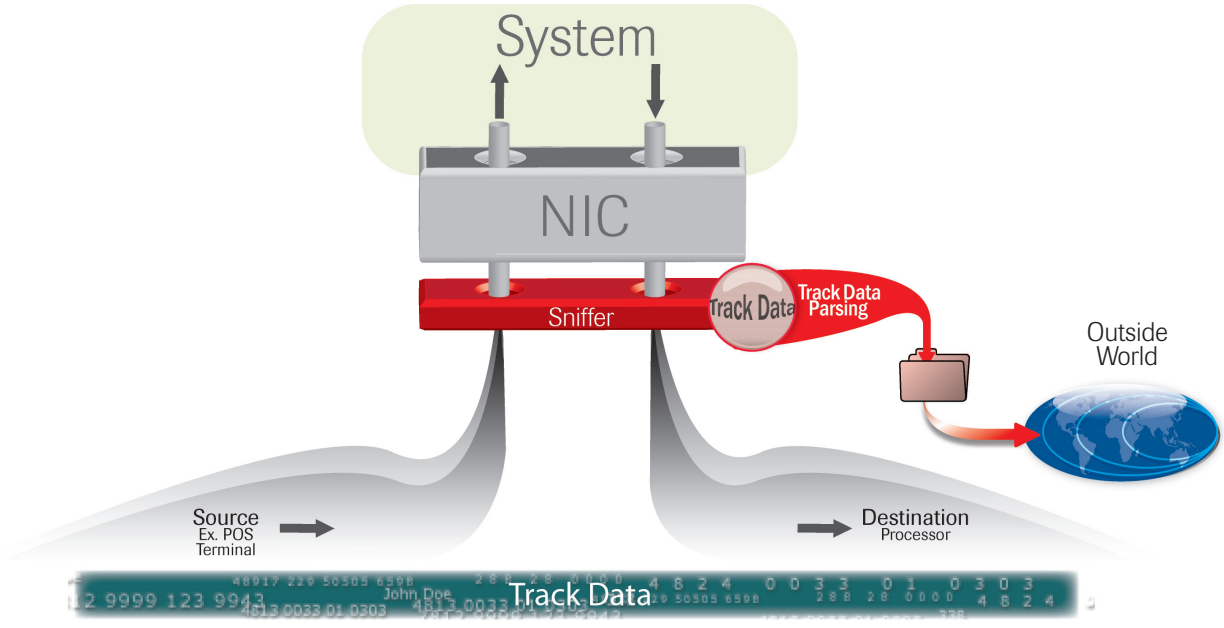
Memory Parser (67%)

A memory parser is a software application that is designed to specifically monitor the random access memory (RAM) being used by a certain process. When this process interacts with data, it parses this data for the specific information it is designed to look for. This could be personally identifiable information (PII) or financial information, such as credit card numbers and bank accounts/routing codes.

Keystroke loggers (18%)

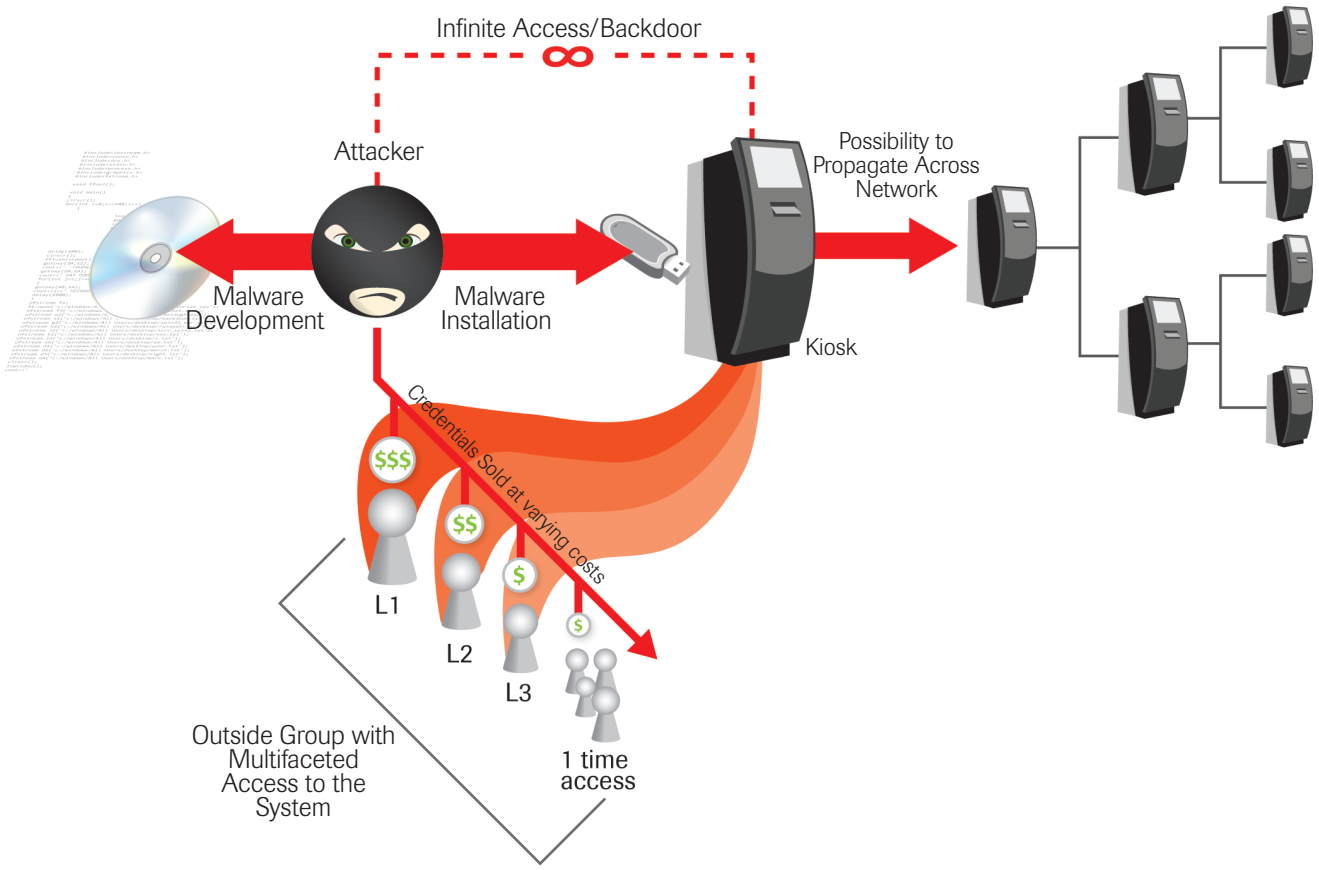
A keystroke logger intercepts data as it is being entered at a computer terminal via the keyboard, touch screen or external data entry device (i.e., card reader). Many implementations we observed included data export functionality via SMTP or FTP, which automatically uploaded the harvested data to the attacker's chosen device. The tool was typically found on end-user systems or end-point systems used for data input versus centralized transaction processing.





Network Sniffer (9%)

A network sniffer is a device or software application that listens to traffic on a network much as a phone tap listens to the verbal conversations on a phone line. It is typically installed on a system that is along a data flow path. This is often a centralized or regionalized system used to receive and process data coming from many endpoints. Ideally, sensitive data being sent over any network would be encrypted, but often clear-text sensitive data travels across private and assumed private networks. At the time of publication, compliance standards, such as PCI DSS, have not yet tackled the issue of private network data traversal encryption.



Credentialed Malware (6%)

The bleeding edge of malware, credentialed malware is multi-user malware frequently used by organized crime groups. We have seen the most widespread use of this malware in the ATM. In its basic form, this type of malware works like any other application. Embedded into the software is the ability to control access to the various functions using a method of authentication, typically a physical token (i.e., an ATM card). The token is encoded with similar data that the targeted application commonly sees, but when the malware recognizes the token it activates a portion of its code for an attacker to use. In the ATM cases, the malware allowed attackers to print out ATM card numbers and PINs recorded by the malware, uninstall the malware from the system, and even withdraw all of the cash stored within the ATM itself. The developer of the malware can sell and distribute tokens of various levels. This technique also allows organized crime groups to authorize lower level members to perform certain tasks of value while leaving the more lucrative tasks, such as ejecting all of the cash from a machine, to trusted individuals.

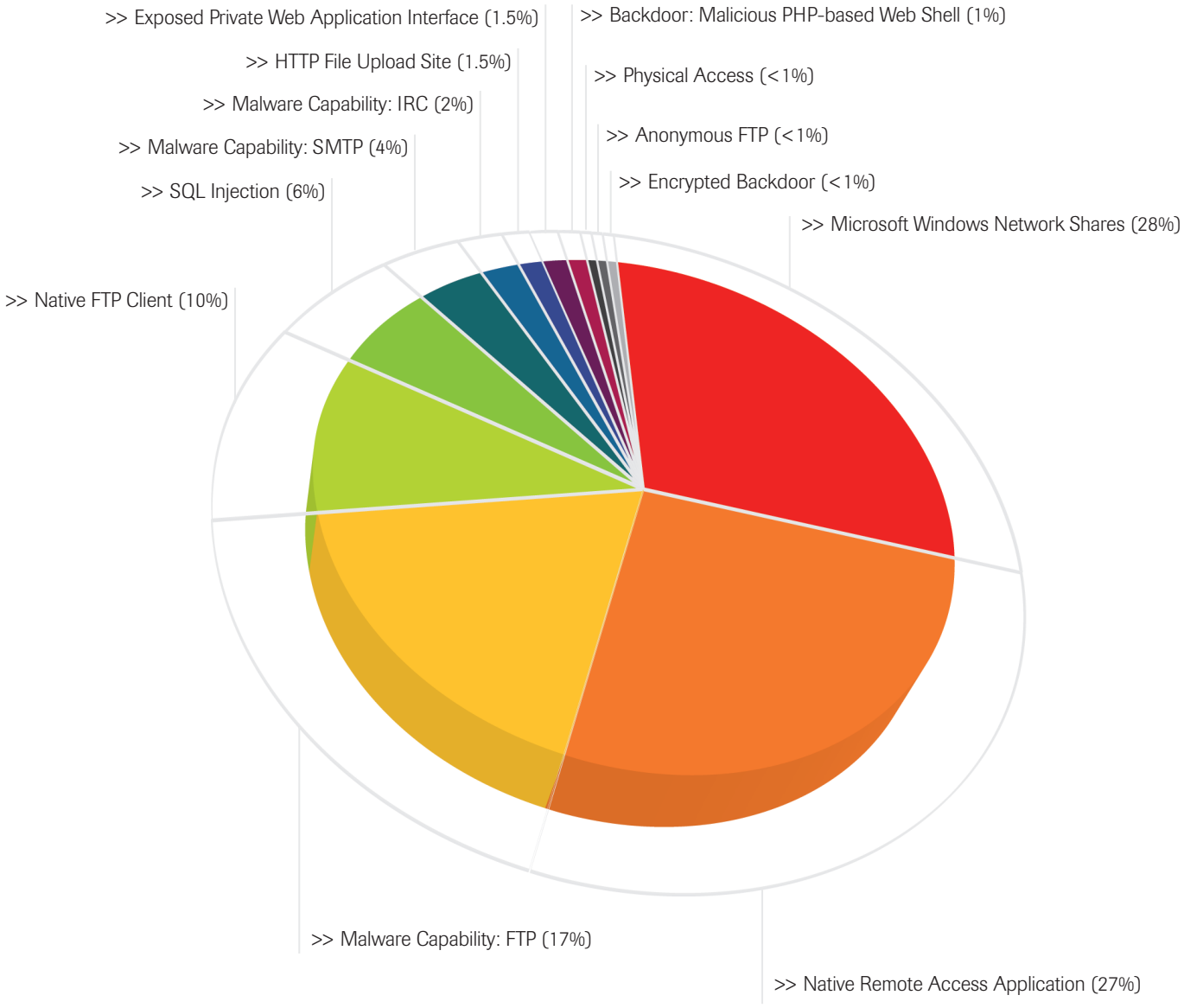
Exfiltration

Exfiltration is usually accomplished by copying the data from the system via a network channel, although removable media or physical theft can also be utilized. In the interrelated hospitality cases, SpiderLabs determined that the attacker broadcasted to additional targets via private network circuits, utilizing native Microsoft Windows Network Shares to exfiltrate data from the local environment.

In 38 cases, the attackers used the remote access application previously utilized for initial entry to extract data. Other existing services, such as native FTP and HTTP client functionality, were also frequently leveraged for data extraction. Specifically, when malware was utilized for data extraction, FTP, SMTP and IRC functionality were regularly observed. (In reverse analysis of custom malware, binaries would disclose the existence of FTP functionality including hardcoded IP addresses and credentials.) With off-the-shelf malware, such as keystroke loggers, attackers most often used built-in FTP and e-mail capabilities to exfiltrate data. When e-mail services were employed for extraction, the attackers often opted to install a malicious SMTP server directly on the compromised system to ensure the data was properly routed.

Only a single case contained the use of an encrypted channel for data extraction, suggesting that criminals are rarely concerned with raising alarm. Due to natively available network services, lack of proper egress filtering and poor system monitoring practices, criminals are using available network services or choosing to install their own basic services.

Percentage of Methods Used to Exfiltrate Data



Off-the-Shelf versus Custom Malware

Off-the-shelf malware is available to the general public and, when utilized in data breaches, is legitimate software utilized with malicious intent. For example, the majority of keyloggers utilized by attackers in our cases are available for purchase online.

The public availability of off-the-shelf malware does have a major drawback: anti-virus vendors can quickly update their signatures for detection. Accordingly, off-the-shelf malware was found to be the technique of choice for many attackers where anti-virus was not maintained.

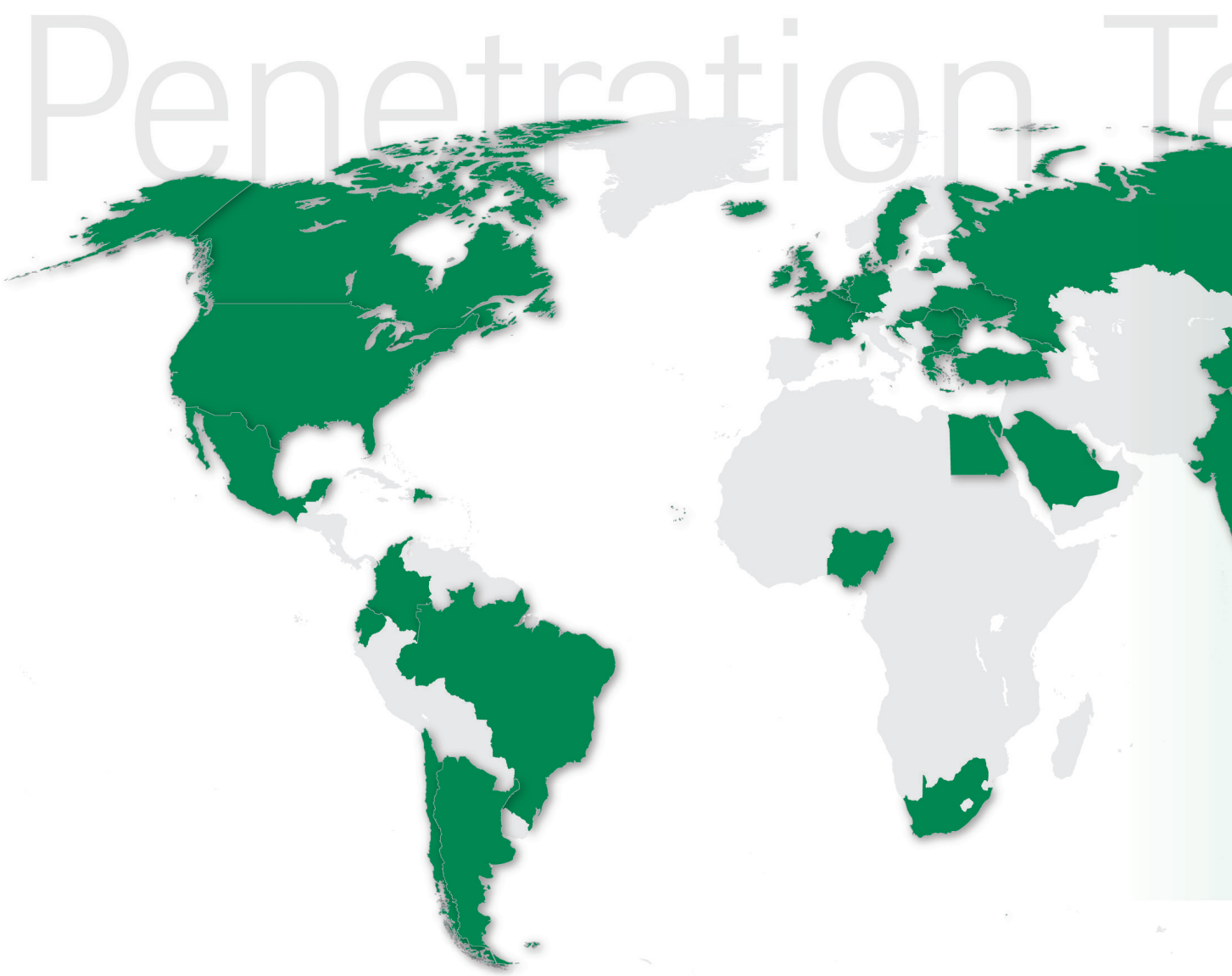
Custom malware, on the other hand, is developed with explicit malicious intent and distribution is tightly controlled. Its main strength is its ability to remain undetected in many instances by being able to bypass traditional anti-virus detection.

Substantial knowledge of the target environment and technical resources are often required to create custom malware. Such is the case with the memory parsing malware SpiderLabs encountered; every instance was custom tailored to the specific environment under attack. Initial implementations of the memory parsing malware were inefficient and easily noticed by anyone who was looking, but over time we observed as many as 15 to 20 different builds–each version addressing issues the attackers encountered along the way.

What about Hardware?

Hardware tampering requires electrical and social engineering to be successful. The attacker needs to obtain the hardware, modify it and then devise a plan to "swap out" the legitimate hardware being used in production with the modified versions. The modifications we experienced in this area were advanced enough to include local data storage and wireless modems for attacker data retrieval.

While the number of cases this past year was relatively small, we believe hardware tampering will grow over the next several years. The prize target for any organized crime group would be to infiltrate the device manufacturing company. Given the general lax state of security in the world today, a crime organization would have little trouble executing this attack at one of the second tier device manufacturing companies, resulting in modified hardware being shipped to customers.



Analysis of 2009 Penetration Tests

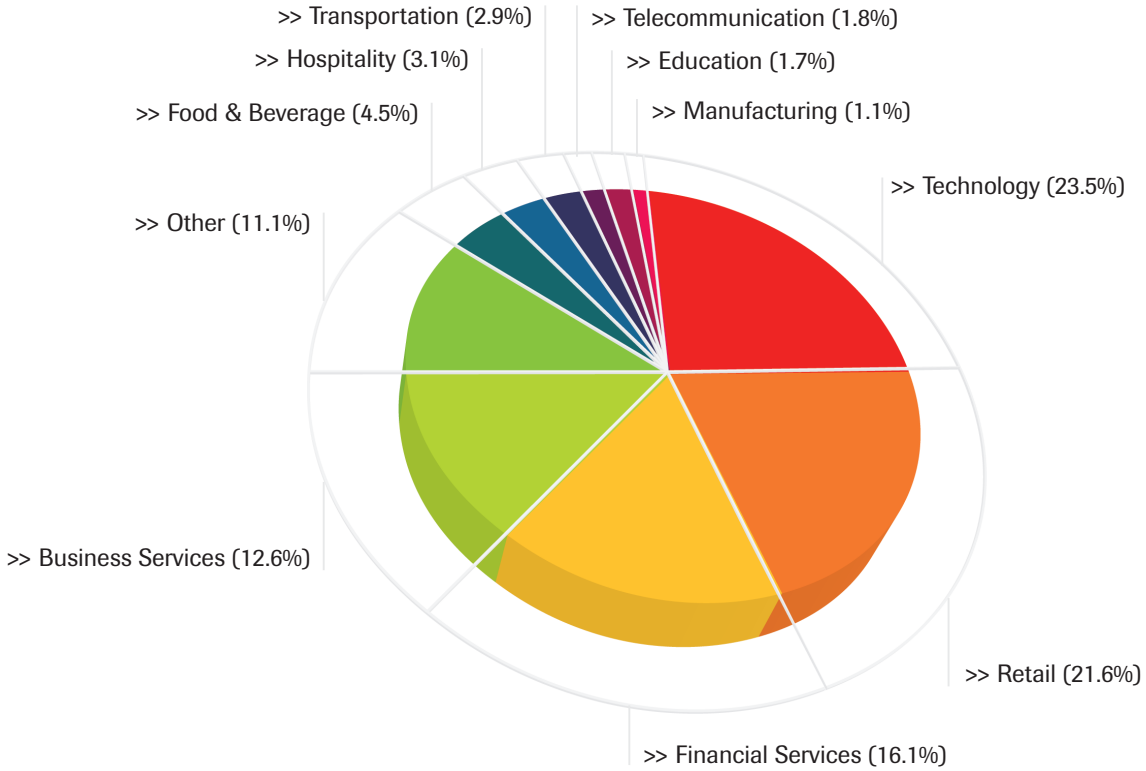
This section contains the analysis of results from more than 1,800 penetration tests and is categorized by test type. The test types include: external network, internal network, wireless, physical/social and application.

Company Size

Percentage of penetration tests by company size.

Countries Represented in 2009

Australia
Argentina
Belgium
Brazil
Bulgaria
Canada
Chile
China
Colombia
Croatia
Denmark
Dominican Republic
Ecuador
Egypt
France
Georgia
Germany
Greece
Hungary
India
Japan
Iceland
Ireland
Lithuania
Luxembourg
Macedonia
Malaysia
Malta
Mexico
Moldova
Netherlands
Nigeria
Republic of Cape Verde
Romania
Russian Federation
Saudi Arabia
Singapore
South Africa
Sri Lanka
Sweden
Switzerland
Taiwan
Turkey
Ukraine
United Arab Emirates
United Kingdom
United States



The penetration tests performed in 2009 represent 10 different top-level industries.

The Top 10 Issues Identified by Test Type

External Network Penetration Test

An external network penetration test is a time-based test that simulates a targeted attack on an organization's system or set of systems accessible via the Internet.

After initial reconnaissance of the target environment, we review each device or service and manually attempt to identify both known and unknown vulnerabilities. We then attempt to manually string together vulnerabilities we have identified to create an attack vector. (An attack vector is simply an avenue for potential compromise.) Because each test is time-based, some attack vectors will result in a compromise and others will not.

1 Unprotected Application Management Interface Available from the Internet

Definition: When, on a Web server, the management interface for the application engine (e.g., Cold fusion, Websphere, tomcat, jboss, etc.) is in the default location, available from the Internet and provides little or no protection (weak or no credentials).

Circa: This was a known issue with Web server software such as Netscape Fasttrack and Enterprise Server back in 1994. At that time, the documentation warned users to only start the administrative server when one needed to use it.

Impact: Many times, administrative access to the application management interface means that an attacker can deploy their own malicious applications on the Web server, or alter the parameters for other applications. This can lead to system level access on the Web server and possible compromise of a current application's data.

Attack Difficulty:

2 Unprotected Network Infrastructure Component, Management Interface Available from the Internet

Definition: In a network infrastructure component (i.e., router, switch, firewall or VPN concentrator), the management interface (e.g., Web, ssh, telnet) will be in the default location, available from the Internet and provide little or no protection (weak or no credentials).

Circa: This vulnerability is a combination of weak passwords (1979) and improper access control (1993).

Impact: Administrative access to a network component can allow an attacker to change network access controls, redirect and sniff network traffic, and perform a host of other attacks that can lead to multiple device and data compromises.

Attack Difficulty:

How the Top 10 List Works

Each section details how the tests are performed and any methodology associated with the test type. For each test type, we analyzed the vulnerabilities we discovered within a client's target network, environment or application, and noted the most frequent occurrences to develop a Top 10 list. In addition to listing the top 10 vulnerabilities for each type of test we performed, we also included these items to help the reader with remediation planning:

- Definition: The description and relevant details of the vulnerability.
- Impact: The effect the vulnerability could have on an organization.
- Circa: An estimate of the year the security industry became aware of this type of vulnerability, along with any tools developed to exploit the vulnerability.
- Attack Difficulty: The level of difficulty and/or skill needed for this type of attack.

3 Unauthenticated Access to Internal Applications via the Internet

Definition: Internal applications sometimes share a server with applications that host external content. When this happens, the internal application can be accidentally exposed to the outside due to firewall rules that permit access to the external content. This happens frequently with Web applications, where the firewall allows all traffic destined to port 80 (http) to pass.

Circa: A point of exposure since the introduction of Web applications, this vulnerability may have been around since 1997, with the actual concept of Web applications introduced in the Java servlet specification released June 1997.

Impact: Access to internal applications, particularly those that have poor or no authentication, can give an attacker a great deal of information about internal resources. This is especially problematic if the application provides access to a database with sensitive information, such as credit card data.

Attack Difficulty:

4 Misconfigured Firewall Rules Permit Access to Internal Resources

Definition: Depending on the complexity of the firewall access control list, mistakes can cause data to be forwarded to hosts inside the network.

Circa: Firewall misconfigurations have existed from the inception of firewalls. TIS, under a broader ARPA contract, developed the firewall toolkit (FWTK), and made it freely available under license on October 1, 1993 (www.fwtk.org).

Impact: This often occurs when port-forwarding is in place; an erroneous rule here can give an outsider access to an internal, sometimes vulnerable, machine. Poorly implemented firewall rules can provide access to the firewall itself. By scanning all ports open to the outside and performing service fingerprinting tactics, services such as unsecured local test databases can be used to gain access to the internal infrastructure, resulting in an internal network compromise from the Internet

Attack Difficulty:

5 Default or Easy to Determine Credentials

Definition: Hosts such as conferencing systems, network appliances and test machines are sometimes deployed with weak passwords. Examples include "admin:admin," "admin:password," and various combinations of the organization's name.

Circa: Robert Morris and Ken Thompson first published their findings on this vulnerability in the 1979 paper, "Password Security: A Case History."

Impact: Weak administrative passwords often provide substantial access to the target infrastructure. Passwords are frequently gained via brute force attacks or guessing. The vulnerability can easily lead to a compromise of the local network, further compromise of credentials or expanded systems access.

Attack Difficulty:

6 Sensitive Information, Developer Files or Source Code Publicly Accessible by Web Server Directory

Definition: Sensitive information such as text files with PII or cardholder data (CHD), archive files of the application source code, or old developer files can be left in an accessible directory of the Web server. These files are found through URL directory traversal or through simple file name enumeration or guessing common file names and/or paths.

Circa: Data available by the Web server directory has been an exposure point since the advent of Web servers (approximately 1990).

Impact: Depending on the information contained in the files, they can provide an attacker with information about passwords, such as database passwords hardcoded in the source code, or the files can contain PII or CHD. This can lead to system compromises, and a list of CHD is a data compromise in itself.

Attack Difficulty: 2 Easy

7 Static Credentials Contained in Client

Definition: A downloadable thick client uses hardcoded credentials that could be easily obtained by an attacker. Thick clients are full-featured applications that are run on a workstation and are functional whether connected to a network or not. For example, an FTP account or a VPN account, with a statically encoded username and password, might be hardcoded within an application that contains a data transfer function.

Circa: The configuration oversight has existed since the advent of thick client applications, which originated in the 1980s.

Impact: The presence of static credentials could be observed by an attacker and used to access the statically encoded user account—even if the password is hashed or encrypted in the code or configuration file. This vulnerability may lead to a contained system or data compromise. In a worst-case scenario, this could lead to a compromise of the internal network from the Internet.

Attack Difficulty: 2 Easy

8 DNS Cache Poisoning

Definition: The domain name system (DNS) is the naming system used for computers connected to the Internet. DNS cache poisoning is a maliciously created (or unintended) situation that provides data to a caching DNS server that did not originate from authoritative DNS sources. Once a DNS server receives non-authentic data, subsequently caching it for future performance enhancement, it is considered "poisoned." When a DNS server is poisoned, the local users' DNS requests will result in traffic being diverted to an attacker's computer.

Circa: This has been an exposure point in the industry since the DNS was implemented; however, this was brought to the public's attention at Black Hat USA 2008 by Dan Kaminsky.

Impact: An attacker who poisons DNS entries may make attacks possible from the Internet. This vulnerability in itself does not compromise any system; however, it can facilitate malware and other attacks that can lead to compromise.

Attack Difficulty: 3 Medium

9 Aggressive Mode IKE Handshake Supported

Definition: Internet key exchange (IKE) is a protocol standard that, in conjunction with the Internet Protocol Security (IPSec) standard, provides authentication of IP packets. When IKE aggressive mode is enabled and requested by the client, the VPN concentrator provides a hashed pre-shared key (PSK) prior to beginning encrypted communications. The hashed PSK can be easily captured with a network sniffer; this does not require any authentication credentials or an established IPSec tunnel.

Circa: Aggressive mode IKE was a much discussed topic in 2001 and 2002. Several tools and papers were written in 2001 and IKE-Crack (a tool set for this attack) was presented at Toorcon 2002.

Impact: In certain scenarios where PSK is the only authentication method used, if the attacker is able to retrieve the plaintext PSK (by cracking the captured hash), the vulnerability could result in unauthorized access to the internal network over the VPN. This will only work in circumstances with older or more simply configured VPNs; about 99% of modern VPNs use a secondary authority (hybrid mode IKE) and an attack on these will only result in a log-on prompt.

Attack Difficulty:

10 Exposed Service Version Issues (Buffer Overflow Attacks)

Definition: Exposed service version issues occur when an Internet-facing server is running an older or unpatched service that is vulnerable to a buffer overflow attack and exploit code is available.

Circa: Aleph One first published "Smashing the Stack for Fun and Profit" in 1996.[2]

Impact: Buffer overflow attacks against enterprise services are becoming rare and harder to exploit, but one can still find legacy servers that are vulnerable to these attacks. When exploitable, an attacker can compromise the system hosting the vulnerable service, allowing them to run commands on the server itself with the privilege level of the vulnerable service.

Attack Difficulty:

[2] Aleph One. "Smashing the Stack for Fun and Profit." *Phrack*. 7:49 (1996). Available at http://www.phrack.com/issues.html?issue=49&id=14#article.

Internal Penetration Test

Internal network penetration tests are performed within a target organization's environment. Historically, an internal network penetration test included an onsite visit by a tester who might be onsite from between three and 25 days, depending on the size of the environment.

Due to high demand for remote testing from our clients, in mid-2008 we replicated portions of the Trustwave Managed Security Service (MSS) environment and customer premise equipment (CPE) devices to develop a testing platform we were able to remotely and globally deploy to a client's environments. We developed this method because, though clients previously requested work performed over VPN, we could not agree to do so as we would not be able to run a large family of attacks against the internal environment over VPN, namely Layer 2 attacks (see sidebar). In 2009, nearly all of our client's internal network penetration tests were performed using our remote penetration test (RPT) appliance and infrastructure.

Like the external tests, the internal testing follows the same methodology and the results look very similar, although the scope and targets are different. From our perspective, the scope of these engagements should always originate from a network segment with the least privileges, such as a common user segment, targeting the critical assets.

How Layer 2 Attacks Work

The Open System Interconnection (OSI) Reference model describes seven "layers" of network communication. Layer 2 describes how devices talk to each other on a shared network. This network extends to any device in which packets, also called "frames" in this context, can be sent without traversing a router.

At Layer 2, the media access control (MAC) address is the primary identifier and each network card has its own unique MAC address "burned in" at the factory. Since these addresses are chosen by the manufacturer and not provisioned by the local organization their values can vary greatly, and a list of MAC addresses on any given network will likely be diverse. These addresses are also sometimes called "physical addresses" due to their proximity to the physical layer. The more useful address is the one we assign ourselves: the IP address, also called a "logical address."

Though IP addresses are used as destinations by network applications, all real communication is actually done at Layer 2 using physical addresses. To accomplish this, a protocol was developed in 1982 called the Address Resolution Protocol (ARP). This protocol is used constantly in modern networks, allowing a device to discover the physical address (MAC) of a device using its logical address (IP). After broadcasting an ARP request on the local network, the device listens for a reply. If received, the device updates its "ARP cache" with the result.

An example of an ARP cache entry would be:

IP Address	MAC Address
192.168.1.5	00:25:4B:9A:B1:EE

Layer 2 attacks involve the interception and manipulation of data at this level of network communications. Using technologies such as ARP and dynamic host configuration protocol (DHCP; used to automatically provision IP addresses), attackers can enumerate resources and fool devices into communicating with their own devices. These attacks are almost invisible: as Layer 2 technologies "just work," so do many Layer 2 attacks. The most common Layer 2 attack, called "ARP cache poisoning," works by overwriting a victim's ARP cache in order to intercept communications. Due to the dynamic nature of local networks, this attack often succeeds, allowing an attacker to read and manipulate data at the lowest layer possible, where often no security controls exist. This is almost always a precursor to more serious attacks.

Top 10 Issues Identified during Internal Network Penetration Tests

1 Address Resolution Protocol (ARP) Cache Poisoning

Definition: ARP cache poisoning, also called ARP spoofing, involves an OSI Layer 2 attack. An attacker sends out a gratuitous ARP message to one or many machines on the subnet stating that the MAC address of the subnet gateway has changed. The message will usually contain the attacker's MAC address as a substitute. When the attacker turns on IP forwarding, sent packets will all be routed through the attacker's machine.

Circa: Articles on this topic appeared around 1999. Many components of the Dsniff MITM toolkit (including arpspoof, one of the first widely known tools that exploited this vulnerability) were published in 2000.

Impact: Services such as POP, Telnet and FTP are still commonly used. Any authentication via these protocols through a man-in-the-middle (MITM) host will expose the user's login credentials. Also susceptible to a proxy style attack are older implementations of remote desktop protocol (RDP) and secure sockets layer (SSL) communications with clients using browsers with broken root certificates. This technique can lead to more advanced attacks such as the LANMAN / HALFLM challenge attack and can be used for everything from passive eavesdropping to active MITM techniques. Credential mining, session hijacking and delivering malware are all possible. Because the vulnerability is inherent to the way the IP MAC layer works, network access control (NAC) is usually the best defense against this vulnerability.

Attack Difficulty:

2 Microsoft SQL Server with Weak or No Credentials for Administrative Account

Definition: Miscrosoft (MS) SQL server may have an easily guessed or null password for administrative accounts, such as the system administrator account.

Circa: Robert Morris and Ken Thompsons first published details on this vulnerability in the 1979 paper "Password Security: A Case History."

Impact: Using the xp_cmdshell stored procedure gives an unauthorized user full system level access to the Windows operating system.

Attack Difficulty:

3 Weak or Blank Password for an Administrative Level Windows or Unix Account

Definition: Windows or Unix systems may have an easily guessed or null password for accounts with administrative pirvileges.

Circa: Addressed in "Password Security: A Case History," published in 1979.

Impact: Even on a seemingly "unimportant" system, an attacker with this level of access can read sensitive information such as the security account manager (SAM) hive, which stores users' passwords, or the shadow file. This vulnerability gives an attacker full system access to the operating system. If the system is a domain controller or back-up domain controller, it could well provide administrative access to the entire Windows Domain.

Attack Difficulty:

4 Client Sends LAN Manager (LM) Response for NTLM Authentication

Definition: Windows NT is a suite of operating systems produced by Microsoft, and NT LAN manager (NTLM) is a Microsoft authentication protocol. When exchanging files between hosts in a local area network or sending commands to a remote system, Windows uses a protocol called the Common Internet File System (CIFS). CIFS uses NTLM for authentication; this is sometimes referred to as NT Challenge/Response (NTCR). NTLM is a challenge response authentication protocol. The server authenticates the client by sending an 8-byte random number, the challenge. The client performs an operation involving the challenge and a secret shared between client and server, such as a password. The password is used twice to generate an NT password hash and a LM password hash (for backwards compatibility with older systems). The client then sends the response (hashes) to the server. The server verifies that the client has computed the correct result, and from this infers the identity of the client.

Circa: The inherent weakness of LM passwords was first an industry issue with the introduction of L0phtcrack (a popular password cracking tool of its time) in 1997.

Impact: Any number of mechanisms can "trick" a client into attempting to authenticate to a malicious server/service (e.g., MITM, DNS or DHCP attacks, embedded links in Web pages) making this vector easy to implement. If a user is an administrator of his or her own system (very common), compromise of the host is easier to accomplish and an attacker will have access to the local system, domain or domain administrator credentials. By implementing a server with a known NTLM 8 byte challenge, it is possible to perform cryptographic attacks against a captured LM client hash using a combination of pre-computed hash tables (rainbow tables) and brute force to reveal the plaintext password.

Attack Difficulty:

5 Crypto Keys Stored Alongside Encrypted Data

Definition: According to various private and public regulations, certain types of data are required to be stored only in an encrypted format. Unfortunately, the knowledge of how these crypto mechanisms work is not common and many database administrators will simply store the keys to the stored data right along with it.

Circa: This vulnerability has been accessible since the advent of cryptography, one of the oldest fields of technical study and dating back about 4,000 years. The Data Encryption Standard (DES) was invented by IBM in 1974 and subsequently led to the types of data encryption we are familiar with today.

Impact: Any attacker with basic database administrator experience will be able to determine how the target data is to be decrypted, and then modify or write appropriate utilities to do so.

Attack Difficulty:

6 Cached Domain Credentials Enabled on Networked Windows Host

Definition: When local system credentials and encrypted domain credentials for the last 10 log ons are available, these can be gathered and deciphered offline.

Circa: Active Directory was previewed in 1999 and released with Windows 2000.

Impact: Windows Active Directory feature provides an escalation of privileges, largely dependent on who has legitimately logged into the machine, but still provides an attacker who has obtained local system credentials of a machine an easy way to gain domain administrative credentials.

Attack Difficulty:

7 NFS Export Share Unprotected

Definition: Depending on its configuration, the network file system (NFS) can permit unauthorized access to file systems. NFS also depends on other systems to authenticate users and provides access according to user ID or group ID. Either of these conditions results in unauthorized access for attackers with root access to their own device.

Circa: This problem probably dates back to 1989 with the release of NFS in RFC1094.[3]

Impact: An attacker may obtain unauthenticated access to data on an NFS share. In some cases an attacker may be able to write or overwrite files on an NFS share; depending on what files are present, this may allow access to various hosts using the NFS share.

Attack Difficulty:

[3] WRFC1094-NFS: Network File System Protocol spcification, available at www.faqs.org/rfcs/rfc1094.html.

8 Sensitive Information Transmitted Unencrypted on the Wire

Definition: Sensitive information, such as CHD, PII or social security numbers, is not encrypted while traversing internal networks.

Circa: The initial public release of tcpdump, version 2.0, occurred on January 13, 1991.

Impact: When combined with a technique such as ARP cache poisoning, attackers will have access to data in the process of transmittal on a network.

Attack Difficulty:

9 Storage of Sensitive Information Outside the Designated Secured Zone

Definition: Sensitive information is stored in unencrypted files on local workstations or network file shares.

Circa: TIS, under a broader ARPA contract, developed the FWTK, and made it freely available under license on October 1, 1993. Since that time, network segmentation has been an industry issue.

Impact: Sensitive data stored, contrary to policy and/or on the wrong systems, in lower security environments allows an attacker easy access.

Attack Difficulty:

10 Virtual Network Computing Authentication Bypass

Definition: Versions of virtual network computing (VNC) are susceptible to an authentication-bypass vulnerability. This issue is due to a flaw in the authentication process of the affected package. Exploiting this issue allows attackers to gain unauthenticated, remote access to the VNC servers using the null authentication method.

Circa: The Common Vulnerability and Exposures (CVE) entry for this vulnerability was published on May 15, 2006 and updated Nov 15, 2007. The CVE number for this is CVE-2006-2369 and it is rated a 7.5 CVE score (high).[4]

Impact: This exploitable version of the VNC remote access software is still widely used in environments today, and can lead to user or administrator access to the system running it.

Attack Difficulty:

[4] For a dictionary of CVE identifiers, visit http://cve.mitre.org/.

Bonus: Additional Vulnerabilities

Missing Critical Non-system Software Patches

Definition: Many systems and services are installed using third party software that is not supported by native patch management processes delivered as part of an operating system (e.g., Microsoft Windows). Unless companies have instituted a comprehensive patch management process, such software is often overlooked.

Circa: Aleph One published "Smashing the Stack for Fun and Profit" in *Phrack* magazine 49, 1996. Patching for buffer overflows has been relatively common since that publication.

Impact: These services often run with advanced privilege on their respective hosts. Exploits of these services can lead to the complete compromise of the affected host.

Attack Difficulty:

Note: Up until the 1980s, software vendors distributed patches on paper tape or on punched cards, expecting the recipient to cut out the indicated part of the original tape (or deck), and patch in (hence the name) the replacement segment.

Missing Critical System Software Patches

Definition: For various reasons, many organizations forego the integrated automatic update process included natively with popular OSs.

Circa: This problem probably dates back to 1989 with the release of NFS in RFC 1094.

Impact: These services often run with advanced privilege on their respective hosts. Exploits of these services can lead to the complete compromise of the affected host.

Attack Difficulty:

Wireless Penetration Test

The scope of a wireless penetration test can be very dynamic. We perform these tests using direct attack-based logic to identify the real risks inherent in a wireless infrastructure and what that risk means to sensitive data stored elsewhere. Wireless testing can encompass not only modern 802.11 Wi-Fi networks, but also legacy 802.11 protocols, ZigBee, Bluetooth and sometimes proprietary wireless protocols.

Over the past few years, and in 2009 specifically, the primary focus for SpiderLabs wireless testing has shifted, along with the industry, with the realization that wireless network infrastructure and security surrounding wireless access points are not the only wireless attack vectors. Pursuit of the wireless client vector can give a bigger payoff to an attacker and this was confirmed in 2009, as the top results of our penetration tests were wireless client issues. Previously, this vector seems to have been ignored due to the countless fast ways to abuse wired equivalent privacy (WEP), a deprecated algorithm used to secure IEEE 802.11. But this past year, when we tested wireless client issues along with the wireless infrastructure, high impact issues appeared, including compromises of sensitive data even when the organization otherwise followed best practices for the wireless infrastructure, and in locations where no official wireless network existed.

The Frequency Hopping Spread Spectrum (FHSS) Myth[5]

Certain wireless technologies, such as Bluetooth and older, legacy 802.11 Wi-Fi networks operate using a wide band technology known as frequency hopping spread spectrum (FHSS). While many organizations use these technologies, few properly secure them due to misconceptions regarding the inherent security of these networks. However, this inherent security is only a myth and a prime example of "security through obscurity."

FHSS was designed during World War II as a secret communications system that could guide torpedoes to a target without being intercepted by the enemy. FHSS was based on the idea of sending radio signals from transmitter to receiver over multiple frequencies in a random pattern, making radio-guided torpedoes more difficult to detect; this idea was originally patented by movie actress Hedy Lamarr and composer George Antheil.

Using the concept of FHSS, current wireless networks typically use one of 78 different hop sequences (defined in the ANSI/IEEE 802.11 standard) to hop to a new 1MHz channel (out of a total of 79 channels) every 400 milliseconds. Synchronization information is sent between access points and stations, or between stations in an ad hoc situation.

Due to the nature of the FHSS physical layer, it is greatly resistant to any narrow band interference and narrow band jamming. FHSS security often refers to "resistance to jamming," but this is true of any wideband technology.

Many legacy 802.11 FHSS deployments can still be found among large retailers and manufacturers (e.g., warehouse facilities), and are typically used for applications such as wireless barcode scanners, wireless printers, and wireless IP phones. The most prolific brand of FHSS based 802.11 WLAN technology is Symbol Spectrum24 equipment, but one also can find RangeLAN2 adapters, and older Proxim AP and LAN adapter equipment.

Potential reasons for the maintenance and utilization of these legacy networks are:

- FHSS is seen as a security feature and the conventional thinking is that these networks are hidden from attackers and robust against eavesdropping.
- The initial deployment of these networks was a significant investment, and they still serve their primary purpose well.
- It would be another significant investment to update these networks and there is not a clear gain from increased features and/or speed.
- Updating these networks to other technologies would require an additional security investment and network architecture to increase segmentation and access control (e.g., additional routing hops, firewalls, IPS, NAC, etc.).

Many of these reasons are built on the faulty first premise that FHSS is a security feature. Whereas this might have been true during World War II, and in earlier days of 802.11 FHSS, now it can only be described as simple security though obscurity.

[5] For the detailed white paper, titled "FHSS Network Security," please visit www.trustwave.com/spiderLabs-papers.php.

Top 10 Issues Identified During Wireless Penetration Tests

1 Client Associates with Wireless Access Points While Connected to Wired Network

Definition: With rare exception, Windows clients will probe for and associate with wireless access points even though they are already connected to a wired network. It is possible to create a wireless access point, which will answer to and allow association with, any request probe received. Any client which associates with that access point can then be targeted for attack.

Circa: The hostapd tool was introduced at the 21st Chaos Communication Conference held in December 2004. The introduction of this tool widely popularized this attack vector.

Impact: If the wired side network is a corporate network, and the attacker can gain access to the client from the wireless side, this client can be used as a jump point into the corporate network for an attacker. Depending on countermeasures implemented on a laptop, this could result in an outsider having remote access to the internal corporate network. We found most organizations do not have countermeasures simply because they have not considered this vector.

Attack Difficulty:

2 Wireless Clients Probe from Stored Profiles When Not Connected (KARMA Attack)

Definition: A KARMA attack consists of an attacker issuing a bogus wireless access point that will allow association and access for any client probe from a wireless stored profile. In this way the client actually connects to the KARMA access point instead of the intended access point. If the attacker's KARMA access point has Internet connectivity and is configured to route traffic, the victim can do anything they would normally and might not even know they are connected to an attacker.

Circa: The original KARMA tool suite was released at the Microsoft Blue Hat conference in 2005.

Impact: A KARMA attack against a corporate campus, or at a nearby location where employees frequent, such as a coffee house with Wi-Fi, can be particularly devastating. Once connected, the victims can fall prey to MITM attacks that steal their credentials, download malware or a number of other MITM techniques. They then return to the corporate network with a compromised machine. This is another client side attack for which many organizations do not have countermeasures tested, and many find the countermeasures they have in place do nothing to stop this attack. Recently KARMA was added to a popular attack suite of tools to make it more automated and accessible.

Attack Difficulty:

3 Continued Use of WEP Encryption

Definition: WEP is a protocol for encrypting transmissions over IEE802.11 wireless networks. Packets are encrypted using the stream cipher RC4 under a root key shared by all radio stations. Countless security analysis of WEP shows that it is inherently flawed; one can find a publication and an exploit tool for almost every single step in the encryption process.

Circa: Stubfield, Ioannidis and Rubin published "A Key Recovery Attack on the 802.11b Wired Equivalent Privacy Protocol (WEP)" in May 2004.

Impact: WEP does not really protect wireless transmissions. In some cases an attacker can recover a WEP root key in approximately 40 seconds. WEP attacks allow–at a minimum–eavesdropping on wireless traffic and may provide the attacker with access to the wireless network.

Attack Difficulty:

4 Easily Determined WPA/WPA2 PSK

Definition: Wi-Fi protected access (WPA/WPA2) is another protocol for encrypting transmissions over IEE802.11 wireless networks. In one version of WPA/WPA2 based on a PSK, a single root key is shared by all radio stations. If the root key is simplistic, that is, if it is a dictionary word or does not include numbers or symbols, an attacker could use a dictionary attack to recover this root key.

Circa: The process for ratifying WPA and eventually WPA2 lasted from 2004-2006. The first tool for WPA cracking coWPAtty was written by Josh Wright in July 2006. In 2009, a cloud computing service to crack WPA PSKs became available.

Impact: Once an attacker has a root key they can decrypt all encrypted traffic captured over the air and, in many cases, join the wireless network and interact with it.

Attack Difficulty:

5 Legacy IEEE 802.11 FHSS Wireless Networks Implemented with No/Minimal Security Controls

Definition: The 1999 IEEE 802.11 standard specified a method to use FHSS as a physical layer for the wireless network. Formerly widely implemented, FHSS fell out of favor due to restrictions on transmission speed. Legacy networks still exist, however, often without security or access controls, or segmentation between wireless and wired networks. There is a misconception that these networks are inherently secure, but by using software radio attackers can find and connect to them the same way they connect to modern networks.

Circa: This vulnerability started to be addressed in 2004, when Cisco introduced a feature to prevent this issue on Aironet devices circa 2004. Other vendors in recent years have adopted a PSPF-like feature for access points.

Impact: Due to the architecture and lack of security controls, once an attacker connects to these networks, they have remote access to the internal network.

Attack Difficulty:

6 Lack of Publicly Secure Packet Forwarding (PSPF) Features Enabled on Network Access Points

Definition: Public secure packet forwarding (PSPF) prevents client devices associated to an access point from inadvertently sharing files or communicating with other client devices associated with that access point. It provides Internet access to client devices without providing other capabilities of a LAN. In public networks, the lack of this feature allows an attacker access to attack other wireless clients.

Circa: This vulnerability started to be addressed in 2004, when Cisco introduced a feature to prevent this issue on Aironet devices circa 2004. Other vendors in recent years have adopted a PSPF-like feature for access points.

Impact: Once connected the victims can fall prey to MITM attacks, such as those that will steal their credentials or force a download of malware. The lack of a PSPF feature could introduce virus or worm traffic to the internal network or, in a targeted attack, allow the attacker to place malware such as key loggers or connect-back shells on the compromised host allowing them later access to the corporate network.

Attack Difficulty:

7 Wireless Clients Using the Public "Guest" Network Instead of Secured Private Network

Definition: Many organizations have a "guest" wireless access point for visitors, vendors or consultants to use. Attackers can often access these guest networks as well. When employees start connecting to the guest network with corporate assets, an attacker can attack the corporate asset. Surprisingly, we find this fairly often. It usually happens when employees gather in a conference room for meetings with a vendor or other, similar situations.

Circa: This attack technique has been around since about 2003, when most corporations started introducing wireless networks.

Impact: Once connected, victims can fall prey to MITM attacks, such as those that will steal their credentials or force a download of malware. When a victim returns to the corporate network, they return with a compromised machine which could introduce virus or worm traffic to the internal network.

Attack Difficulty:

8 Lack of Segmentation or Access Controls Between Wireless and Wired Network Segments

Definition: Lack of network segmentation or properly implemented access controls means that once the wireless network is breached, it becomes a nearly trivial exercise for the attacker to breach internal network resources.

Circa: Firewall misconfigurations have existed from the inception of firewalls. TIS, under a broader ARPA contract, developed the firewall toolkit (FWTK), and made it freely available under license on October 1, 1993 (www.fwtk.org).

Impact: An attacker will have free reign to attack the internal corporate network and likely anything on the other side of a WAN connection. This is the vulnerability that allowed a major retail compromise in 2005. If there had been adequate segmentation between the WEP access point and the internal network, the possibility of an attacker cracking WEP and joining the Wi-Fi network would have been much smaller.

Attack Difficulty:

9 Wireless Device Configured to Connect to Secured Network Left Unattended

Definition: In retail environments (and sometimes in other businesses with public areas), wireless devices such as handheld devices or scanners are frequently left unattended in an easy to access place. These devices contain all information necessary to connect to the wireless network. An attacker merely has to "borrow" the device for a few minutes to read the settings or dump the configuration.

Circa: In 2000, wireless devices were becoming more common in public places. If physical access to a device was available, the configuration could be extracted, including WEP keys.

Impact: This provides the attacker with enough information to join the wireless network with an alternate device. Depending on the security controls present on the internal network, they would have access to anything internally that could be reached via Wi-Fi.

Attack Difficulty:

10 WPA Used with TKIP and 802.11e QoS

Definition: Named after the tool developers, the Beck-Tews attack is much like a known attack against WEP called "chopchop" which allows individual packets to be decrypted without cracking a WEP key. This attack is conditional as it relies on a lot of moving parts and some relatively modern equipment used with somewhat non-modern encryption (WPA with TKIP instead of WPA2 with AES). The attack relies on channels created by 802.11e QoS services to bypass replay protection. Though 802.11e was invented in 2005, it only became part of the IEEE 802.11 standard in 2007.

Circa: The process for ratifying WPA and eventually WPA2 lasted from 2004 to 2006. The first tool for WPA cracking (coWPAtty) was written by Josh Wright in July 2006. In 2009, a cloud computing service to crack WPA PSKs became available.

Impact: If this attack is executed successfully, an attacker can decrypt WPA-TKIP traffic in real-time without knowing the WPA key.

Attack Difficulty:

Physical/Social Penetration Test

Physical penetration tests are often employed by organizations concerned with threats via physical means. Our clients in this area range from large financial institutions desiring to protect data centers, to retailers looking to safeguard warehouses with inventory and/or client data.

Although this SpiderLabs service is really no more expensive than others of similar time and effort, many organizations don't feel the need to perform these tests because they either know they have serious weaknesses, or they falsely believe they do not have any risk. Unfortunately for most, this is one of the quickest paths to gain access to networks, systems and applications that allow access to the data organizations are looking to protect.

Tests we perform within this type of engagement vary widely based upon the client's needs, but often include: talking our way past security guards, tailgating staff through doors, faking a delivery (e.g., pizza, flowers, etc.), climbing walls and fences, and rappelling down skylights. SpiderLabs uses the same approach to physical security testing as for any of our systems tests, performing extensive reconnaissance and analysis for the tested organization's physical security.

Top10 Issues Identified during Physical/Social Penetration Tests[6]

1 Lack of Plate Covering Gap from Door Latch to Strike Plate

Definition: Many organizations deploy electronic locks that require a proximity card or other authentication means to release a magnetic retainer in the strike plate and open the door. However, if the latch is accessible through the door gap, these can be bypassed with a stiff card or needle nose pliers, often in a few seconds.

Impact: The lack of a cover plate over the latch and strike plate could allow an attacker to bypass an electronic lock and open a door in a few seconds. A simple steel plate can mitigate this for only a few dollars.

Attack Difficulty:

2 Motion Sensors to Allow Egress from Highly Sensitive Areas

Definition: Many organizations deploy electronic locks that require a proximity card or other authentication means to enter a sensitive area. However, a motion sensor may be used that will automatically open the door upon leaving that area. An attacker can use a coat hanger, balloon, piece of paper, or a number of materials to shove through the door gap on the secured side and trigger the motion sensor on the other side.

Impact: This could allow an attacker to bypass an electronic door lock and access a secured area.

Attack Difficulty:

[6] While the technology used to take advantage of these vulnerabilities may be new, the concept of physical security is not, therefore no circa dates are supplied in this section.

3 Sensitive Data Left in Plain View

Definition: Employees leave sensitive data on their desks, taped to a computer monitor or otherwise in plain view.

Impact: This allows anyone walking through the office (friend or foe) access to sensitive data, including passwords and PII.

Attack Difficulty:

4 Credentials or Pretext not Verified Effectively

Definition: The staff at the location does not verify a story or check the credentials before granting an attacker's request. Or staff will verify the story or check the credentials in a way the attacker suggests, such as, "I'm here to see Jim. He said to call him on his cell to verify–here's his cell number."

Impact: The impact here depends on the request being granted, but it has the potential to be major if the request is for access to sensitive systems, or for a contractor's badge with access to all areas. This could result in a compromise of the network or data, the premises or nearly any information or physical asset owned by a corporation.

Attack Difficulty:

5 Dumpsters are Accessible and Unlocked

Definition: Employees do not always follow policy for discarding sensitive documents and some organizations do not have these policies in place at all.

Impact: Information such as financial records, ID photocopies, canceled checks, signatures, network diagrams, passwords, and other sensitive data could be stolen, and data and systems compromised.

Attack Difficulty:

6 Bypass Route from the Public Area to the Secured Area is Available

Definition: An organization may have locks on all doors and in/out access control; however, there may be a path that leads from the unsecured area into the secured area that bypasses the locked doors.

Impact: This could allow an attacker to bypass an electronic door lock and access a secured area.

Attack Difficulty:

Top 5 Techniques to Unlawfully Enter a Data Center (Without being Destructive)

5 Ceiling Tiles

Get into a bathroom or any easily accessible space and push up a couple of ceiling tiles (against a wall gives a good pull up position). Once in the ceiling, and with a little careful walking, you can walk over any room you want to enter (including the data center). Just lift a tile when you're over the room you want and drop right in. As a word of caution, be careful what you use while in the ceiling to keep your balance as some conduits have a fair amount of grease on them, increasing the odds of slipping and falling.

4 Hinge Science

A common mistake in the construction of a facility has occurred when the hinges of doors (and sometimes windows) are placed on the outside (or "unsecured side") of the data center. A pair of pliers and a simple screwdriver will allow for the whole door to come off its frame. Obviously the expensive biometrics device securing the door is no longer an issue.

3 Tailgating

Probably the most common vulnerability no matter what the facility, "tailgating" is the act of an unauthorized individual gaining access to a facility by following directly behind an authorized one. The first person walks up to the door, uses his or her RFID card or key, and opens the door. The intruder will either prop the door open as it swings closed to prevent it from latching and wait until the employee has left the immediate area, or simply go into the facility directly behind the employee. Despite the huge sign saying, "No tailgating allowed," and tailgating being covered in your employee awareness training, 99% of employees will not question the person following them about having a badge or who they are. This is usually because people do not want to confront or be rude to strangers.

2 Pins and Plates

This is yet another construction problem. In a "properly" constructed data center, a great deal of money may have been spent on generators, air cooling systems, raised floors, biometrics, security cameras and computer hardware. Usually none of these items are cheap and all go toward the functionality and security of the data center. Yet when it comes to the physical lock on the door, a total of five dollars may have been spent. Lock picks are easy to get, easy to learn how to use and in most states do NOT require a license to own or purchase (despite popular belief). The five-dollar lock in the front door of the facility will take between 30 and 90 seconds for an average skilled attacker to pick.

But wait, you did spend more than five dollars and have a high security lock in your door. It even said "pick proof" on the box! But did you ever notice that thin gap between the door and its frame? Instead of lock picking, "shimming" will be used to bypass your hardened lock and door. Without a plate covering the appropriate area, a thin piece of metal or plastic (e.g., gift cards) can be inserted in that gap and then manipulated to push the door latch out of the door frame, allowing the door to swing open freely. A simple metal plate over the gap covering the area of the door latch would have prevented this–and the cost is minimal.

1 Social Engineering

The walls go all the way up to the actual ceiling, all the hinges are on the inside of the data center, the employees are running around questioning everyone about who they are and whether they have a badge, and hardened locks and plates are on every door. The last way in is social engineering.

Browse a Web page and look at the beautiful pictures showing off what kind of servers are used, what cooling systems are in place, and what generator keeps the power going. A little dumpster diving will give up the names of employees, their positions in the company, and what contracting company is used most often. A quick shopping trip to the nearest thrift store and seven dollars later an intruder now appears to be one of those contractors, or a contract network employee, or a vendor representative, or even an overnight delivery person. With a little confidence, and the right employee's name, it is possible for an intruder to be GIVEN access to the data center by the company's own employees.

The only way to combat this is common sense. Call the contracted company to see if they sent someone out, or e-mail the vendor contact and ask why there is someone representing their company onsite. And delivery personnel, no matter how bad the economy gets, will not show up for a delivery driving rental cars.

7 Motion Sensors Mounted Incorrectly Creating a Zone of No Coverage

Definition: Motion sensors have effective ranges where they can detect a moving object. If the sensor is mounted incorrectly, it may cause a significant gap in coverage where the motion sensor will not detect motion.

Impact: This could allow an attacker to bypass triggering an alarm system and gain access to a highly sensitive area.

Attack Difficulty:

8 Unlocked and Otherwise Accessible Computer Workstations

Definition: Workstations that are not locked when unattended could provide an attacker with access to the internal network, password files or other sensitive data. They could also be infected with malware by an attacker.

Impact: An attacker might use this machine to access the corporate network. This could introduce virus or worm traffic to the internal network or, in a targeted attack, allow the attacker to place malware such as key loggers or connect-back shells on the compromised host allowing them later access to the corporate network.

Attack Difficulty:

9 Network not Protected Against Rogue Device Placement

Definition: An attacker that has accessed the building in some way may try to plant a rogue device on the network to facilitate later access from a safer or remote location. The network accommodates the placement of an unauthorized device.

Impact: A device might be a wireless access point, a small computer or some other device. Without proper access controls any device can be placed on a network, expanding the attacker's effective reach and providing a back door. This can result in compromise of the internal data network.

Attack Difficulty:

10 Sensitive Data Cabling is Accessible from Public Areas

Definition: In an otherwise secured building, cabling for the private data network runs through accessible wall panels or ceiling tiles of public areas such as public restrooms.

Impact: An attacker entering the public restroom can tap into the private data network by moving a wall panel or ceiling tile, possibly resulting in compromise of the internal data network..

Attack Difficulty:

Application Penetration Test

Application penetration tests depend on the target application, but the goal of the test is always the same: find vulnerabilities in an application's interface that allows an attacker to gain access to confidential data, physical goods or even cash.

During an application penetration test, we map out the functions of the target applications, input/output mechanism and data storage and communications components. We manually test the Web-based and thin or thick-client applications against simulated attacks, testing each and every aspect of the application in scope for flaws in areas such as those listed in our Top 10.

Top 10 Issues Identified during Application Penetration Test

1 SQL Injection

Definition: SQL injection is a vulnerability that allows an attacker to insert arbitrary commands into a SQL query or statement. This attack is possible when user-supplied input is not properly sanitized before being used in a command sent to the database server.

Circa: This method of attack has existed since 1998; on Christmas Day, a security researcher known as rfp detailed the attack in an article called "NT Web Technology Vulnerabilities" for *Phrack* magazine, issue 54.

Impact: SQL injection can allow an attacker to extract data stored in the targeted database. Under the right circumstances, it can be used to modify data, execute operating system commands, read and write local files, and even tunnel internal network traffic to the Internet.

Attack Difficulty:

2 Logic Flaw

Definition: A logic flaw vulnerability allows an attacker to bypass intended application controls. An example of this is when an attacker is allowed to set arbitrary prices for goods purchased from a site.

Circa: Logic flaws have been around since the start of computing and can exist in hardware as well as software. Logic flaws first gained traction as security issues in the mid-1980s.

Impact: The impact is typically fraud related, and the fraud depends on the application. Online stores are a frequent target and exploitation typically results in theft of goods. Depending on the application, a simple logic flaw could have devastating effects on the data being used within the system.

Attack Difficulty:

3 Authentication Bypass

Definition: Authorization bypass is the result of poorly enforced authorization rules. These occur when a user can perform actions that they should not be allowed to do. For example, an action that requires a different permission level within the application. This could be a horizontal change (gaining access to another unprivileged user's data) or privilege escalation (gaining access to administrator functions).

Circa: Authorization issues for applications began with the first stateful Web applications in the mid-1990s. Later, these issues entered the realm of Web applications as the first stateful Web applications began appearing between 1997 and 1998.

Impact: Attackers can steal or modify another user's data or perform administrative actions. If they gain administrative-level access, then the nature of the exploit is only limited by the application's functionality. Usually the application and data are compromised, but could also lead to system-level in some cases.

Attack Difficulty:

Automated versus Manual: *You Still Can't Filter Out the Stupid*

Everyone wants to stretch their security budget as far as possible; in recent years, automated application security tools have become a popular choice for doing so. However, manual security testing isn't going anywhere until significant artificial intelligence innovations are made. Only manual application testing provides strong protection against modern threats, and companies that are serious about application security and have reviewed both options are consistently choosing manual testing.

Logic flaws may not get the press that vulnerabilities like SQL injection or cross-site scripting (XSS) do, but they can be devastating to an application. Every application is going to have its own unique set of logic, so it is impossible to automate tests for logic vulnerabilities. Because logic flaws often require no "hacking" skills, standard users often discover the vulnerabilities on their own. Examples from SpiderLabs' penetration tests range from the simple, such as a shopping cart application that accepts bogus coupon codes, to the very complex–sensitive information disclosure by combining query results across multiple systems.

Many vulnerabilities are simply too complicated to practically detect with an automated tool. For example, it is very common for Web applications to provide complex data structures such as serialized objects to the Web browser. Examples of such frameworks or techniques include Microsoft's .Net, Java ServerFaces, JSON, and Adobe Flex. Since a developer can place any type of data in these structures, an automated tool cannot be expected to reliably test them. Analyzing these structures can be a very complex process that requires the ability to understand the data in the context of the application.

An experienced penetration tester can identify complicated vulnerabilities in the same way that a human attacker does. A human tester has the ability to understand a developer's behavior and intent in how they created the application, and how it is designed to operate. This understanding is critical for identifying how the system can be subverted. Human testers can also work out the business logic rules, even if they are not explicitly documented. When business requirements are documented and provided to the tester, the quality of testing is even greater.

Manual source code reviews present even more benefits by identifying vulnerabilities that require access to source code. Examples include "hidden" or unused application components, which may have been left intentionally as backdoors by disgruntled developers. There are many forms of blind SQL injection with no evidence in the response, exotic injection attacks (i.e., mainframe session attacks), vulnerabilities in back-end systems, and intentional backdoors.

4 Authentication

Definition: To protect sensitive data or functions, applications typically rely on authentication controls as their first line of defense. Attackers can sometimes bypass these controls.

Circa: Authentication issues have been around since the first uses of login screens. These issues date back to early mainframe applications of the 1960s.

Impact: If the authentication controls protecting the application's interface are circumvented, the entire system including the data and user credentials could be easily compromised. Depending on the application, this could also include certain application customization that could lead to continual theft of data.

Attack Difficulty:

5 Session Handling

Definition: Session handling flaws come in many varieties. When exploiting them, the attacker's goal is to impersonate a valid and authenticated user.

Circa: The earliest mention of session handling vulnerability and predictability comes from a 1997 research paper "A Guide to Web Authentication Alternatives" by Jan Wolter.[7]

Impact: The impact is typically fraud related, and the fraud depends on the application. Online stores are a frequent target and exploitation typically results in theft of goods. Depending on the application, a simple logic flaw could have devastating effects on the data being used within the system.

Attack Difficulty:

6 Cross-site Scripting (XSS)

Definition: XSS vulnerabilities allow an attacker to insert arbitrary client-side scripts (usually JavaScript) into Web content that will be viewed by another user.

Circa: The first official CERT advisory for XSS appeared in early 2000.

Impact: Once an XSS attack has been launched, the attacker can modify the Web page in any manner. Prompting the user for username and password or for purchase information such as payment card numbers is one possibility. The attacker can also monitor user keystrokes, force the user's browser to attack other Web sites, or even tunnel network traffic through the browser.

Attack Difficulty:

7 Vulnerable Third-party Software

Definition: An application can only be as secure as the platforms it runs on (e.g., application frameworks, servers, etc). Regular patching is the best way to prevent this vulnerability, but choosing a platform with a solid security history is also important.

Circa: Third-party software vulnerabilities have been a security problem since the inception of third-party software.

Impact: Depending on the flaw discovered, the application's data is the initial depth of compromise, but in some situations, the entire system can be compromised as well.

Attack Difficulty:

[7] Wolter J. "A Guide to Web Authentication Alternatives." 1997; last update 2003. Available online at http://www.unixpapa.com/auth/homebuilt.html.

8 Cross-Site Request Forgery (CSRF)

Definition: CSRF is an attack that allows a malicious Web site to force a legitimate and authenticated user to execute commands on the targeted Web application. This is possible when the command is formatted in a predictable manner known by the attacker.

Circa: In 1988 Norm Hardy published a document explaining an application level trust issue he called a "confused deputy."[8]

Impact: Allows a malicious Web site to force a legitimate and authenticated user to execute commands on the targeted Web application, leading to any level of compromise depending on the command executed.

Attack Difficulty:

9 Browser Cache-related Flaws

Definition: A browser can potentially store sensitive data in its history or file cache. There are a number of ways to prevent this, but they must be explicitly implemented by the Web application.

Circa: Browser caching flaws have been security issues since Web applications started hosting sensitive content. These vulnerabilities were recognized as security issues with the first commerce applications in the late 1990s.

Impact: An attacker with access to the user's browser may be able to retrieve sensitive data such as passwords or payment card numbers.

Attack Difficulty:

10 Verbose Errors

Definition: Verbose error messages, which have long been an issue for Web applications, client applications and terminal applications, can provide significant aid to an attacker. These error messages could include sensitive information to how the system has been configured or even a glimpse into the source code being used by the application.

Circa: These flaws were viewed as security issues in the 1980s but were not taken seriously until many years later.

Impact: Vulnerabilities that would otherwise be impractical to exploit can suddenly become trivial. With each error message the attacker could learn more about the application, eventually resulting in a compromise of the application itself.

Attack Difficulty:

[8] Hardy N. "The Confused Deputy." 1988. Available at http://www.cis.upenn.edu/~KeyKOS/ConfusedDeputy.html

Bonus: Additional Vulnerabilities

Secure Sockets Layer (SSL) Misconfiguration

Definition: SSL provides excellent encryption, but only if it is configured correctly. Using weak cipher suites, SSL v2, or improperly signed certificates are the most common examples.

Circa: In 1995, when SSL v2 was released, there were a number of known flaws which lead to the development of SSL v3 being released in 1996. Unfortunately, a number of early adopters built application transport security around SSL v2 and we still find it in use today.

Impact: An attacker may be able to launch a MITM attack without being noticed by the user. Depending on what version of SSL is being used, data can be compromised. Additionally, if SSL is being used to encrypt a VPN tunnel, the entire network could also be compromised.

Attack Difficulty:

User Name Enumeration

Definition: Discovering valid user names is often the first step in launching attacks against an application's authentication. This can be made much simpler by improper log-in error messages or related flaws.

Circa: This vulnerability dates back to mainframe systems of the 1960s. Username enumeration vulnerabilities are still extremely common in modern Web applications.

Impact: Once a valid user name is identified, an attacker can attempt to log in with common passwords or use a social engineering attack to gain access.

Attack Difficulty:

Penetration Testing versus Vulnerability Scanning

A number of different security standards now require penetration tests and/or vulnerability assessments be regularly conducted against infrastructure and applications. But what is the difference between these two activities? For a member of the finance team, the quick answer would be "cost." They would be right, too: conducting a properly defined penetration test can be expensive when compared to a similarly scoped vulnerability assessment.

But the key difference is exploitation. A vulnerability assessment identifies potential flaws without actually exploiting them. Instead, vulnerability assessments are the sum of automated tests, which can be conducted quickly and cost effectively. These tests are done using signatures, much like anti-virus, which makes vulnerability scanning great for finding issues with commonly deployed products, but less useful at finding issues within an organization's custom solutions. Vulnerability assessments are useful tools for frequent automated scans.

The goal of a penetration test is to find and prove that there is a security deficiency by exploiting flaws and gaining access to the target systems, applications and/or data. Exploiting the flaws allows information security investigators to:

- Prove that the vulnerability is not a false positive (Was exploitation possible?)
- Properly understand the potential impact of the vulnerability (Did exploitation give access to the general ledger or last weeks' lunch orders?)
- Join multiple "low risk" flaws together in a chained attack where the overall impact is greater than the sum of its parts
- Mix technical and non-technical vulnerabilities (e.g., physical security weaknesses and poor encryption)

Penetration testing and vulnerability assessments are both useful tools in the information security teams' arsenal. By understanding the benefits and limitations of these tools, information security teams can ensure that their testing regime is effective.

Global Remediation Plan

Whether the reader is the CISO of a global organization with tens of thousands of employees and a very large security budget, or heading up a six-person startup with a half-dozen systems running open source software, a compromised environment can have the same effects: loss of business and reputation.

At SpiderLabs, we have seen firsthand system compromises of large multinational corporations by their own oversight. An organization with thousands of locations and a large IT security staff was once compromised by a wireless access point that was thought to have been overlooked during a WEP to WPA upgrade. We later learned that they knew about the issue, but just didn't think it posed enough of a risk to warrant replacement until other "more important" upgrades could be completed at the same time.

In 2009, we worked with a global financial institution that had built and released a new Web-based application for high-end account holders. The client wanted to test system security before production, but ran behind on the development cycle. Senior management decided to allow it to go live and assume the risks until it could be tested a few weeks later. This portal allowed their account holders access to account balance and transfer funds, as well as other special proprietary features. The portal even had a multilayer, two-factor authentication system: a username and password were required, the client needed to respond to obscure questions based on his or her recent and past financial history, and then a phone call was placed to a number on file and the client, when receiving the call, entered a PIN. This may seem like a robust solution, but when we tested the system we quickly found a major issue. A member of the development team left a directory called "temp" after it was deployed to production. In this directory was the source code of the entire application. Within the source code, we were able to find a way to bypass the "forgot password" functions and reset any account holder's password, dial-back number and even their PIN. Clearly the design and development team spent a lot of time on the authentication solution to ensure their clients' accounts (and money) were safe, but someone not aware of the security implications was given the task of pushing the application to production.

We have hundreds of stories just like this from 2009 and the theme is the same across the board. Organizations large and small are often focused on business growth and very rarely on continually reviewing and monitoring their information security posture. They are leaving holes, some as old as their business, with little understanding if, why or when these vulnerabilities should be fixed.

Prior to 2009 (and even most of 2008), most of the organizations that were compromised seemed to have been victims of attack due to random chance. Today we are seeing a serious uptick in targeted attacks. The attackers are learning through practice about weaknesses in a vertical market. This evolution allowed the hospitality industry to be hit hard. The attackers first learned about both an attack vector and method of data compromise in mid-2008 and perfected it before using it on a massive scale in 2009. Attacks such as these are still a serious threat, as there are tens of thousands of opportunities for attackers in this industry.

While we found the hospitality industry to have its own flaws in the design and implementation of its systems, this industry is not dissimilar from the rest of the world. In 2010 and beyond, another industry will be exposed because of similar problems and the attackers are going to swarm and pillage until the problems are fixed.

Intersecting Investigations with Proactive Penetration Testing

Globally, we all suffer the same problem–security is easy when applied at the micro-task level, given the right skilled and educated implementer, but very difficult at the macro level. Humans make mistakes or get distracted and something falls through the cracks. Where this crack intersects with an attacker's line of sight and motivation, a compromise will almost always occur.

The goal of any organization should be to minimize those cracks in daily operations, but also understand that cracks will happen regardless. It is the organization that both proactively detects and fixes those cracks while having a robust (and tested) reactive plan in place that will most certainly have a better chance of protecting its sensitive data.

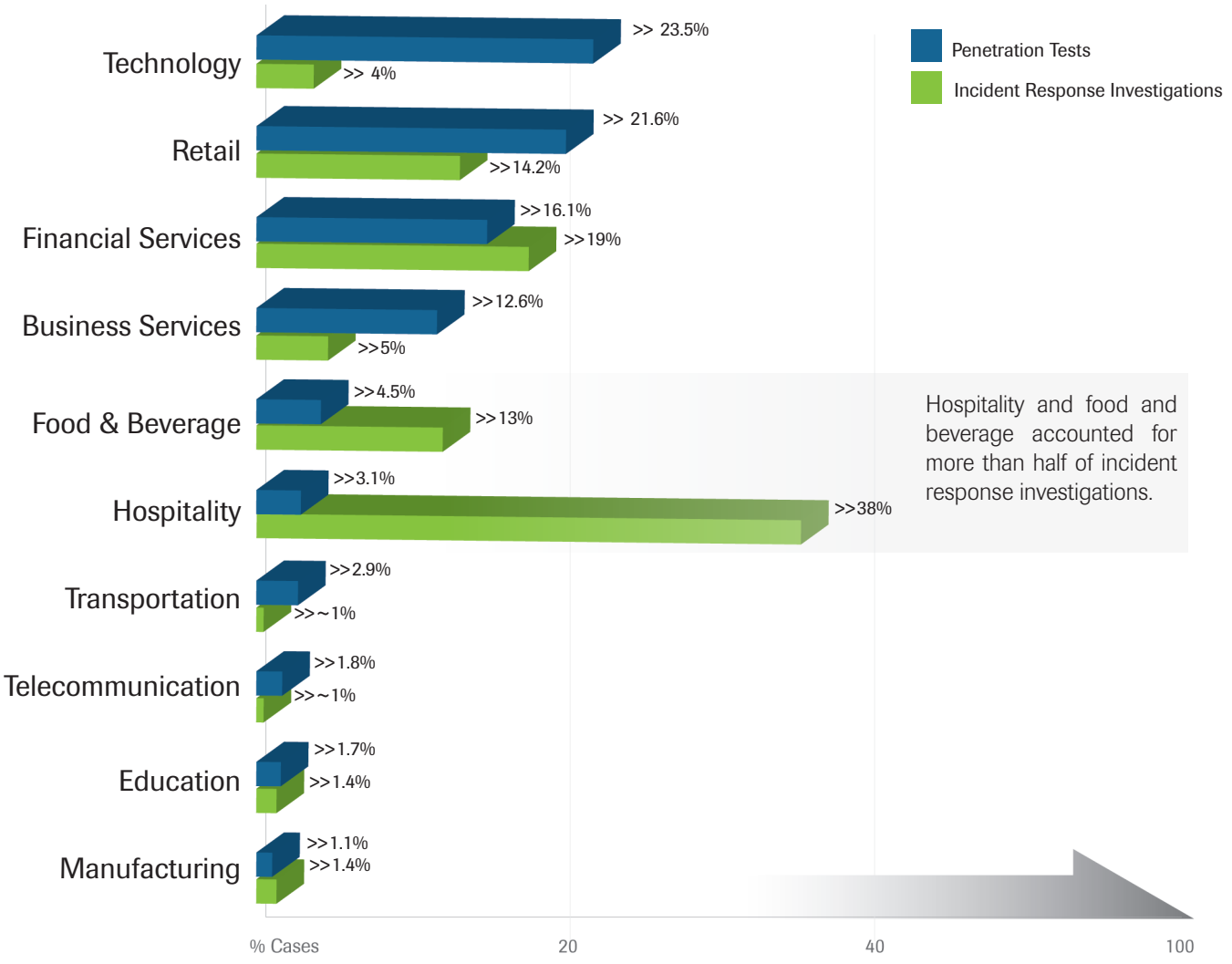
In practice, we recommend that all organizations spend time in 2010 taking a step back from their implementation plans to identify what security measures are needed for the future, and more importantly, review the current or legacy systems or processes to determine where vulnerabilities and threats already exist. We encourage businesses to take control of their security destiny, focusing on the real needs of the organization for protecting both sensitive information and reputation.

In the next section, we define a list of 10 strategic issues that we feel organizations should focus on before jumping to the next level of IT implementations.

When we reviewed top issues found in cases where our clients were proactive in their approach to identifying problems, a trend

was very clear. We have vulnerabilities in our lists that are over a decade old. During the debriefing meeting with the clients, most, if not all, were completely unaware of the vulnerabilities existing within their security posture.

Due to the gravity of the issues, a large number of those clients will struggle to fix the problems we identified because many have built an internal structure that makes it difficult for anyone to take control of the situation. In our work we have witnessed major, possibly business-ending problems, with rather simple fixes, take large organizations up to two weeks to fix, and months or years when identified by an internal IT employee. Generally, when issues are identified by a third party, the response is swifter, but at times not quick enough.



On the reactive, investigative side, we have seen internal employees reporting issues that remain unresolved until a major compromise results, which are often reported one to two years later.

Combined, both technology and business services sector clients make up 36.1% of our penetration testing clients, but only 9% of our compromise investigation clients. Hospitality and food and beverage make up only 7.6% of our penetration testing clients, but a stunning 51% of our investigation clients. Clearly, a proactive approach in security testing is proving beneficial for industries and businesses that are adopting it.

What Every Organization Should Fix Now

To avoid becoming a statistic in our 2011 report, organizations should balance their current initiatives with the security strategies outlined here.

10 Strategic Initiatives for Every Organization

1 Perform and Maintain a Complete Asset Inventory, Decommission Old Systems

Why: When we perform a penetration test for a client, we frequently provide them a more accurate inventory than they had on file, as well as find systems that the client didn't know it had on their network. Keeping an updated list of IT assets should be a priority; this will aid in the tracking and decommissioning of older systems. We often find major vulnerabilities associated with these, but clients tell us they are not concerned about the vulnerabilities because these legacy systems have a planned decommission date. Coincidently, many of our clients use us to re-test their environments in subsequent years. We find that about 75% of our test results that included client responses of "system will be decommissioned" still have those same systems in production a year later.

How to: The type of information that should be in this list should include at a minimum: name of device, DNS names, type of device, operating system, IP address(es), MAC address(es), date of installation, and owner. Once an asset list is established, all adds, deletes and changes should be logged so an up-to-date list can be obtained at any given time. For smaller organizations, this doesn't need to be a complex GUI-based system; it could be a spreadsheet, but it must be maintained.

For decommissioning, establish an internal team with cross-competency work and tackle this problem.

2 Monitor Third Party Relationships

Why: In 81% of cases, third party vendors and their products introduced vulnerabilities, mostly as a result of default, vendor-supplied credentials and insecure remote access implementations.

How to: Choosing a platform and vendor with a solid security history is important, but monitoring those vendors to ensure they are following the same security practices as the hiring organization is equally important. Organizations should also ensure contracts with third party vendors include security control requirements. If a vendor will not agree to security requirements, seek out a new vendor who will be responsive to the security needs of the organization.

3 Perform Internal Segmentation

Why: In 2009, hundreds of enterprises and small organizations had completely flat internal networks. On the enterprise side we are not referring to single locations with a single flat network, but entire global networks, where any device at any location could talk to any other device. This posed a very large problem for the hospitality industry where a single intrusion allowed an attacker to propagate to dozens of other locations. If access controls had been in place between network and segments, the attacker would have likely been stuck at the single location in his or her efforts.

How to: Organizations should work to segment their network into as many "zones" as feasibly possible. This means not only separating users from server segments and allowing only specific traffic between those segments, but also further segmenting servers by separating systems that store critical and sensitive data from those that do not. Special attention should also be placed on the areas of the network that have weaker physical controls, such as public kiosks, public Wi-Fi, and guest Internet access.

4 Rethink Wireless

Why: Wireless is everywhere. Rarely do we work with an organization that does not have wireless in some portion of their environment. We often work with early adopters of this technology, who placed the access points inside their network so that employees could access resources without having to be tethered to a physical network jack. Even with the latest wireless security applied to the implementation, organizations have found this to be a very fast moving target. As ways to crack or circumvent the security controls in use are discovered, access points need to be upgraded faster than many organizations are capable of successfully performing.

How to: We recommend organizations never place wireless access points within their corporate core network; instead, they should treat them as any other remote access medium. Users are able to use a wireless access point at a café or hotel and securely connect back to corporate resources, so they should use the exact same process when they are in the office. The wireless access points should be placed outside the network and any security controls in place should keep unwanted visitors from using a company's Wi-Fi as an open access point.

5 Encrypt Your Data

Why: With all the focus on data compromises in the last few years, one would think that organizations would have a better handle on data encryption, yet this is not the case. We have found clear-text sensitive data in all types of places. We found it where expected, such as fileservers and other databases, and also where least expected, such as Web server and application logs. In addition, clear-text data on internal networks is a large problem when an unauthorized user or employee starts monitoring the traffic. We often obtain data from our internal penetration tests by "sniffing" traffic.

How to: Understand where data is located, purge what isn't needed and encrypt the rest (including data while in transit, regardless if it is being sent to a third party, or across the internal LAN/WAN). Also, don't try to internally create an encryption solution. Evaluate vendors, run a pilot, have a trusted and knowledgeable third party validate and then implement across the enterprise.

6 Investigate Anomalies

Why: Many times we hear from compromised companies that had an outage or server crash months before a breach was detected. During review we find that a system or network administrator focused on fixing the problem, noted some odd network traffic or files on a system, but wrote them off as benign because of the pressure to restore services for the organization. Very often, no follow-up is performed promptly until a compromise occurs and subsequent investigation is performed.

How to: Look at every anomaly with a degree of suspicion and if the path of investigation leads to an unexplained occurrence, bring in a trusted expert to assist you or, at the very least, to determine if further investigation should be performed. A 10-minute conversation or log review by an expert, could save an organization millions of dollars in damages should the anomaly be something more serious.

7 Educate Your Staff

Why: Security awareness training is not a silver bullet and isn't going to stop an insider with malicious intent, but it can mean earlier notification and detection of a potential incident. Even an entry-level employee may notice something if trained to be security aware. This is especially important when working to combat the physical and social threat.

How to: Organizations should look to implement a security awareness training program and make it mandatory for each and every employee, regardless of function. Repeat this training on an annual basis and make it part of new hire orientation.

8 Implement and Follow a Software Development Life Cycle (SDLC)

Why: We often train developers on how to code their applications securely or debrief them on the results of an application penetration test. Through this work, we have found that the majority of organizations spend a great deal of time in the planning and implementation phases, but not a lot of time in the analysis, design and maintenance phases. They quickly go from idea to code to production, usually in a single threaded fashion. This means that a single individual makes both tactical and strategic decisions on their code, without input or oversight from others internal or external to the organization. When not properly implemented, a simple module like "reset my password" could result in major consequences to the security of an application and, potentially, to the entire organization.

How to: Implementing a comprehensive SDLC process which, from the start, includes security planning, review and testing is crucial to successfully developing secure applications. Many organizations hire SpiderLabs to perform an application review and penetration test—only after they have already gone live with the application. This far too often results in our discovering major security design and/or implementation flaws that should have been identified and resolved early in the SDLC process—and certainly long before the application was ever to be put into production. Organizations should review their current development methodology, and make the necessary modifications to ensure that security is not simply addressed as an afterthought, but rather as an integral and indispensable part of their SDLC process.

9 Lock Down User Access

Why: End users' ability to use a corporate-issued workstation as their personal computer poses a major risk to the security of an organization. Simply put, individuals should not be given the ability to install any software they download from the Internet.

How to: Perform an analysis of role and access privileges for all the various users and groups within your organization. Most people, you will find, including executives, do not require all the access rights they have been granted. Lock down their access, and only increase it for a valid business requirement. This is not going to prevent a breach outright, but it will limit the exposure should an attacker gain access to one of these accounts and/or systems.

10 Use Multifactor Authentication Everywhere Possible

Why: If we could, we would declare single-factor authentication dead in 2010. But we can't. There are likely 10,000 applications that use single-factor authentication for every one using multifactor. Unfortunately, when given the choice, humans often create poor (weak) passwords. Even employees within the security industry—those that should know better—often choose weak passwords to protect their systems for one simple reason: strong passwords are harder to remember. And as the number of network devices grows, the more difficult it is to keep them all straight.

How to: Multifactor authentication does not work everywhere, but should be strongly considered where it is possible. The cost of implementing a multifactor solution is far less than the impact of a major breach of the corporate network and loss of critical data.

For 2010, all organizations should review their own information security infrastructure, paying particular attention to how data flows within the organization, what security responsibilities are assigned to IT staff, and what vulnerabilities may currently exist within the organization. Organizations should review their security posture form a historic vulnerability standpoint as well, as attackers are coupling old methods with new to infiltrate systems and steal sensitive data.

Attackers have been shifting their focus from opportunistic to targeted attacks over the past year in strong numbers. Attackers will always gravitate towards the most valuable data, putting organizations that store, process or transmit credit card and other financial data at a higher risk of a targeted attack.

By applying the protective information security methods detailed in this report, organizations will be able to reduce their risk to compromise and protect both sensitive data and company reputation.

Contributors

Authors

Nicholas J. Percoco (Lead)
Rob Havelt
Charles Henderson
Colin Sheppard

Content Contributors

Marc Bown
David Byrne
Ryan Jones
Steve Ocepek

Contact Us

For comments or questions regarding this report please contact SpiderLabs directly at the information listed below.

To request information about our services for environments or applications, we at SpiderLabs are available to discuss any organization's needs.

+1 312 873-7500

GSR2010@trustwave.com

https://www.trustwave.com/spiderlabs

Twitter: @SpiderLabs / @Trustwave

About Trustwave®

Trustwave is a leading provider of information security and compliance management solutions to large and small businesses throughout the world. Trustwave analyzes, protects and validates an organization's data management infrastructure—from the network to the application layer—to ensure the protection of information and compliance with industry standards and regulations such as the PCI DSS and ISO 27002, among others. Financial institutions, large and small retailers, global electronic exchanges, educational institutions, business service firms and government agencies rely on Trustwave. The company's solutions include on-demand compliance management, managed security services, digital certificates and 24x7 multilingual support. Trustwave is headquartered in Chicago with offices throughout North America, South America, Europe, the Middle East, Africa, Asia and Australia.

Corporate Headquarters
70 West Madison St.
Suite 1050
Chicago, IL 60602

P: 312.873.7500
F: 312.443.8028

EMEA Headquarters
Westminster Tower
8th floor
3 Albert Embankment
LondonSE1 7SP
P: +44 (0) 845 456 9611
F: +44 (0) 845 456 9612

LAC Headquarters
Edificio E-Tower
Rua Funchal, 418—35 Andar
Vila Olímpia—São Paulo—SP
CEP 04551—BRASIL
P: +55 (11) 3521-7314
F: +55 (11) 3521-7070

APAC Headquarters
Level 26
44 Market Street
Sydney NSW 2000, Australia

P: +61 2 9089 8870
F: +61 2 9089 8989