Global Information Assurance Certification Paper

Methods for the Controlled Deployment and Operation of a Virtual Patching Program

GIAC (GCIA) Gold Certification

Author: William C. Vink, William.Vink@Villanova.edu
Advisor: Tanya Baccam
Accepted: 4/20/2018

Abstract

In today's rapidly changing IT environments, new vulnerabilities are identified at an increasing pace and attackers are becoming more sophisticated in their ability to exploit these vulnerabilities. At the same time, systems have become more complex and are still used in conjunction with older technologies which results in challenges in testing and deploying traditional patches.

Virtual patching allows companies to provide a layer of defense to prevent exploitation of these vulnerabilities and give them additional time to assess the risks, test and deploy virtual patches or when necessary keep the virtual patch rule in place where patching is not achievable. The deployment and management of a Virtual Patching system must be done in a secure, controlled manner as all technology deployments should be.

This paper will discuss how virtual patching is securely deployed to block the exploitation of vulnerabilities using web application firewalls (WAF) and Intrusion Prevention Systems (IPS), and the prerequisites for a successful virtual patching program. The controls needed to implement the different virtual patching taxonomies will be discussed in the context of standard control frameworks such as COBIT and ITIL.

Table of Contents

Introduction

What is Virtual Patching?

There has been a great deal of material written on virtual patching that discusses how it is a powerful tool for protecting networks from external attacks. The term patch is misleading because the vulnerable system is not being patched. In actuality, it is the insertion of rules to restrict the inputs and outputs to the vulnerable application in an intermediary layer. Virtual patching is defined by the Open Web Application Security Project (OWASP) as "[a] security policy enforcement layer which prevents the exploitation of a known vulnerability." [1]

This method of protection involves the deployment of web application firewalls with rule sets designed to protect the network from the exploitation of specific vulnerabilities. Doing so buys additional time for the organization to evaluate the risks associated with the vulnerability in their environment and develop a mitigation strategy. In some cases, the organization will be able to address the vulnerability at the application level, which will allow virtual patching rules for that vulnerability to be removed. In other situations where actual patching is not feasible or cost-effective, the rules specific to that vulnerability may remain in effect.

To properly manage the testing, deployment, and retirement of these rules it is necessary to follow the same types of IT controls that are required for any other system in the company's environment. These controls include code management, SDLC controls, Technology Asset Management, Change Management, and Configuration Management.

Understanding all the software on the network including all the versions and patch levels will allow analysts to determine where to place the virtual patching devices. The organization must have an up to date network map that details the different input and output paths to the applications. Otherwise, a path could be left unprotected, or a device could be deployed in a network path that is not at risk. As shown in figure 1 below the virtual patching device must sit between the attacker and the system that has the vulnerability.

[1] https://www.owasp.org/index.php/Virtual_Patching_Best_Practices

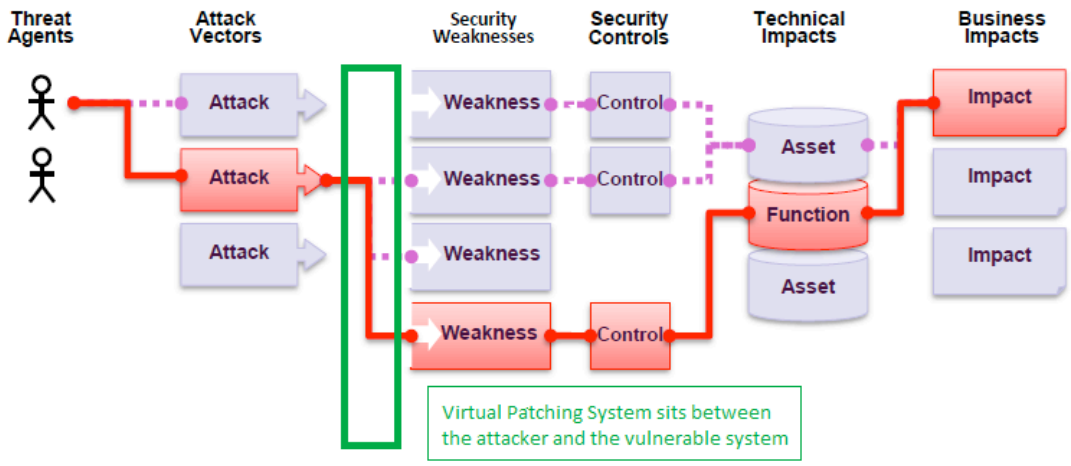

Figure 1: Virtual Patching Placement in the Network [2]

What are the Advantages of Virtual Patching?

The two primary goals of virtual patching are to quickly implement safety measures and to reduce the organization's exposure. Virtual patching allows the organization to more quickly protect its networks from vulnerabilities identified in applications that would require an extended period to remediate or that cannot be remediated in a cost-effective manner.

Today's systems are very complex, with multiple dependencies and interrelationships. It takes time to develop a fix and test it in operation- like conditions. Implementing a virtual patch does not alter the operation of the underlying application or the systems that interact with it. Testers and analysts are also able to run the patch in monitor mode to evaluate any potential impacts prior to turning on the blocking functions.

In some cases, the virtual patch may not be able to remediate the vulnerability fully but can reduce the ability of an attacker to exploit it by limiting inputs and outputs of interactions on the system. For instance, it may still be possible for the attacker to send the attack to the system but the WAF would block any outputs that would be returned to the attacker.

What are the Disadvantages of Virtual Patching?

While a Virtual Patch protects the underlying system from attacks, it is possible that the WAF is not deployed on all the entry points into the system. It is also possible that multiple web pages could be using the same vulnerable code.

[2] https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

Unless the devices are deployed to protect all these web pages, the system may still be vulnerable to the attack.

A danger in the use of virtual patching is that it could give management the impression that they no longer need to worry about the vulnerability which would reduce the appetite to spend money on fixing the code. Future development efforts, configuration changes or additional instances of the code could re-expose the vulnerability to attackers if not properly managed. Virtual patching is not a "set it and forget it" solution. “Any sort of change to the topology of the network, or even minor configuration changes, can render a virtual patch ineffective, exposing an application all over again. It's important to follow up with a plan to fix the root problem, including deploying the vendor's patch whenever possible and practical."[3]

With virtual patching, there are also risks associated with the configuration of the rules. A rule that is more restrictive, one that blocks more traffic, has a higher chance to block legitimate traffic. A rule that is more permissive allows more traffic through, but it increases the possibility of letting malicious traffic through. The organization must find the rule structure that best suits their particular network and risk appetite.

Why use Virtual Patching

As discussed previously, there are limitations and risks associated with the use of virtual patching. Despite these limitations and risks, virtual patching should still be used because it provides the quickest way to provide protection, even if it is temporary. When possible, the preferred solution would be to fix the vulnerable application. Once the code is fixed analysts do not need to worry about the virtual patch not being deployed on all the ingress points that can access the code. They also do not need to ensure that new deployments of the code are covered by the patch. In most organizations this is not always feasible for multiple reasons:

- Not all organizations have the number of resources needed to be able to pull them for unplanned work and still manage the keep the engine running activities.
- In some cases, the vulnerabilities are found in third-party software where the organization does not have access to the source code or have permission to alter it.
- Some organizations do not manage their own systems and are at the mercy of their service provider to manage development projects.

By using a virtual patching system, an organization buys itself the time needed to free up resources, work with its vendors, and develop a sound risk response without leaving systems vulnerable during this process.

[3] https://searchsecurity.techtarget.com/tip/Use-virtual-patching-to-ease-short-staffed-patch-management-procedures

As noted above the virtual patch should be viewed as a temporary fix wherever it is possible to remediate the code. In situations where remediation of the code cannot be achieved, the acceptance of the virtual patch as a long-term solution should be fully documented and added to any testing documentation for the application.

Virtual Patching Deployment Process

This section will discuss the implementation of virtual patching and the risks associated with not having the proper foundational controls in place to support the deployment and management of the program.

The different components of the virtual patching system through the phases of design, build, implementation and operation will be addressed. The management of the firewall rules that are used to mitigate vulnerabilities will also be discussed. Virtual patching needs to be approached in a controlled manner the same as any other IT Project. The virtual patching system is made up of hardware and code that is going to be deployed on the organization's network, and the same development, security, and change management controls used to develop and deploy other web applications should be followed.

The systems used to implement virtual patching are usually either a web application firewall (WAF), Intrusion Detection System (IDS), or an Intrusion Prevention System (IPS). These systems are perform scans of network traffic and evaluate that traffic against rule sets designed to detect attempts to exploit a particular weakness. The virtual patching system is reliant upon a secure process to develop, test, deploy and manage those rule sets. These processes and controls are the same as the controls that are used for any standard web application development process. Once the virtual patching system is in place, it needs to be monitored. There is the monitoring of the system itself for failures or unauthorized access and changes, as well as monitoring instances where the rules are triggered by network traffic. The process used to monitor the virtual patching system for failures would be the same as standard network device monitoring and the processes and controls for monitoring the triggering of rules would follow those used for security event and incident management.

As we discuss each step in the implementation of the virtual patching, we will identify the relevant control objectives. This demonstration will be using the Control Objectives for Business Information Technology (COBIT) 5 framework to identify the necessary controls for the management of the virtual patching system. COBIT is a widely accepted control framework that is often used by IT auditors. COBIT is managed by the Information System Audit and Control Association (ISACA). The framework has five main components that correlate to the governance, development, implementation operations and monitoring of IT systems. This paper will discuss using the Control Objectives for Business Information Technology (COBIT) framework to ensure that security processes and controls meet industry standard and are followed in a consistent, repeatable

manner. The Build Acquire and Implement Controls from COBIT 5 provide guidance on what controls to include in that repeatable process. Control Objective *"BAI01 Manage Programs and Projects"* states that organizations should "manage all programs and projects from the investment portfolio in alignment with enterprise strategy and in a coordinated way. Initiate, plan, control, and execute programs and projects, and close with a post-implementation review."[4] Failure to follow the company's standard development methodology and deployment controls could result in missed requirements, inadequate testing, or insecure code. The virtual patching tool is an appliance on the network and has access to traffic to and from servers. The critical security controls to secure configurations for other network devices apply to the web application firewall. The secure configuration of the devices must be established and is subject to monitoring and a secure change control process which prevents attackers from exploiting vulnerable settings and services on these devices. As with all devices, any services that are not needed should be disabled.

Components involved in the virtual patching deployment process include:

- Network Hardware: Web Application Firewalls, Network Taps, etc.
- Software: SIEM, Virtual Patching Rules,
- Processes: Hardware Asset Management, Code Management, Code Deployment, Change Management, Configuration Management, and Data Management.

The areas that are essential to a successful deployment of a virtual patch management system begin with Build Acquire and Implement. The controls here cover knowing what is on the network, what versions of software are running, how the systems are configured, and the processes that ensure this data is up to date and accurate. The controls that cover secure development and testing are also included in this section and are necessary to ensure that virtual patch rules are developed properly and included in the testing of the application changes.

The controls in "Deliver and Support" cover the deployment of the virtual patching system and its rule sets in a controlled manner and subject to proper management.

Finally, the "Monitor, Evaluate, and Assess" domain covers the review of the outputs of the virtual patching system, including the effectiveness of the rules and the follow up on identified threats.

The chart below illustrates how the COBIT controls align to the different phases of the development, deployment, operations and oversight phases of the system lifecycle. Key controls can be found where the system transitions from

[4] COBIT 5 http://www.isaca.org/cobit/pages/default.aspx

one phase to another, ensuring proper documentation and defined roles and responsibilities for the controls in each.

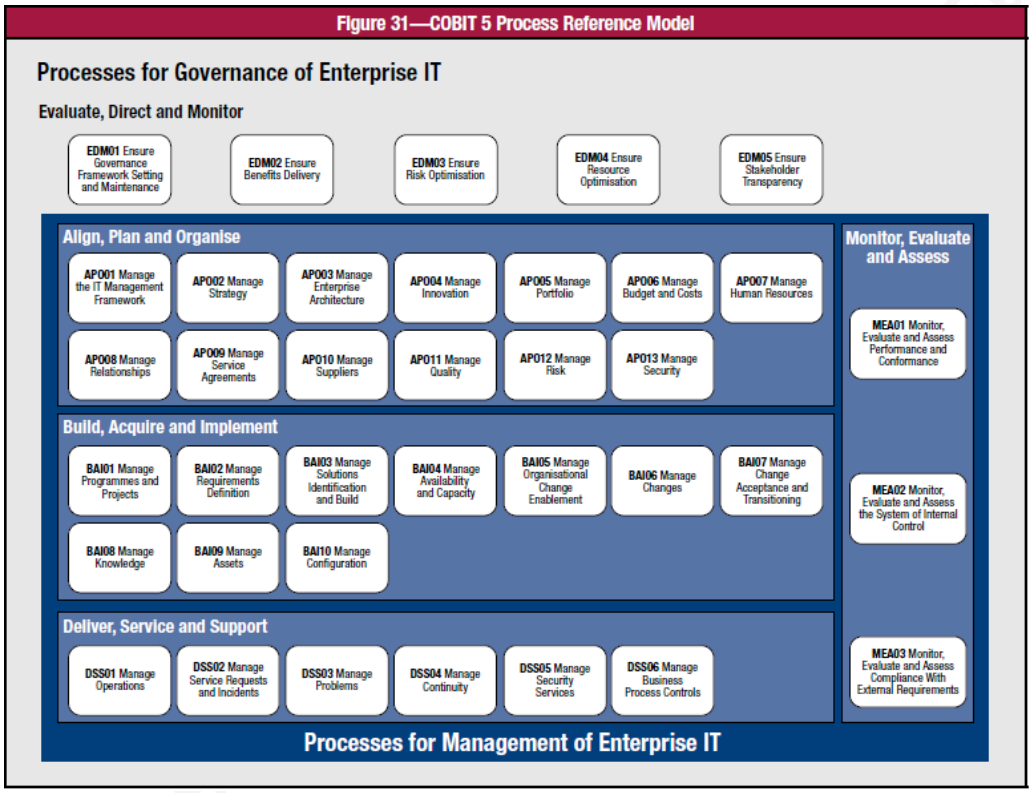

Figure 2: COBIT 5 Process Reference Model[5]

The first step is to identify the requirements of the virtual patching system. Virtual patching is very good at handling injection-type vulnerabilities that are a key to several of the OWASP Top 10 risks including; un-validated input, cross-site scripting (XSS) flaws, buffer overflows, improper error handling, and injection flaws. As we look at each of these vulnerabilities, they all are related to data protection. In order to design a system to provide data protection, we must understand what our data is, how it is used, and where it is stored. This step requires the analyst to evaluate whether they have enough information gathered to make an informed decision and whether the data is accurate and up to date.

We will start by reviewing the system's Data Classification, Data Management, and Technology Asset Management. These controls are covered under the Align Plan and Organize control objectives. *APO01.06, "Define information (data) and system ownership"* requires that the organization classifies information and systems and provides the appropriate level of protection based on this classification. Classifications schemes often use terms like Confidential, Highly Confidential, Internal Use only and Publicly Available. Systems that contain Highly Confidential of Confidential Data would be prioritized for a virtual patching deployment. Doing so will require an up to date inventory of these

[5] COBIT 5 - A Business Framework for the Governance and Management of Enterprise IT - ISACA

systems and systems documentation on the processes and interfaces that can access them. *APO03.02 – "Define reference architecture"* provides controls objectives associated with defining the reference architecture that describes the current and future state architecture data and applications. This documentation will help analysts decide where the virtual patching system should be deployed.

If an organization's data is not properly inventoried and classified, the virtual patching system may be deployed in a place or manner that does not protect all the pathways that an attacker can use to infiltrate the network. For example, the organization could identify that their client records are stored on one server and deploy the virtual patching appliance in between that server and the perimeter. If the organization does not maintain an up to data network diagram with data flows their analysts could miss that there are alternate paths to that server. An analyst could also miss that there is the ability to change data while it is being processed on another network segment and returned. The virtual patching system will protect that data while it is on the server identified in the system the documentation, but the result is that the system can still be compromised.

Once the organization has established what data and systems most need to be protected, the process of documenting requirements can begin. *BAI02 Manage Requirements Definition* calls for us to identify solutions and analyze requirements before acquisition or creation to ensure that they are in line with strategic enterprise requirements covering business processes, applications, information/data, infrastructure, and services. The security organization must coordinate with affected stakeholders to the review of feasible options including relative costs and benefits, risk analysis, and approval of requirements and proposed solutions.

While the analyst can identify what confidential data is on the network, it is the business that should be deciding what data is critical and prioritizing what they want to protect. From a security analyst standpoint/viewpoint, the configurations, credential store and encryption keys must be protected. However, the business side of management is responsible for classifying business data such as customer records, transactional data, operational plans, and any other data they would not want to be made public. The business must also be involved in the discussions of the level of acceptable false positives. They must be made aware of how much valid traffic may be disrupted and decide on what their risk appetite is.

To correctly assess the risk the software vulnerability has to the organization the analyst needs to know what data is at risk, where it is stored and processed, and how it is accessed. This information will be critical in knowing where to deploy the virtual patching device on the network. The business' analysts will need inputs from the network and system documentation to assist in gathering the requirements. The quality of that documentation will be impacted by the controls in place to ensure they are complete and up to date.

If the organization has implemented the proper controls to meet *APO03 Manage Enterprise Architecture*, analysts can answer the following questions:

- Where is my data stored? The answer to this will come from system documentation and database architecture.
- What is my critical data and how is that data used? The answer will rely on the data classification, metadata maintenance, data mapping, and process inventory.

Organizations must have a process in place to keep this information up to date. Not only is this information needed to know where to deploy the virtual patching system and rules, but also it is required if there is a breach to identify the data that was exposed. (Failure to have this data available could result in regulatory fines.)

The next step is to look at what hardware is used for the systems identified in the previous step. The analyst will want to know the network configurations and what operating systems are being run on it. Control objective *BAI09.01 "Manage Assets - Identify and record current assets"* calls for the organization to maintain an up-to-date and accurate record of all IT assets required to deliver services and ensure alignment with configuration management and financial management. The system should identify what versions of software are installed to determine if they are affected; otherwise, a patch may be installed that is not needed or one that is not the correct patch for the configuration being used. Some vulnerabilities are only able to be exploited with specific configuration settings, so it may also be possible to avoid the vulnerability with a simple configuration change. If the organization's documentation is not up to date, they may deploy rules that do not block all the correct inputs to prevent exploitation of the vulnerability.

There is a dependency on change management and configuration management if the analyst is to be able to rely on the data. In *BAI10 Manage Configuration*, the organization will define and maintain descriptions and relationships between key resources and capabilities required to deliver IT-enabled services, including collecting configuration information, establishing baselines, verifying and auditing configuration information, and updating the configuration repository.

If the organization does not have the most up to date application and hardware inventory they will not be able to accurately identify whether they have the version of the software that has the vulnerability or which rules would protect the inputs that would be used to exploit them. A patch that protects version 1.0 of an application may be different than the patch needed to protect a version 2.0 if the input fields and parameters have changed.

Preparation

Following COBIT Control *BAI06 – the "Manage Changes"* option will ensure that all changes are managed in a controlled manner, including standard changes and emergency maintenance relating to business processes, applications, and infrastructure. These controls include change standards and procedures, impact assessment, prioritization and authorization, emergency changes, tracking, reporting, closure, and documentation.

Select the Hardware / Tools That Will Be Used to Facilitate Virtual Patching

BAI03: Manage Solutions Identification and Build calls for the organization to establish and maintain identified solutions in line with enterprise requirements covering design, development, procurement/sourcing and partnering with suppliers/vendors. The technical operations group will use these controls to manage configurations, develop and execute appropriate testing for all requirements. The management and maintenance of business processes, applications, information/data, infrastructure, and services should be reviewed and updated to support the deployment of the virtual patching.

To select the right hardware and software to implement the virtual patching system, the analyst will need to have a strong understanding of the network they are deploying it on. They must know what platforms it will be running on, the software and databases it will be protecting and the type of traffic that will be covered by the patching system.

Vulnerabilities are made up of flaws in specific applications, systems, hardware, and devices. In most cases, they are also identified as existing in specific versions or patch levels of those system components. To properly identify what vulnerabilities are on the organization's network, the inventories of the authorized and unauthorized software and devices on the network must be kept up to date as prescribed in the Center for Information Security's (CIS) top 20 controls. If an organization does not know what is on its network, not only will they not be able to identify all the vulnerabilities that exist on their network, they would also not be able to select the correct remediation for those that are identified. If the vulnerability exists in different versions of an application that has variations in the input methods the virtual patching rule set that protects, one version may be ineffective at protecting the other. If there are two versions of an application running and rules are deployed that cover both, the network can still be protected. However, this is dependent on the analysts knowing if and what versions are running.

Architectural Options

There are multiple options for deploying a virtual patching system, each with its advantages and disadvantages. The analyst must decide if the patching system will sit on dedicated hardware or if it will use services on existing servers.

Using existing servers saves costs, but any process that evaluates and filters traffic can impact performance.

Another decision is whether to have the virtual patching in-line on the network or have it as a tap. Using a tap can reduce performance impact but limits the ability to filter and stop traffic. Once the decision as to what type of hardware and deployment will be used, the selection of the actual tools and deployment requirements can begin. There are several different tools that may be used to facilitate virtual patching. Besides using a WAF or IPS, there is the option to use a web server plugin like ModSecurity, or Application layer filters such as ESAPI WAF.

Once the decision on where to deploy the virtual patching device has been made, there are multiple options as to what tool to use. Tech Target published an article that compared some of the best Web-application-firewalls. "Appliance-based WAFs include; F5's BIG-IP ASM, Imperva's SecureSphere, Barracuda's WAF, the Citrix NetScaler MPX WAF and Dell's Sonicwall WAF. Qualys' WAF and Imperva's Incapsula are WAF products covered in the cloud/hybrid category, while the ModSecurity Web Application Firewall is the lone entry addressed in the rather unique code integrated product category."[6]

Cloud deployments should also be considered as an architectural option. Do not assume that your cloud provider is managing all the security for your applications running on a public cloud. They provide the security of the container but, the organization is responsible for protecting what you have running in cloud deployments. Services such as Trend Micro's Hybrid Cloud Security provide Virtual Patching Solutions for Cloud.

One input that will help determine what tool to use is the method of deployment. If the network is running on Apache Servers, it is a simple solution to enable MOD Security and run virtual patches with it for the applications running on that server. The advantage is that no additional hardware is needed, and it can be deployed quickly. The drawback is that it only protects the apps on that server and would need to be applied to each server.

Another option is to use a reverse proxy method utilizing MOD security. Using the reverse proxy method allows the patch to be implemented once on the proxy, protecting multiple instances of the applications. The downside here is that it creates a potential bottleneck and single point of failure. This option also runs the risk of not protecting a server that has an alternate path to the internet.

Today the Web Application Firewall (WAF) has become a large part of the security posture in many organizations. Once a WAF is in place it is easy to leverage it to implement virtual patching rules. Like Mod Security there are different deployment options. WAFs can be set up as a reverse proxy or

[6] http://searchsecurity.techtarget.com/feature/Comparing-the-best-Web-application-firewalls-in-the-industry

embedded similar to MOD Security. A reverse proxy WAF has an IP address and takes the external connections and relays them to the destination server, applying the rules before passing it on. An embedded or server resident WAF is the same concept as running MOD Security on the Apache server.

Some WAFs are run out of band using a network tap. This configuration limits the WAF's functionality. Since the WAF is reading the data at the same time, it is being routed it is unable to block it and is only able to alert on malicious traffic. This deployment is acceptable for testing a virtual patch in monitor mode but, because it is not in-line, its limited ability to block malicious traffic makes it a poor selection for virtual patching.

Another method of deploying a WAF is as a layer two bridge. The deployment is similar to a reverse proxy but, while the WAF is able to pass traffic quickly and drop packets that meet the rule parameters, it cannot decrypt traffic.

Criteria that must be reviewed in selecting a virtual patching architecture include the following:

- Is the data to be protected stored in a central location or is it dispersed to multiple network segments?
- What is the network configuration and how does the data flow through the network?
- What type of hardware and applications are running on the network? Are there systems from multiple vendors?
- Is the network running Windows, Unix/Linux, another operating system, or some combination?
- What is the business' risk appetite? Is it more towards no false negatives vs. no false positives?
- Is the network managed internally or outsourced?
- Will the virtual patching system be run by employees or a managed security service provider? (Does the organization have the knowledge in-house?)

Threat modeling and use cases can help narrow down the choices. For the purposes of this paper, Web Application Firewall or Intrusion Prevention Systems will be the focus.

Develop and Deploy

The device selected will be on the production network and must adhere to all security practices that the other hardware follows. Using the COBIT 5 standard "*BAI09.01 Manage Assets - Identify and record current assets*" includes "maintaining alignment with the change management and configuration management processes, the configuration management system, and the

financial accounting records."[7] Any changes to the applications and network can impact the ability of the virtual patching system to block malicious traffic. By using the COBIT controls the organization will have documented processes that force the review of new assets or changes to existing ones. This documentation should be updated to reference any virtual patches that are protecting the asset so that the patch can be included in the requirements and testing.

Systems must be tested before being deployed to production and must be provisioned with the appropriate access. Proper segregation of the roles for developing, testing, and elevating to production should be exercised to prevent the elevation of unauthorized or untested code. These controls are in place to prevent a developer, even a well-intentioned one, from making changes in production which could result in system errors, outages or disabling of the security device.

Virtual Patches – Firewall Rules

When and Where to Apply a Virtual Patch

To deploy the virtual patch, it is necessary to

- Identify that the vulnerability exists,
- Identify whether specific systems/ software are vulnerable,
- Know how the vulnerability is exploitable,
- Know what firewall rules would prevent an attacker from being able to execute the exploit.
- Determine whether the capabilities of the Web Application Firewall can detect an attempted exploit and apply the rule.

There are two different approaches for identifying if there is a potential vulnerability in the network. These methods should be considered as complementary rather than an either-or decision.

- One method is to perform Source code reviews or Dynamic Application Assessments to identify flaws. These internal assessments include penetration testing or vulnerability scanning with tools such as Qualys. This method is more effective in situations where there are multiple applications interacting, or if there is customized code.
- The second method is subscribing to outside services either from the software vendor or a security services provider. Alerts should be enabled with any of the vendors. It is also a best practice to include language in contracts that address their responsibilities in identifying vulnerabilities and supporting remediation efforts. Many vulnerability announcements include sample exploit code that shows how to demonstrate the vulnerability. These samples can be used for the development and testing of the virtual patch.

[7] http://www.isaca.org/COBIT/Pages/COBIT-5-Enabling-Processes-product-page.aspx

Determining if Virtual Patching is Appropriate

As noted above, virtual patching works well with injection vulnerabilities. There are other types of vulnerabilities that virtual patching does not solve for as well or at all. The functionality of the WAF being used should be considered to determine if it has the capability to detect and block the exploitation of the identified vulnerabilities. Each vulnerability needs to be reviewed to determine the appropriate remediation approach and how to achieve attack surface reduction for that attack type and category.

Virtual Patch Creation – Once it is determined that a virtual patch rule is appropriate, it must then be determined how to create the rules.

When a vulnerability is identified, it should logged into the bug tracking system for tracking purposes as any other network issue would be. Some commercial options include Jira or Peregrine. Use the public CVE name/number assigned to the vulnerability where applicable. If the organization has identified the vulnerability via their internal reviews, then it should assign each a unique identifier to each vulnerability. The organization may choose to report the vulnerability to their vendor or MSSP, and it may later get a CVE that can be assigned. Bug tracking systems have a risk or criticality level that should be used to prioritize the remediation of the vulnerability and reporting.

Threat Agents	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impacts	Business Impacts
Appli-cation Specific	Easy: 3	Widespread: 3	Easy: 3	Severe: 3	Business Specific
	Average: 2	Common: 2	Average: 2	Moderate: 2	
	Difficult: 1	Uncommon: 1	Difficult: 1	Minor: 1	

Each organization is unique, and so are the threat actors for that organization, their goals, and the impact of any breach. If a public interest organization uses a content management system (CMS) for public information and a health system uses that same exact CMS for sensitive health records, the threat actors and business impacts can be very different for the same software. It is critical to understand the risk to your organization based on applicable threat agents and business impacts.

Figure 3: OWASP Risk Rating Model 8

With the growing number of vulnerabilities and the complexity of the patches needed, most organizations do not have the dedicated resources to create all their virtual patches manually. Therefore, they must either rely on a service such as Trustwave and Alert Logic which provide Managed Rules for AWS WAF services for Amazon Cloud. If the organization's automated scanning

[8] https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

tools can produce an XML report the OWASP Core Rule Set (CRS) includes scripts to auto-convert XML output from tools such as OWASP ZAP into ModSecurity Virtual Patches.[9]

Many of today's web application firewalls have the capability to import XML report data and automatically adjust their protection profiles. For homegrown code or unique network configurations, it may be necessary for the analyst to modify rules or create their own.

Acquire and Implement (SDLC, Change Management)

The implementation of the hardware system should follow standard code management controls under using COBIT 5 Control Objective *BAI03.03 Develop solution components*. Analysts should "develop solution components progressively in accordance with detailed designs following development methods and documentation standards, quality assurance (QA) requirements, and approval standards. Ensure that all control requirements in the business processes, supporting IT applications and infrastructure services, services and technology products, and partners/suppliers are addressed."

Virtual Patches need to be implemented quickly so an expedited emergency change process should be followed. Virtual patches are not modifying source code and, if run in monitor mode prior to going live, they do not pose the same risk level as other code elevations. These patches should be managed similarly to other Network IDS/IPS signatures.

It was mentioned above that virtual patches do not have the same level of risk as other changes that involve code modification. This does not mean that they get to bypass testing. Virtual patches should initially be loaded set to log not block. Loading in monitor mode provides the opportunity to evaluate what traffic would have been blocked and identify any potential false positives that could impact the business. The analyst could then run packet captures of potentially malicious traffic that was identified through the virtual patch to determine if the exploit would be blocked. By doing this, it allows the opportunity to test for false negatives without disrupting network traffic.

It may not be possible to implement a patch that achieves zero false positives and zero false negatives. If it is likely that a virtual patch rule will disrupt legitimate network traffic here should be a discussion with management about the potential risk of each scenario, and the duration during which the patch is expected to be in place, and an agreement on the course of action. It is possible to reduce the volume of decisions the virtual patching logic must make by limiting the inputs or transaction on the web interface. There are two models for limiting website inputs, White Listing and Black Listing.

[9] http://blog.spiderlabs.com/2012/03/modsecurity-advanced-topic-of-the-week-automated-virtual-patching-using-owasp-zed-attack-proxy.html

White Listing, sometimes referred to as Positive Security takes the approach of excluding anything that is not explicitly allowed. This more restrictive approach provides greater security but requires the documentation of all the permitted inputs. Black Listing or Negative Security allows any inputs not explicitly flagged for rejection. Black Listing typically identifies inputs associated with known attacks and allows everything else. It normally flags all commands and characters associated with SQL commands or other items associated with injection attacks.

To accurately test out the newly created virtual patches, it may be necessary to use an application other than a web browser. Some useful tools are[10]:

- Web browser
- Command line web clients such as Curl and Wget.
- Local Proxy Servers such as OWASP ZAP (https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project).
- ModSecurity AuditViewer (http://www.jwall.org/web/audit/viewer.jsp) – which allows the analyst to load a ModSecurity audit log file, manipulate it and then re-inject the data back into any web server

In order to properly use the outputs of logging systems to analyze vulnerabilities and the effectiveness of the virtual patch, it may be necessary to modify the systems HTTP Audit Logging. Most web servers come configured with a standard logging format that does not provide all the data needed for full incident analysis. It is recommended that logging capture the full request header, request body and the URI query that called the page. The full response header and response body will also be needed. Without this information, it is not possible to perform proper forensics on an incident. As a wise instructor once said, "PCAP or it didn't happen!"

Deliver and Support

Once the patching system is up and running, it will be turned over to IT operations to manage. The Deliver and Support Domain covers the management of the production hardware and software and controls needed to provide for a secure system. The Deliver and Support Domain also covers the following relevant controls:

- *DSS01 Monitor the IT infrastructure*
- *DSS02 Manage Service Requests and Incidents*
- *DSS03 Manage Problems*
- *DSS04 Manage Continuity*
- *DSS05 Manage Security Services*

[10] https://www.owasp.org/index.php/Virtual_Patching_Cheat_Sheet

- *DSS06 Manage Business Process Controls*

COBIT 5 Deliver and Support Control Objective *DSS01 Manage Operations* calls for monitoring of the IT infrastructure and related events. This control objective requires the organization to store sufficient chronological information in operations logs to enable the reconstruction, review, and examination of the time sequences of operations and the other activities surrounding or supporting operations.

In order to meet this standard, the organization must meet the following criteria:[11]

1. Log events, identifying the level of information to be recorded based on a consideration of risk and performance.
2. Identify and maintain a list of infrastructure assets that need to be monitored based on service criticality and the relationship between configuration items and services that depend on them.
3. Define and implement rules that identify and record threshold breaches and event conditions. Find a balance between generating inconsequential minor events and significant events so that event logs are not overloaded with unnecessary information.
4. Produce event logs and retain them for an appropriate period to assist in future investigations.
5. Establish procedures for monitoring event logs and conduct regular reviews.
6. Ensure that incident tickets are created in a timely manner when monitoring identifies deviations from defined thresholds.

Monitor and Evaluate

A virtual patching system has two components of monitoring. There is the monitoring of the hardware, software, and rules that make up the system itself. There is also the monitoring of the outputs of the implemented rules. In order to meet the conditions of COBIT's control objectives, we must develop a program for System Monitoring, Problem Management, and Event and Incident Management as it pertains to virtual patching and the measurement of its effectiveness. This program should be integrated into the reporting and oversight of the problem ticket system. Tickets opened as part of the patch management processes should be monitored to make sure they are being updated and tracked to establish the amount of time and effort needed to

[11] COBIT 5: Enabling Processes - http://www.isaca.org/COBIT/Pages/COBIT-5-Enabling-Processes-product-page.aspx.

remediate code or to age the patch for periodic re-assessments. Over time vulnerabilities may become no longer applicable due to changes in network architecture, the retirement of the impacted code, or updates to third-party products.

It is also best practice to monitor virtual patch alerts to track any activity that triggers a virtual patch. This data can be used for threat evaluation and to evaluating the program's operating effectiveness. Monitoring when the rules are triggered can also help with fine tuning the rules to reduce false positives.

A decrease in rule alerts does not necessarily mean the risk is over. It could mean that the attacks used to exploit the vulnerability have evolved to evade the current rule set. Further analysis is needed to determine if the rule should be modified or retired. The results of this analysis should be integrated into a continuous process of threat evaluation and be considered when evaluating the program's operating effectiveness.

Additional Uses

The efforts to deploy a virtual patching system are considerable but, there are many benefits, such as:

- The rules used to implement the virtual patch can also be used to identify previous intrusions: By taking the activity logs from the network and transforming them into PCAP files it is possible to run them through an analysis tool such as SNORT, BRO, Wireshark/TSHARK, SILK or TCPDump. Doing this will enable the analyst to run the traffic through the tool and see if it triggers the rules, indicating that there was an attempt to exploit the vulnerability prior to implementing the patch. This analysis can then be used for forensic purposes to determine if the system has been compromised. It can also provide assistance in a follow-up investigation.
- The same virtual patching rules used in production can be implemented in monitor mode on a Honeypot to gather intelligence on how attackers are attempting to exploit the vulnerability and then be used to fine-tune the rules.
- Logs of traffic blocked by the virtual patching rules protecting vulnerability are a metric that can be used to show the value of the security program. The up-front work of performing a business impact analysis allows the analyst to use the business' evaluation of a potential breach to show the potential loss the system has prevented along with the negative publicity and reputational impact.

Conclusions and Recommendations for Further Analysis

Virtual patching is an effective tool in a defense-in-depth approach to vulnerability management. It provides security and buys additional time to

remediate code vulnerabilities. As with any hardware and applications running on a network, there is an element of risk that requires adequate governance and oversight to ensure that it is operated in a secure manner and provides its expected value.

COBIT 5's governance model provides a framework of proper controls for managing the virtual patching program. Adherence to its control objectives in the rest of the IT operations ensures the inputs to designing and deploying the system is correct. The COBIT controls give guidance on developing applications and systems in a repeatable well- documented process that ensures the documentation needed to assess the network for vulnerabilities accurately. This documentation is also needed when the security architects are designing the security of the network including web application firewalls and virtual patching systems. Failure to follow the controls for the development and deployment of the system could result in there being gaps or failures in the protection of the network. Up to date system inventories with versions and patch levels is needed to ensure the organization is protecting the right systems with the right input filters.

Once the systems are up and running on the network, the "deliver and support" controls ensure that the systems are properly maintained, patched and any problems or incidents are properly documented and dispositioned. The same controls are needed for the virtual patching system. Strong change management procedures will identify the need to update the interfaces to the virtual patching devices as well as the need to update the filtering rules.

Additional controls are identified in COBIT to provide oversight of the monitoring the configuration of the devices, any unauthorized access or changes, and any filtering rules that are triggered by network traffic. Additional monitoring of the network using tools other than the web application firewall should be performed to identify successful IPS evasion or entry points that were missed in the analysis that pertained to where to place the virtual patching system.

While the majority of this paper discusses the methods of deploying a virtual patching system in a traditional environment, we must also consider future research on the impacts of cloud architecture on virtual patching. While the cloud providers are responsible for maintaining the security of the cloud container, it is the client’s responsibility to ensure the security of the applications and environment they are running in the cloud. As more organizations move into the cloud, modifications of our existing models will be required to maintain their effectiveness.

Bibliography

Center for Internet Security. (n.d.). *The Critical Security Controls*. Retrieved December 2017, from CIS Controls™: https://www.cisecurity.org/controls/

Cobb, M. (n.d.). *Use Virtual Patching to Ease Short-staffed Patch Management Procedures*. Retrieved December 2017, from Tech Target: https://searchsecurity.techtarget.com/tip/Use-virtual-patching-to-ease-short-staffed-patch-management-procedures

Faust, Joseph (2010, June). *Reducing Organizational Risk Through Virtual Patching*. Retrieved from SANS Reading Room: https://www.sans.org/reading-room/whitepapers/threats/reducing-organizational-risk-virtual-patching-33589

Hoelzer, D. (2015, October). *What Are Their Vulnerabilities?: A SANS Survey on Continuous Monitoring*. Retrieved from SANS Reading Room: https://www.sans.org/reading-room/whitepapers/analyst/vulnerabilities-survey-continuous-monitoring-36377

ISACA. (n.d.). *COBIT 5*. Retrieved December 2017, from ISACA International: http://www.isaca.org/cobit/pages/default.aspx

ISACA. (n.d.). *COBIT 5: Enabling Processes*. Retrieved December 2017, from ISACA International: http://www.isaca.org/COBIT/Pages/COBIT-5-Enabling-Processes-product-page.aspx

Joshi, Ashlesha (n.d.). *Detecting Past and Present Intrusions through Vulnerability Specific Predicates*. Retrieved December 2017, from Colorado State Univesrity: http://www.cs.colostate.edu/~cs451/Slides/joshi05.pdf

Open Web Application Security Project (OWASP). (n.d.). *Virtual Patching Best Practices*. Retrieved January 2018, from OWASP, https://www.owasp.org/index.php/Virtual_Patching_Best_Practices

Open Web Application Security Project (OWASP). (n.d.). *OWASP Best Practices: Use of Web Application Firewalls*. Retrieved January 2018, from OWASP, https://www.owasp.org/index.php/Category:OWASP_Best_Practices:_Use_of_Web_Application_Firewalls

Open Web Application Security Project (OWASP). (n.d.). *OWASP Top 10 2017*. Retrieved January 2018, from OWASP, https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

Open Web Application Security Project (OWASP). (n.d.). *OWASP Virtual_Patching_Cheat_Sheet*. Retrieved January 2018, from OWASP, https://www.owasp.org/index.php/Virtual_Patching_Cheat_Sheet

Pubal, J. (2015, March). *Web Application Firewalls: Enterprise Techniques*. Retrieved from SANS Reading Room: https://www.sans.org/reading-room/whitepapers/application/web-application-firewalls-35817

Trend Micro. (n.d.). *Trend Micro - HYBRID CLOUD SECURITY*. Retrieved December 2017, from Trend Micro Products and Solutions: https://www.trendmicro.com/en_us/business/products/hybrid-cloud.html

Author Name, email@address

Zelster, L. (2015, July). Pros and Cons of Virtual Patching to Address Vulnerabilities. Retrieved December 2017, from Information Security in Business: https://zeltser.com/pros-and-cons-of-virtual-patching/#

Author Name, email@address

Last Updated: January 22nd, 2019

Upcoming Training

SANS Security East 2019	New Orleans, LA	Feb 02, 2019 - Feb 09, 2019	Live Event
Security East 2019 - SEC503: Intrusion Detection In-Depth	New Orleans, LA	Feb 04, 2019 - Feb 09, 2019	vLive
SANS London February 2019	London, United Kingdom	Feb 11, 2019 - Feb 16, 2019	Live Event
SANS Northern VA Spring- Tysons 2019	Tysons, VA	Feb 11, 2019 - Feb 16, 2019	Live Event
SANS New York Metro Winter 2019	Jersey City, NJ	Feb 18, 2019 - Feb 23, 2019	Live Event
SANS Scottsdale 2019	Scottsdale, AZ	Feb 18, 2019 - Feb 23, 2019	Live Event
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201902,	Feb 27, 2019 - Apr 04, 2019	vLive
SANS San Francisco Spring 2019	San Francisco, CA	Mar 11, 2019 - Mar 16, 2019	Live Event
SANS Madrid March 2019	Madrid, Spain	Mar 25, 2019 - Mar 30, 2019	Live Event
SANS 2019	Orlando, FL	Apr 01, 2019 - Apr 08, 2019	Live Event
Blue Team Summit & Training 2019	Louisville, KY	Apr 11, 2019 - Apr 18, 2019	Live Event
SANS Riyadh April 2019	Riyadh, Kingdom Of Saudi Arabia	Apr 13, 2019 - Apr 18, 2019	Live Event
Community SANS New York SEC503	New York, NY	Apr 29, 2019 - May 04, 2019	Community SANS
SANS Security West 2019	San Diego, CA	May 09, 2019 - May 16, 2019	Live Event
SANS Northern VA Spring- Reston 2019	Reston, VA	May 19, 2019 - May 24, 2019	Live Event
SANS Amsterdam May 2019	Amsterdam, Netherlands	May 20, 2019 - May 25, 2019	Live Event
San Antonio 2019 - SEC503: Intrusion Detection In-Depth	San Antonio, TX	May 28, 2019 - Jun 02, 2019	vLive
SANS San Antonio 2019	San Antonio, TX	May 28, 2019 - Jun 02, 2019	Live Event
SANS London June 2019	London, United Kingdom	Jun 03, 2019 - Jun 08, 2019	Live Event
SANSFIRE 2019	Washington, DC	Jun 15, 2019 - Jun 22, 2019	Live Event
Security Operations Summit & Training 2019	New Orleans, LA	Jun 24, 2019 - Jul 01, 2019	Live Event
SANS Paris July 2019	Paris, France	Jul 01, 2019 - Jul 06, 2019	Live Event
SANS Rocky Mountain 2019	Denver, CO	Jul 15, 2019 - Jul 20, 2019	Live Event
SANS Columbia 2019	Columbia, MD	Jul 15, 2019 - Jul 20, 2019	Live Event
SANS Boston Summer 2019	Boston, MA	Jul 29, 2019 - Aug 03, 2019	Live Event
SANS Chicago 2019	Chicago, IL	Aug 19, 2019 - Aug 24, 2019	Live Event
SANS Copenhagen August 2019	Copenhagen, Denmark	Aug 26, 2019 - Aug 31, 2019	Live Event
SANS Oslo September 2019	Oslo, Norway	Sep 09, 2019 - Sep 14, 2019	Live Event
SANS Network Security 2019	Las Vegas, NV	Sep 09, 2019 - Sep 16, 2019	Live Event
SANS London September 2019	London, United Kingdom	Sep 23, 2019 - Sep 28, 2019	Live Event
SANS OnDemand	Online	Anytime	Self Paced