Global Information Assurance Certification Paper

Preparation@Incident Response.security

By Dan Widger

Submitted: 6/15/04
GIAC Practical
For GSEC Certification

ABSTRACT:
Of the six main steps involved in Incident Response, Preparation is arguably the most important. When a security event occurs, incident responders require a wide range of information and resources which will equip them to assess the event. An orientation to the affected application/system is required, as well as information regarding what security considerations were built into the application. In order to have all of this information available when needed, it will require contributions from the applications development group, as well as the IT operations group, in the form of planning and appropriate documentation. Within this document a range of topics will be covered that should help establish policy, as well as a framework around an application document set, all of which will aide the incident responder in responding more effectively to a security event.

It's 2:00 in the morning, and the phone rings. There's been an event, a network anomaly. A critical application is no longer working, and the suspicion is that the host was hacked. As the designated security agent of the moment, you're responsible for response for a large enterprise organization, with single digit divisions, triple digit sites, triple digit servers, and thousands of workstations. Some application somewhere no longer performs as expected, and you, the security agent, are struggling to determine where to begin. This document is about the factors that could possibly prevent a security incident, or if it occurred, could reduce the impact, and return systems to normal operations with minimum effort.

To set the stage for the broadest audience of IT readers of this document, Incident Response is a guide to responding to security incidents. Its very easy to see in hind sight what was a security incident after the incident has occurred. Many readers will be able to vividly recall events that were clearly security incidents like Blaster infection, Slammer infection, Code Red, Nimda, and perhaps even a critical corporate web server being attacked. Other less obvious security events have occurred with little or no fanfare, like the email harassment of a employee using corporate email, the employee who got a little into internet smut while at the office on corporate equipment, or the employee who was able access in-appropriate data, and bragged to the wrong person. To respond to incidents like these, there are 6 steps or phases[i] to the investigation, or incident management. These steps are:

- Preparation
- Identification
- Containment
- Eradication
- Recovery
- Lessons Learned / Follow up

It is from events like these, that a security incident handler begins to build a mental checklist from the lessons that you'd only like to learn once. This paper is not about the steps (2-6) of incident response. It's about the first step, Preparation, which is all about an intentional methodical approach to security. This paper will intentionally overlook the specific elements of creating a CIRT team, or the tools and gadgets that go into the black bags or jump bag of those who respond to security events.

This paper will approach the security from the perspective of the whole system which I will refer to as the application, which processes the data set for a given client. An application is composed of an executable (or set of executables), which processes data, on a given (set of) host(s). In the given illustration of a security event, where the first sign of impact is on a specific application, which resides on a specific server, but it is foolhardy to assume that this is where the extent of the damage is limited to. All upstream and down stream processes need to be understood to examine what else may be impacted. I think the most accurate example to reflect a potential security incident would be the potential case of an anomalous event occurring against a web server that delivers sensitive data to an important audience.

One of the most useful tools that would assist a security agent at the time of a security incident would be an Application handbook. Don't limit your thinking to the paper equivalent, but think instead of what format is most utilizable in your in your environment that would contain the basic data sets that will be described here. Perhaps the format for this may be a folio CD, or a database for a PDA, or maybe it's a database that resides on your laptop that gets sync'd up automatically every day you connect to the corporate network. The focus is much less about the media, but the content itself. How many hours of security events are wasted as people do initial security event assessment, just to assimilate the core of what this paper discusses.

To prepare this kind of Application summary, this paper (Preparation @ Incident Response.security) advocates that this should become a responsibility of all projects for all application implementations and enhancements. I believe that Security (or Risk Management protocols) development and operations practices should be woven into the fabric of the process and project implementation methodology of the organization. In essence, Plan for Failure. To facilitate continuity, account for the potential of discontinuity. If this application summary is presented to the implementation team the day before the application goes live, they will never complete this. The recommendation instead is that the different parts of this be completed at different points in your organization's project management / implementation methodology. When security (a.k.a. vulnerability remediation or risk management) is baked into the productive bread, it is saturated, integrated, and produces a higher quality product, instead of being added (like butter or jam) as an afterthought. Security should be "baked in" like it was an original ingredient, not added to like an afterthought to the original product.

I will attempt to summarize the proposed components of an application summary, for the purposes of an Incident Response. The components will have direct impact on different groups within an IT organization, like application development and Operations, but the crux of Incident Response is having this information available for the Security practitioner at the time of need.

What data is processed in this application?
- **How valuable is it?**
- **Does it merit performing an Incident Response?**

What is the security posture of the application?
- **What security mechanism(s) are in this [application / data / host] system(s)?**

What are the historical security patterns of this application or server?
What security mechanism(s) are in place outside of affected [application / data / host] system(s)?
What is this application / server supposed to look like?
- **Can a comparison be assessed against current profile and original profile?**
- **What information will be needed in an Incident Response event?**

How fast and when should the [application / data / host] system(s) be re-constituted?

I adhere to the school of thought that the different groups in IT should consider the elements of this Application Summary in light of how the prescribed elements will contribute to quality execution of their priorities in their realms of practice. I believe that if they include these policies in their environment, it will benefit their members and priorities and will also advance the cause of the Information Technology services in their organizations.

Data Assessment

I propose that any given IT environment be categorized by the data sets that are processed within it. In preparation for security or business continuity (which is what this paper is all about), a clear ranking of all data sets by class should be identified[ii]. For the purposes of the application summary, this could be a multi-dimensional matrix or database where the different components are associated with the applications that process them and the servers that host the applications. Other elements may be useful to include as well, such as time zone, geography, or other critical factors. From this, the data assets that bring maximum impact to the organization can be ranked. I prefer a three tier ranking for the sake of simplicity. In on chapter of my experience, the rankings (from highest to lowest) were 1) Critical, 2) Urgent, and 3) Important. With this ranking system, all customers would agree that even the least impacting processes were "important", but everyone knew to drop everything when there were issues with the company's "critical" processes.

Based on my own experience as a security practitioner, the most valuable asset in electronic systems is the data that is processed. In some environments, the frame of reference is the server or maybe the process component (a particular piece of software or maybe a communications piece) that breaks most frequently. However, for purposes of this paper, I will suggest that the primary point of reference be the complete application that processes a given set of data. The OS can change, the hardware can change, the application interface (installed client executable or web interface or other) can change, but the constant is the business data and the client set who interacts with it (e.g. the accounting department will always interact with the accounting data, no matter what platform is used). The critical characteristics and the legal obligations are not likely to change, regardless of what platform(s) are used for presenting this data to the clients.

To consider the value and merit of the data involved, one needs to consider who owns the data, what type of data that it is, and what protections are appropriate for the data. Consider the life cycle for the data management process in terms of a software development process, wherein someone prepares a specification for a new application or for a new release of an existing application. Do the parties who write the functional specifications of the applications understand the merits and obligations that accompany the data?

Are the parties responsible for the data aware of legal or contractual obligations to protect the data? There is an increasing degree of regulation in our current business

environment. There are U.S. regulations which require protection of employees, and consumers. There is even increasing protection from state regulations (e.g. California SB1386) that mandate new protections for consumers data. There is increasing international legislation, the Basel Accords in Europe for example, that explicitly mandates data protections and even require defined incident response processes[iii]. There was legislation in Europe in the last weeks of April, 2004, that obligated corporations to turn over incident response information to non-government authorities, such as auditors and insurers, upon request[iv]. As a result of the rapid progression of legislation that requires protection of personal data and even information about cyber security events and how they were handled, parties who bear responsibility for data management should reassess their responsibilities for protection of their data on an annual review.

For the purposes of Preparation@Incident Response.security, how is this organization's data classified and can this be derived (vertically) by application and/ or (horizontally) by host? Optimally, the security practitioner and the associated incident handling team should always begin every incident with an orientation as to how critical the data set is to the organization. Without some benchmark of the data's importance, the application designers, the developers, and the operations staff won't place appropriate protections around the application, the operational equipment supporting the application, and ultimately the data itself.

Responsible Parties

Who owns the data? In some organizations this is unclear perhaps because the question has never been asked. From personal experience, I believe that in most cases the "owner" or primary custodian of the data is the business department that is the primary user or the primary originator of the data. So the questions that should be asked from a business perspective include:

- What department (or organizational entity) is the designated "owner" of the data?
- Who is the designated senior manager of this department?
    - Is contact information available for this person?
- Who has the senior manager designated as the operational administrator for the business unit for the data and the application that manipulates it?
    - What is their contact information?

Similar to the legal role of an attorney representing a client as their agent, in many business environments there is a designated IT agent who represents the interests of the business client. A question to be asked in every organization would be: Who is the IT responsible party that represents the interests of the internal business client for this application? This, of course, depends on the culture of the company. From personal experience, the "Agent" for the client / data owner has been the IT Business Liaison, sometimes called the "Systems Analyst", or perhaps internal applications manager, who represents the interests of the (internal) business client to the IT (development) staff. This "agent" has general responsibilities to interpret client requests to IT and to ensure that IT services are delivered consistently within defined parameters (Service Level Agreement) to and for the clients.

In the event of a crisis, it is necessary to have a calling tree for each application. A calling tree is comprised of contact information for technical staff who are able to diagnose and remedy problems, middle management staff whose operations are affected, and finally, senior management staff who are responsible for strategic decisions regarding applications, data, or systems.

Security Posture of Application System

For purposes of this discussion, we must consider the difference between "functional" and "secure". In functional delivery the effort was, "I did enough setup so that it works". In a secure delivery the effort would be, "I built it securely so that it is stable, provides consistent service to only the appropriate clients with enough protections to withstand a range of likely (and some unlikely) threats."

We can begin by reviewing the elements that affect the security of the host O.S. Assuming that the O.S. is running in its default state, a critical concern would involve the "hardening" of the O.S. While this may change in years to come, virtually all commercial O.S.'s are insecure in their default state. Therefore, most O.S.'s should be 'massaged' or re-constructed in a more spartan fashion to provide a secure level of operation without the distraction and risk of unnecessary services. Have the host(s) associated with a given application been appropriately hardened for a particular level of service? As an illustration, have the hosts providing web services been hardened suitably for service in a DMZ where it is exposed to a hostile environment, or was it hardened for service inside a data center within the confines of a corporate enterprise environment? Was the hardening process using some external benchmark (NSA's Windows Guide) as a guide or did it use some internally developed standard?

In addition to hardening, it is necessary to insure that all security mechanisms for the host are kept current and fresh. If a given host is hardened, did the process include vulnerability scanning by a currently effective tool? While much credit is currently attributed to vulnerability scanning tools (or other security tools or processes), there is a strong dependency on the skill of the parties operating scanning mechanisms (or security tools), the freshness and comprehensiveness of the (scanning) policy, and of the (vulnerability) remediation process.

Frequently, it is the engineers who posses the skill sets of vulnerability management (e.g. vulnerability scanning, anti-virus, and patching) who make up part of the incident response teams. These engineers must maintain current knowledge about threats and defenses (patching) so they can help identify and remediate the risks, even preventing the potential of a security incident. When an engineer or security practitioner takes a leave of absence (for vacation, illness, etc.), there is a period of time where the engineer must be immersed back into the current flow of threats and vulnerabilities. For this reason it benefits the engineer and the organization that the engineer supports for there to be subscriptions to security content material which updates them on new vulnerabilities and new exploits. Technical security training and conferences are highly important aspects of keeping the engineer abreast of threats, exploits, and defensive

solutions. Likewise, it is important to maintain current maintenance and subscriptions on all security tools.

One element of vulnerability management is the process of patch management. Patch management has evolved into an effective means to counter the current threat of buffer overflows and other vulnerabilities associated with common O.S.'s and primary applications like messaging, databases, and web services. Valuable lessons have been learned over the past several years, including patching the O.S. in addition to the primary applications (like SQL Services, web services, etc.). But as poignant as these lessons are, they are different threads of the same cord of vulnerability management. If there is no repeatable process to fix the vulnerabilities identified, the benefit of vulnerability scanning is negated. For vulnerability management to be effective, it requires an effective process utilizing skilled staff, implementing effective policies, which have current data. Experience has shown me that patch management is most effective when patching is implemented quickly and on a routine basis. If O.S. and primary application patching is applied at random intervals, such as when the whole system is scanned for vulnerabilities, then the window of risk (difference in time from when a risk is identified until the risk is mitigated by patching) is bigger than necessary. My conviction is that patching should occur at least monthly and vulnerability scanning by application systems should occur periodically (approx 1-3 times per year).

In addition to patch management, the practice of security monitoring should also be in place. Host-based security monitoring (a.k.a. host IDS) should be implemented, particularly where transport encryption (like SSL) is utilized. Host IDS directly analyzes for current threats and is able to respond directly to threats that have exact matches. However, a newer technology is coming which monitors for malicious behaviors and is able to protect the host based on the type of behavior instead of just the "signature" of the attack. This behavioral component is evolving rapidly and includes buffer overflow protection, however, it may take several more years for this technology to reach the effectiveness of the current signature based tools.

Within most local security monitoring and protection tools, there is a form of event logging[v]. Another security process that can add value is the analysis and correlation of the event[vi]. The correlation function , when applied properly, should sift through many events and consolidate them together into a minimal number of occurrences which require investigation.

To support effective logging and correlation, there should to be a central time source. The firewall, security monitoring (IDS), and logging by operating systems and applications should be using a single reference point. I'm familiar with a client who tried to identify where Nimda first entered a corporate environment where the firewalls used a single NTP time server from the internet, the routers and infrastructure equipment used a different NTP source, the Unix systems used another time, and the Windows systems used yet another time source. Obviously, in this case, identifying the original source of infection was impossible. It is important for all systems to be set on single time standard, for example Central Daylight time. If the systems affected where

dispersed across time zones, events would need to be adjusted appropriately against a standard time clock. For an environment to prepare for this time element, a prescription could be made for an investment in a "trusted time source", protected and enforced by policy and procedures, and some time spent to identify time utilities that could be used for diagnostics and forensics.

Elements of the architecture of application software can also affect the posture of an application system. If security was a criteria for the processing of the system, many elements could have been designed into the application. The 3 A's (Authorization, Accounting, and Access control) can be woven into the fabric of the system much deeper if it is integrated into the functional specifications of an application at the beginning. When security criteria is included in the application specifications, it results in a hardening of the application. For example, careful and methodical implementation of other security techniques such as encryption and logging can contribute to securing sensitive data and accounting for accesses and transactions.

Similar to the vulnerability scanning function of the O.S., there are new technologies becoming available which can scan application environments for vulnerabilities. Tools like Sanctum's AppScan and Kavado's ScanDo will scan web applications for vulnerabilities. These are evolving from a static tool that is used at the stage after development and before production to a new role wherein they can be integrated into the application IDE (integrated development environment).

If a business environment has a particularly critical application or if they are initiating new secure application development methodologies (including outsourcing), they should consider involving a trustworthy external consulting resource for an application risk assessment. The benefit gained by this kind of experience can extend far beyond a single application for those involved. A tremendous amount of security insight can be learned which can then be reinvested in future projects.

How secure is the application if there is only a single layer of protection (e.g. anti-virus)? Has security been integrated into the host O.S. platform? Has security been applied to the application? These are security functions for both applications development staff and I.T. operations. If no one can answer these questions, security wasn't properly addressed.

What are the historical security patterns for this application [or server(s)]?

In order to improve the quality of an application over time, there must be a closed-loop system where the effectiveness is evaluated and improvements are integrated into the next version. For example, when an existing application is reviewed for possible enhancement, it could be reviewed based on a given set of criteria, (e.g., usability or security) to see how it could be improved. To test the effectiveness of a given component, it is necessary to build in controls or audit points. To improve usability or client satisfaction, it would be important to identify or build in controls that give some

quantifiable data for this, such as might be available from a central help desk. Consider how data could be obtained from a help desk ticketing system. An application's ineffective attributes could be recorded and measured (e.g., x% of help desk calls for a given application are about a particular function that either fails or the user doesn't understand the process). Likewise for security, it is important to set up or identify appropriate mechanisms that would capture and archive security events and attack patterns. In the same analogy, security information could be derived from the same help desk ticketing system regarding password resets for a given application or other security relevant functions. From a security operations perspective, it is important to be able to answer the question, "Can historical information be queried from the data that is available regarding the patterns of attacks or elements that constitute risk against an application (or the hosts that serve the application?".

Elements which constitute risk (a.k.a. attack patterns) can be gathered from help desk tickets, O.S. event logs (login failures), applications event logs, IDS/security monitoring logs, and possibly firewall logs (if the application is a web application in the DMZ). Occasionally, logs are used as a programming debugging tool. However, logging should not be limited strictly to this function. Effective logging should record authentication failures, instances where a process is unsuccessfully initiated, and general exception events. Logging functions for measuring risk events must be included as a priority when writing the functional specifications for the application.

When a security event occurs, involving a given application, it is useful to know what good patterns in the log files look like and what bad log file patterns might look like. Has the developer provided any tools or resources that would help in the analysis of the audit trail? In addition to the application, have resources been identified within the organization that would help analyze the O.S. event logs? Every organization can benefit from having identified resources that are effective at analyzing O.S. logs, with the objective of finding the relevant one to three lines of event data among the millions of lines in the audit logs. Incident Response teams can benefit from knowing where relevant event logs are stored for each application, knowing about useful log analysis tools, and who is most adept at applying these tools to the data. This can also lead to alerting mechanisms, such as host IDS, which can monitor for known patterns in the log files and send alerts to appropriate parties.

Applying this to an operational security example, consider a web server operating in a DMZ which has just experienced a potential security event, also known as an anomalous network event. Would it be useful to know how many events (both host IDS and/or anti-virus) have been occurring against this host in the recent past (in the past 30-60 days)? Part of incident response is the forensics effort of reviewing events occurring against the application's host systems; forensics being the practice of examining evidence and applying that evidence to the resolution of issues. Forensics includes separating events, beginning with the first moment of suspicious activity, from all previous activity.

Here are a series questions that the security practitioner and /or incident responders must ask at the outset of an event to help determine if this is a security incident:

- Was the event an attack or merely a technical coincidence?
- When was it last attacked?
- When was the first attack of this particular event?
- What distinguishes this particular event from other recent events?
- How did the attacks occur?
- Was it a deliberate attack by a determined party or was it attacked as a part of a worm-like pattern? What is the best way to determine this answer?
- Were the attacks that occurred part of known patterns or were they complete anomalies?
- Was the attack from outside the enterprise or from inside?
- Is it possible to determine where the attacks came from?

The purpose of presenting these questions is to indicate that there is a need for resources to be able to provide the answers. The proactive effort should include the appropriate recording and logging mechanisms and building in appropriate analysis tools early in the development or deployment cycle of an application. It should be noted that the answers to these questions may not immediately seem equally relevant or interesting to different parties. An application developer may not care about whether the attack was outside or inside, but might be more interested in how the attack occurred (e.g. a buffer overflow). The business data owner may not care about the method of attack (buffer overflow of the xyz .dll component), but may have strong interest in whether the attacker was an insider or an outside competitor. If audit controls and security measures are not included in the specifications of an application, the result could be an application that works but which leaves an insufficient audit trail to identify mis-use, which would ultimately make the business data owner very unhappy.

If the specifications for the application at the beginning of the development were to include business performance metrics like support, security, and operation accountability, many functions and measures could be built in. For example, if the application's effectiveness were measured against how many support calls it generated, then its ease of use would be measurably improved. If the application were measured by frequency of application components failure or host system outages, then these measures would drive accountability in these areas and, hopefully, an improvement in the application. This extends also to security. If security "events" were measured by an application, the application could eventually become more secure. Effectively, what gets measured, gets noticed.

The ability to effectively enter into an event with potential security impact is reasonably strengthened when the capacity is there to identify previous patterns pertaining to the event. Part of the Preparations for Incident Response of a Security event is to have appropriate security monitoring in place.

What security mechanisms are in place outside of affected [application / data / host] system(s)?

In discussions thus far regarding applications, attention has been paid to the hosts that provide the processing and services used by the applications. In most current business environments, there are other shared services that contribute to the overall security of the environment. One of the most common security components operating in businesses is a firewall. Problems can arise when decision makers place all of their trust in this single layer of security. Firewalls typically allow common services such as web traffic, (SMTP) mail, FTP, and DNS to traverse in and out of the network, while blocking all other traffic. It should be no surprise that most of the attacks against businesses are now coming in via web traffic and mail, through the ports that the firewall does allow. This same firewall that allows the traffic that we use and trust to traverse our business boundaries, typically doesn't analyze it for malicious threats.

This specific lack of sufficient protection has given rise to new technologies which inspect web and mail traffic, also known as deep packet inspection or packet analysis. This new technology is the root of intrusion detection. When packet analysis occurs on the network, it provides a service to all hosts, sharing the cost and the protection among all network participants, except where network transport encryption is used. If transport encryption (e.g., IPSec or SSL) is used, then the packet analysis function on the network is ineffective because the analysis can't see into the contents of the packet. The alternative strategy to remediate this is to use IDS technology on the hosts. Another important function of packet analysis technology is the capability to filter packets that have been identified as having threatening content. This is now available in newer technology known as IPS (or Intrusion Protection System). Network IDS/IPS has many of the same traits as vulnerability management, including a dependency on an effective process, utilizing skilled staff, implementing effective policies, with current data.

A developing technology which is related to packet analysis technology or deep packet inspection is known as application firewalling. In the area of web traffic, there are now web application firewalls which intercept all HTTP traffic on port 80 and analyze this traffic for malicious content. Traffic that meets specific criteria can be blocked or filtered. In some cases, these devices are being deployed as a short cut to increase security in applications. Once again, problems can arise if developers do not implement secure development practices because they place all of their trust in this single layer of protection. Ultimately, security is improved when secure development practices are implemented and additional barriers are employed which prevent malicious traffic from affecting the environment.

There are additional mechanisms that can affect the risk of the systems that are currently operating. One example would be a load balancing architecture that is designed to provide high availability of applications to the client. Solutions like F5 Load Balancers send network traffic to identical servers so that in the event one server fails, the other continues to provide service. Load balancing or content switching should be properly documented so that incident handlers or security practitioners are aware of

their presence in the event of a security incident. Likewise, it is also important to know of the existence of virtual server mechanisms (VMWare, Virtual PC, etc.) and what role they play in the overall architecture, as well as where storage solutions (NAS or SANs) may fit into the architecture.

In preparing for incident response handling of an application, the security practitioner will need to understand the ip filtering and the security monitoring (IDS/IPS) mechanisms in order to identify threats occurring within the different systems that host the applications. The practitioner will need to know the full TCPIP traffic flow patterns. This information can be provided by answering the following questions:

What firewall is in place and what technology does it use?
What is the effective policy of the firewall?
What are the significant exceptions in the firewall rules set?
Is network IDS or host IDS in place?
What policy is running for host IDS and/or network IDS?
Is encryption in place (that would impact the effectiveness of NIDS)?
What is the response policy for the (IDS) security monitoring system?
What are the update regimens for security monitoring components?
What other kinds of packet filtering or IP management mechanisms may be in place (HTTP command, URL inspection, Network AV, caching, load balancing, etc.)?
Which system addresses are physical and which are virtual?

The security practitioner should take a pro-active role in preparing for incident responses by becoming familiar with every attribute of the architecture in which a client's application occurs. Where inadequate security exists, it is their duty and responsibility to identify these shortcomings and propose enhancements that will improve security.

What is this application/ server supposed to look like? What does it look like now?

In order for an incident responder to determine if something has been added, changed, or deleted on a host or an application, they must know what that host or application originally looked like. In information security terms, there is a function called a baseline, which takes a snapshot of the current environment. This baseline function should occur at deployment and should be updated after every change event.

To provide an orientation of the host computing environment, particularly in medium to large environments, maps are needed[vii]. Physical mapping can be very important, such as the physical location of a host (X rack, in Y row in a data center, or a geographic map displaying different cities in different time zones, or even in different continents). Mapping can also refer to how the process data flows between different application systems. It could also refer to different ip protocol ports and/or how the data flows from the client to the middle application processing tier to the back end database. Mapping can also be used to show how and where a given system fits in with other systems within the business environment. For instance, a map could show where the real time

transaction data is processed and how or where the overnight processing occurs. Another example of this process may include how a given data set enters a business environment from an external source and what processing happens to it before it leaves the environment.

To facilitate the proactive nature in the preparation stage of incident response, it would be beneficial to have standard mapping points in the development or deployment process and a standard mapping tool or exchange medium that allows exchanges of maps. The benefit of a standard mapping tool (e.g. MS Visio) or exchange medium (.vsd - Visio, or .PDF – Acrobat , or .dwg - Autocad) would prevent incident responders from having to locate the appropriate tools necessary to read a map during a security event. In fact, maps can benefit IT business continuity concerns just as much as they can benefit normal operations.

I recommend that mapping functions be a required element in the Change Management process. Maps are as much a normal part of IT operations as are developer comments in program source code. It would be constructive for physical rack and data center maps to be created or updated when new physical devices are introduced into the data center. In pre-production testing phases, application process data flow maps could be produced as a check list to validate full impact to the network. For applications that cross network boundaries (e.g. web applications that straddle a DMZ or across a GWAN), maps should be made available as a standard part of pre-production validation. Like much of security, this kind of detail should be done in the development/deployment phase of an application as a normal component of the Change Management process. It is painful for all parties to backtrack and document after the application is in production and the knowledgeable parties have moved on to other projects.

A very important element of mapping in current Information Security is identifying the TCPip profile (i.e., what ports are open?). It is crucial that the incident responders know TCPip ports that are supposed to be open. For example, it should be well documented which ip ports are supposed to be open in an MS SQL server constructed according to the organization's standards and should include any standard management tools (e.g., AV, IDS, backup, remote access solutions such as SSH, RDP, Xwindows)[viii]. Each application should have appropriate documentation about the ip ports that are used for clients, middle tier application processing, and back-end database processing systems. There are many tools that can be used for this such as Nmap,[ix] p0f, or most commercial vulnerability scanners. When the incident responder knows which ports are supposed to be open, any variances send up a red flag indicating possible compromise or infection. Knowing which ports are used by valid business applications that run on a given network is a foundational element of the infrastructure. For applications that cross network boundaries, good documentation about which ip ports are used is necessary to setup the application for correct functioning across firewalls and routers. If security is the only party requesting this information, it is a burden to those who have to provide it. If having an ip profile for all applications and hosts is a normal part of IT processes, the burden is absorbed unilaterally and everyone benefits.

In addition to mapping the applications and hosts, it is important to know what the operating parameters are for the servers hosting a given application. Examples of these parameters would include which O.S. and service pack release were included originally in the host and what is currently running? Tools such as Ecora Auditor[x] automate this tedious function. Security practitioners and incident responders also need to know what regimens that the Operations Teams apply to the hosts in order to keep them current. These regimens include the procedures for change management (e.g., how are security patches applied).

Another pro-active technique that could be applied in an incident response situation would be a series of reports that could be automatically initiated at the outset of an event. If a given public web server was suspected of compromise, or an internal server showed signs of a virus, it would be advantageous to immediately begin running management and historical reports for the affected host(s). These reports may include anti-virus activity, security monitoring (IDS) activity, patch management system reports, recent change management reports, system baseline reports, ip port maps, capture event logs, and application log files. Other useful information may also include purchasing information (including serial numbers and software keys) and maintenance or support information for all software and devices affected by the event. Optimally, it would be beneficial if the requests for these reports could be scripted and automatically generated at the outset of an event. These reports can be valuable elements in the investigation, and it promotes a faster time to resolution if they are immediately available. It may also be beneficial to burn these reports to read-only media like CD-rom in order to freeze the investigation data. These evidence files should be "signed" by a PKI certificate to ensure that the data is not manipulated.

System Rebuild

As the application team, consisting of developers and operations staff, considers the risk management and security of hosts that support an application, it would be prudent to consider SLA (service level agreement) or the tolerance for denial of service. When a network anomoly or security event occurs, a decision must be made at some point regarding when and if a system should be rebuilt in order to get the client up and running again.

At the beginning of an application's life or when the data's value is assessed, there could be or perhaps should be an assessment of the time tolerance or Service Level Agreement for delivery of service. While the actual parameter for number of hours or days may vary, it may be constructive to identify a formula that determines when to execute a system rebuild. Consider a structure where all public facing web servers are rated according to the table below:

SLA	Diagnostics Window	Outage Tolerance	Fail-Over
Critical	<2 Minute	<2 minutes	Cluster with redundant hardware

Urgent	<10 Minutes	<1 Hr.	1 hour rebuild from CD
Important	< 16 hours	<1 Day	Restore from tape – 8 Hrs.

If the issue is not diagnosed or resolved satisfactorily within the Diagnostics Window allotted time, a rebuild process would be initiated to recreate the host environment so the application could continue to run.

It is the absence of a plan that forces an IT unit to pursue a "patch and proceed" effort. When incident responders have no plan to refer to (policy, guidelines, standards, or procedures), the response may be to continue diagnostic efforts on a compromised host until it becomes functional, regardless of time, effort, resources, or impact. This leads to the question of how a tainted server can ever be trusted again after a "patch and proceed" effort. Can you ever be certain that the compromising experience didn't leave behind a back door program? Is there confidence that the tainted system will be stable? These questions may be addressed with appropriate security policies for compromised systems.

Conclusion

When the call comes at 0 dark 30 about a security event, it's too late to start thinking about all the mechanisms that should have been in place. The preparation for a potential security event should have been handled from the earliest possible consideration, even before the code for the application was written. Further, the security responder can't pull the support information out of thin air, but needs the support of the whole organization to assimilate the documentation that will enable effective response. The entire Information Services organization, comprised of business analysts/liaisons, application developers, and the operational staff, each have a role in defining and outlining how the application fits within the framework of the electronic enterprise. An organization that is already running at high efficiency will be integrating this support documentation into the deployment process. The information reviewed here does not benefit the incident responders alone, but benefits everyone in the organization by clarifying the baseline structures so that everyone can understand how the interrelationships of the hardware, software, and processes fit together to securely accomplish the tasks that make up the company's business.

---

[i] http://csrc.nist.gov/fasp/FASPDocs/incident-response/Incident-Response-Guide.pdf
FCC Computer Security Incident Response Guide, December, 2001

[ii] http://enterprisesecurity.symantec.com/article.cfm?articleid=3576&EID=0
Crafting an Incident Response Plan in Today's Threat Environment, APR 20, 2004 ARTICLE ID: 3576

[iii] http://www.guarded.net/cgi-bin/form_dev_new.cgi?02query1Referring+Page=GuardedNet+Home+Page&01query1Subject=Best+Practices+for+Incident+Response+-+The+Practitioner%92s+Guide+(CSO)&03query1Title=%22Best+Practices+for+Incident+Response+-+The+Practitioner%92s+Guide%22&04query1Send+To=leads@guarded.net
The Practitioner's Guide to Incident Response Best Practices **By Ken Pfeil, CSO, Capital IQ**

[iv] http://www.thisismoney.com/20040422/nm77286.html

[v] http://enterprisesecurity.symantec.com/article.cfm?articleid=3576&EID=0
Crafting an Incident Response Plan in Today's Threat Environment, APR 20, 2004 ARTICLE ID: 3576

[vi] http://csrc.nist.gov/fasp/FASPDocs/incident-response/Incident-Response-Guide.pdf
FCC Computer Security Incident Response Guide, December, 2001

[vii] http://www.networkcomputing.com/1123/1123f1side3.html?ls=NCJS_1123rt
Network Computing Security Feature: **Incident Response 101**

[viii] http://csrc.nist.gov/fasp/FASPDocs/incident-response/Incident-Response-Guide.pdf
FCC Computer Security Incident Response Guide, December, 2001

[ix] http://networking.earthweb.com/netsecur/article.php/1429131
Audit Your LAN Before the Bad Guys Do with nmap

[x] http://www.ecora.com/ecora/products/enterprise_auditor.asp

Last Updated: April 21st, 2018

Upcoming Training

SANS Seattle Spring 2018	Seattle, WA	Apr 23, 2018 - Apr 28, 2018	Live Event
Mentor Session - AW SEC401	Detroit, MI	May 01, 2018 - May 17, 2018	Mentor
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
SANS Northern VA Reston Spring 2018	Reston, VA	May 20, 2018 - May 25, 2018	Live Event
SANS Atlanta 2018	Atlanta, GA	May 29, 2018 - Jun 03, 2018	Live Event
Community SANS Bethesda SEC401	Bethesda, MD	Jun 04, 2018 - Jun 09, 2018	Community SANS
Community SANS New York SEC401	New York, NY	Jun 04, 2018 - Jun 09, 2018	Community SANS
SANS London June 2018	London, United Kingdom	Jun 04, 2018 - Jun 12, 2018	Live Event
SANS Rocky Mountain 2018	Denver, CO	Jun 04, 2018 - Jun 09, 2018	Live Event
Community SANS Madison SEC401	Madison, WI	Jun 18, 2018 - Jun 23, 2018	Community SANS
SANS Crystal City 2018	Arlington, VA	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS Cyber Defence Japan 2018	Tokyo, Japan	Jun 18, 2018 - Jun 30, 2018	Live Event
SANS Oslo June 2018	Oslo, Norway	Jun 18, 2018 - Jun 23, 2018	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 18, 2018 - Jun 23, 2018	Community SANS
Minneapolis 2018 - SEC401: Security Essentials Bootcamp Style	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	vLive
Community SANS Nashville SEC401	Nashville, TN	Jun 25, 2018 - Jun 30, 2018	Community SANS
SANS Minneapolis 2018	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS Cyber Defence Canberra 2018	Canberra, Australia	Jun 25, 2018 - Jul 07, 2018	Live Event
SANS Vancouver 2018	Vancouver, BC	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS London July 2018	London, United Kingdom	Jul 02, 2018 - Jul 07, 2018	Live Event
SANS Charlotte 2018	Charlotte, NC	Jul 09, 2018 - Jul 14, 2018	Live Event
SANS Cyber Defence Singapore 2018	Singapore, Singapore	Jul 09, 2018 - Jul 14, 2018	Live Event
SANSFIRE 2018	Washington, DC	Jul 14, 2018 - Jul 21, 2018	Live Event
SANS Malaysia 2018	Kuala Lumpur, Malaysia	Jul 16, 2018 - Jul 21, 2018	Live Event
SANSFIRE 2018 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 16, 2018 - Jul 21, 2018	vLive
Mentor Session - SEC401	Jacksonville, FL	Jul 17, 2018 - Aug 28, 2018	Mentor
Community SANS Bethesda SEC401	Bethesda, MD	Jul 23, 2018 - Jul 28, 2018	Community SANS
SANS Pittsburgh 2018	Pittsburgh, PA	Jul 30, 2018 - Aug 04, 2018	Live Event
San Antonio 2018 - SEC401: Security Essentials Bootcamp Style	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	vLive
SANS August Sydney 2018	Sydney, Australia	Aug 06, 2018 - Aug 25, 2018	Live Event
SANS Hyderabad 2018	Hyderabad, India	Aug 06, 2018 - Aug 11, 2018	Live Event